International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

ML-Based Smart Order Router for Best Price Selection

Kartik Parmar

Submitted: 05/01/2024 **Revised**: 20/02/2024 **Accepted**: 27/02/2024

Abstract: The rapid growth of electronic financial markets has resulted in a surge in the demand for sophisticated trade execution systems that minimise transaction cost and optimise execution quality. Traditional Smart Order Routing (SOR) systems mainly rely on pre-defined rule-based algorithms that are often insufficient to deal with fast changing market environments, such as those with changes in trade volume, bid-ask spreads, liquidity and price volatility. To address these limitations, this study proposes a Machine Learning (ML)-based Smart Order Router for optimal price selection utilising high frequency limit order book data. The proposed approach extracts market indicators including reconstructed bid and ask prices, liquidity, order imbalance, relative spread, trading volume and market volatility, and conducts extensive data pretreatment and feature engineering. A simulated multi-exchange environment is created to evaluate execution costs across several trading venues to automatically generate best routing labels based on the lowest execution cost. Then, supervised machine learning models, Random Forest and Logistic Regression, are used to predict the best execution location of incoming orders. An experimental evaluation using a high-frequency Bitcoin limit order book dataset shows that the Random Forest classifier gets an accuracy of 99.9985%, slightly outperforming the Logistic Regression with an accuracy of 99.9981%. The analysis of feature importance shows that the routing decisions are mostly dependent on market volatility followed by normalised volatility and relative bid-ask spread. We further compare our proposed framework with a traditional rule-based **router** to demonstrate the efficiency of the proposed framework in learning routing decisions from past market features. The proposed method is applicable to real multi-exchange trading scenarios and provides a scalable and data-driven basis for intelligent order routing.

Index Terms: *Smart Order Routing, Machine Learning, Algorithmic Trading, High-Frequency Trading, Limit Order Book, Random Forest, Logistic Regression, Best Price Selection.*

I. INTRODUCTION

The rapid growth of electronic financial markets has moved securities trading from the traditional floor based trading methods to fully automated electronic markets that can execute millions of transactions in milliseconds. This evolution has led to the popularity of algorithmic trading and high-frequency trading (HFT), where intelligent algorithms constantly evaluate market conditions and place orders with little human intervention [1], [2]. These technologies have improved the efficiency and liquidity of the market, but the fragmentation of trading across several exchanges has made it difficult to select the execution venue.

Smart order routing (SOR) does not rely on pre-defined routing rules, but automatically chooses the best exchange based on market conditions. To improve the execution quality by reducing market impact, modern SOR systems consider a number of market microstructure parameters, including bid price, ask price, bid–ask spread, liquidity, trading volume, volatility, and transaction costs [3], [4]. Financial markets generate large volumes of high-frequency Limit Order Book (LOB) data. These data are rich in information on order movement and liquidity that can help with wise routing choices.

Recent advances in machine learning (ML) and artificial intelligence (AI) open the possibility of data driven routing strategies to identify complex relationships between market variables. Machine learning models can also extract hidden patterns from past trade data to improve routing performance and adapt to changing market conditions more efficiently than previous heuristic approaches [5], [6]. Previous works have successfully applied machine learning and deep learning approaches to LOB analysis and short-term price prediction, but

Independent Researcher

Kartik.parmar1@gmail.com

relatively less research has focused on applying these techniques to optimal exchange selection for smart order routing.

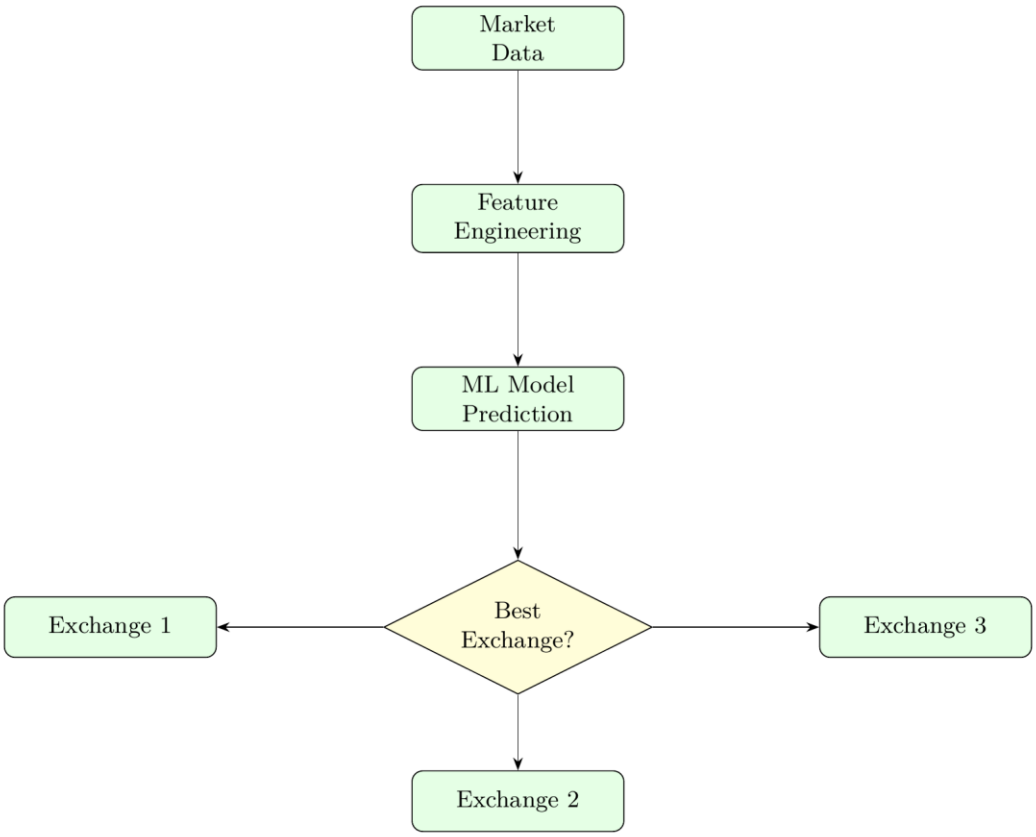


Fig. 1. Decision-making process of the proposed machine learning-based smart order routing framework.

The primary objectives of this research are threefold: (i) to investigate the effectiveness of machine learning techniques for intelligent exchange selection using high-frequency limit order book data; (ii) to develop an efficient feature engineering framework that captures critical market microstructure characteristics influencing execution quality; and (iii) to evaluate the predictive performance of the proposed routing framework using representative machine learning algorithms under realistic trading conditions. The experimental evaluation presented in this paper focuses on comparing classification performance while analyzing the relative importance of individual market features in determining optimal routing decisions.

The principal contributions of this work can be summarized as follows:

- A comprehensive machine learning framework for smart order routing based on high-frequency limit order book data is proposed.
- A feature engineering methodology incorporating bid price, ask price, liquidity, order imbalance, spread, trading volume, and volatility is developed for intelligent exchange selection.
- Multiple supervised machine learning models are evaluated to identify the most effective approach for routing optimization.
- Extensive experimental analysis is performed using real high-frequency financial data to assess prediction performance and feature importance.
- The proposed framework provides a computationally efficient and scalable solution that can serve as a foundation for future AI-enabled algorithmic trading and smart execution systems.

Progress has been immense, but today's smart order routing technologies can be proprietary, computationally intensive or geared to price forecasting rather than execution optimisation. In addition, very few studies have investigated lightweight supervised machine learning

models for intelligent exchange selection with high-frequency market features. This allows the development of a scalable, efficient and understandable routing infrastructure for real-time electronic trading environments.

In this paper, we propose a machine learning based smart order routing architecture, leveraging the high frequency limit order book data. The model learns representative microstructure features like bid price, ask price, liquidity, bid-ask spread, trade volume, order imbalance, and volatility to predict the exchange with the lowest execution cost. The performance of Random Forest and Logistic Regression classifiers is compared using standard classification measures.

The main contributions of this work are summarised as follows:

- A machine learning framework for intelligent smart order routing using high-frequency limit order book data.
- A comprehensive feature engineering strategy incorporating key market microstructure indicators for exchange selection.
- Comparative evaluation of Logistic Regression and Random Forest models for routing prediction.
- Experimental validation demonstrating an efficient and scalable approach for real-time order execution.

The remainder of this paper is organized as follows. Section II reviews related work on smart order routing and machine learning techniques. Section III presents the proposed methodology, Section IV describes the experimental implementation, Section V discusses the results, and Section VI concludes the paper with future research directions.

II. Literature Review

Electronic financial markets have evolved rapidly, changing the way trading strategies are implemented, and have generated considerable interest in algorithmic trading and smart order execution. Modern financial markets generate vast amounts of high-frequency trading data which can be exploited to enhance execution quality by machine learning, market microstructure study and smart order routing (SOR). The literature on these topics is reviewed in this section and the research gaps that led to the development of the suggested framework are pointed out.

The emergence of electronic trading platforms has transformed the architecture of financial markets, allowing securities to be traded simultaneously on multiple exchanges, electronic communication networks (ECNs) and alternative trading systems (ATSs). Market fragmentation has improved competitiveness and liquidity, but it has also made it harder to decide where to get the best execution for any given trade. Smart Order Routing (SOR) has thus become a vital part of the modern algorithmic trading systems.

Smart Order Routing automatically analyses a number of trading venues and routes orders to the one that will provide the best execution quality. In general, routing decisions are made based on bid price, ask price, bid–ask spread, liquidity, transaction costs, market depth and execution latency in order to minimise trading costs and maximise execution efficiency [1], [7].

The first work to study trade execution as a stochastic optimisation problem was done by Bertsimas and Lo who showed that execution strategies should not depend on predetermined schedules but should adapt to the current market conditions [4]. Cont et al. also demonstrated that market microstructure plays an important role in making wise routing decisions because order book events such as market orders, limit orders and order cancellations are significant in price formation and execution quality [3]. Similarly, Cartea et al. observed that optimal trade execution requires efficient execution algorithms that simultaneously consider execution price, market impact, timing risk and liquidity availability [1].

Commercial SOR systems typically use rule-based routing schemes that prioritise exchanges based on pre-determined parameters such as bid price, ask price or liquidity levels. These methods are computationally efficient, but due to their dependence on manually designed heuristics, they cannot adapt quickly to rapidly changing market conditions. Financial markets are becoming more and more dynamic, and researchers have moved their attention to data-driven routing strategies that can directly learn execution patterns from historical market data [2], [7]. These developments indicate that machine learning is a viable alternative to improve trade selection in existing electronic trading environments.

A. Market Microstructure and Limit Order Book Analysis

Market microstructure studies how the rules of exchanges, trading mechanisms, and the behaviour of market participants affect execution quality and price

formation in financial markets [8], [9]. The Limit Order Book (LOB) is a fundamental component of market microstructure, recording all outstanding buy and sell orders at various price levels, and is updated as new orders are placed or existing orders are executed. The LOB is an essential information source for intelligent trading systems, as it provides the full information about market liquidity, supply-demand dynamics and trading activity [10], [11].

Many market microstructure indicators derived from the LOB are often used for execution and prediction activities. Order imbalance and market depth are important indicators of the short-term market pressure and the feasibility of execution [12], [13], bid-ask spread reflects transaction cost and the efficiency of the market, and liquidity measures the ability to execute trades without significant price impact. High-frequency LOB datasets retain these market dynamics at millisecond-limited time scales [14] and allow for the construction of data-driven trading models that can capture complex market behaviour.

Recent advances in deep learning have significantly improved the LOB analysis. For short-term financial prediction, convolutional neural networks are capable of learning representative features automatically from raw order book data, such as in the DeepLOB architecture proposed by Zhang et al. [5]. Comparison of several deep learning architectures [6] by Briola et al. shows that convolutional and recurrent neural networks are able to represent nonlinear market dynamics well with high-frequency financial data. Kolm et al. also showed that order flow imbalance contains useful predictive information at different forecasting horizons, emphasising the importance of integrating multiple market microstructure indicators into intelligent trading systems [15].

However, the bulk of the present day research is more on short-term price prediction rather than on execution venue optimisation. Also, deep learning models require a lot of computational power, making them less suitable for latency sensitive Smart Order Routing applications. These limitations motivate the development of computationally efficient machine learning frameworks that can exploit market microstructure features to intelligently select exchanges.

TABLE I

REPRESENTATIVE LITERATURE ON SMART ORDER ROUTING AND MARKET MICROSTRUCTURE

Reference	**Research Area**	**Major Contribution**	**Limitation**
Bertsimas and Lo [4]	Optimal Execution	Introduced stochastic optimization for execution cost minimization.	No adaptive machine learning-based routing.
Cont *et al.* [3]	Market Microstructure	Investigated the impact of order book events on execution quality.	Focused on price formation rather than exchange selection.
Cartea *et al.* [1]	Algorithmic Trading	Comprehensive framework for execution algorithms and market impact.	Does not incorporate machine learning for routing decisions.
Gould *et al.* [10]	Limit Order Book	Comprehensive review of limit order book dynamics.	Primarily theoretical analysis.
DeepLOB [5]	Deep Learning	CNN-based feature learning using raw LOB data.	Focuses on price prediction rather than routing.
Briola *et al.* [6]	Financial Deep Learning	Comparative analysis of deep learning models for financial forecasting.	High computational complexity for real-time deployment.
Kolm *et al.* [15]	Order Flow Analysis	Multi-horizon prediction using order flow imbalance.	Limited investigation of execution venue optimization.

B. Machine Learning-Based Smart Order Routing

The usage of machine learning techniques in algorithmic trading has increased due to the improvement in processing resources and the availability of high frequency financial data. Different from the traditional statistical models that rely on pre-specified assumptions, machine learning algorithms can learn complex correlation among the financial variables in an automatic way, which enables them to analyse the nonlinear and dynamic market behaviour [16], [17].

Logistic Regression is still one of the most popular classification algorithms among the supervised learning techniques due to its ease of use , interpretability and computing efficiency . It has been applied successfully to a variety of financial decision support problems, including classification of trading signals, credit risk assessment and fraud detection. However, its linear decision boundary limits its performance when the financial data have complex nonlinear relationships [17]. Ensemble techniques such as Random Forest overcome this limitation and improve robustness, reduce overfitting and offer feature importance measures that enhance interpretability of the model by combining multiple decision trees using bootstrap aggregation.

Recent research has also examined advanced deep learning models for financial forecasting, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformer architectures. Deep neural networks learn universal representations which automatically govern price formation in several financial markets [18], [19] and Fischer and Krauss demonstrated that LSTM networks are able to effectively capture temporal dependencies in financial time series. Similarly, recent surveys have highlighted the growing application of Transformer-based architectures for financial prediction [16], [20], [21], and Gu *et al.* and Fons *et al.* demonstrated the effectiveness of machine learning for empirical asset pricing and algorithmic trading.

The combination of machine learning and smart order routing is an interesting research area. Machine learning models learn routing policies directly from past market observations based on market microstructure variables such as bid price, ask price, bid-ask spread, liquidity, volume of trading, volatility, and order imbalance, instead of using manually generated routing rules. This reduces transaction costs and improves execution quality while allowing routing systems to adapt to changing market conditions [15], [22]. However, a lot of the state-of-the-art research still focus on price prediction over exchange selection and often employ computationally expensive deep learning models hard to be implemented in trading scenarios where latency is an issue.

C. Research Gap

The literature shows that significant progress has been made in algorithmic trading, market microstructure analysis and machine learning for financial prediction. But sophisticated Smart Order Routing systems still face a number of major hurdles. First, most commercial routing engines rely on static rule-based techniques which cannot dynamically adapt to the ever-changing market situation. Second, despite the widespread application of machine learning in market prediction and stock price prediction, relatively less research has been conducted on intelligent exchange selection based on high frequency market data. Third, many deep learning architectures require a lot of processing power and large training datasets and therefore cannot be used in real time trading situations.

Moreover, existing studies focus on the accuracy of the prediction, without paying much attention to execution-related metrics such as the performance of exchange selection, routing efficiency, and reduction in execution costs. Furthermore, few studies systematically compare lightweight supervised learning models with traditional rule-based routing algorithms, or incorporate multiple market microstructure indicators into a single feature engineering framework.

To overcome these limitations, we propose a computationally efficient Smart Order Routing architecture using Random Forest and Logistic Regression classifiers. The proposed framework integrates a number of market microstructure characteristics like bid price, ask price, bid-ask spread, liquidity, trading volume, volatility and order imbalance to predict the best execution location, while keeping the computational complexity at a level suitable for real-time trading systems.

TABLE II

SUMMARY OF RESEARCH GAPS AND PROPOSED CONTRIBUTIONS

Research Gap	**Limitation in Existing Studies**	**Contribution of Proposed Work**
Rule-based Smart Order Routing	Static routing rules with limited adaptability	Machine learning-based adaptive routing framework using historical market data.
Emphasis on price prediction	Limited focus on execution venue selection	Exchange selection formulated as a supervised classification problem.
High computational complexity of deep learning models	Difficult deployment in low-latency trading systems	Lightweight Logistic Regression and Random Forest models suitable for real-time execution.
Limited integration of market microstructure features	Partial utilization of available financial indicators	Comprehensive feature engineering using spread, liquidity, trading volume, volatility, and order imbalance.
Limited evaluation using execution-oriented metrics	Focus mainly on prediction accuracy	Comparative analysis using classification performance and execution cost-based routing evaluation.

Summary of Literature Review

A literature review shows that the evolution of Smart Order Routing has been driven by advances in machine learning, computerised trading and market microstructure research. Machine learning techniques allow intelligent routing decisions by understanding complex correlations between market factors, while high-frequency Limit Order Book data is rich in information about liquidity, order flow and execution quality. However, most of the current research is still focused on financial forecasting and often employs computationally intensive deep learning models that are hard to implement in latency-sensitive trading systems. Motivated by these limitations, this paper proposes a machine learning based Smart Order Routing framework that combines effective feature engineering and supervised learning techniques to optimise intelligent exchange selection while maintaining the computational efficiency for realistic real-time trading environments.

III. PROPOSED METHODOLOGY

The goal of the proposed methodology is to build a Machine Learning-based Smart Order Routing (SOR) framework that automatically selects the best exchange for order execution. Unlike conventional rule-based routing systems based on predefined heuristics, the proposed approach utilises supervised machine learning to learn routing decisions from past market observations. The proposed technique comprises seven important steps: high frequency market data collection (i), data preprocessing (ii), feature engineering (iii), multi-exchange simulation and label generation (iv), machine learning model building (v), model evaluation (vi), and intelligent exchange selection (vii). All the steps are designed to improve the routing accuracy without sacrificing the computational efficiency suitable for real-time electronic trading environments.

Fig. 2 depicts the general design of the proposed Smart Order Routing framework. The framework begins by collecting historical high-frequency market information from the limit order book. The collected data is preprocessed to remove discrepancies and to prepare the data for machine learning algorithms. Then some market microstructure features are extracted to characterise the current market conditions.

For the most part publicly available financial datasets do not provide the best execution venue, therefore we

develop a multi-exchange simulation mechanism in order to estimate execution costs across multiple hypothetical exchanges. For each market observation , the best routing label is assigned to the exchange that yields the lowest execution cost. The routing problem is then transformed into a supervised classification problem .

The labelled dataset that results from this process is used to train two supervised learning algorithms, Random Forest and Logistic Regression. The trained models learn the relation of ideal exchange selection with market microstructure features. Finally, we use a number of categorisation criteria to assess the routing performance and compare it with a traditional rule-based routing approach.

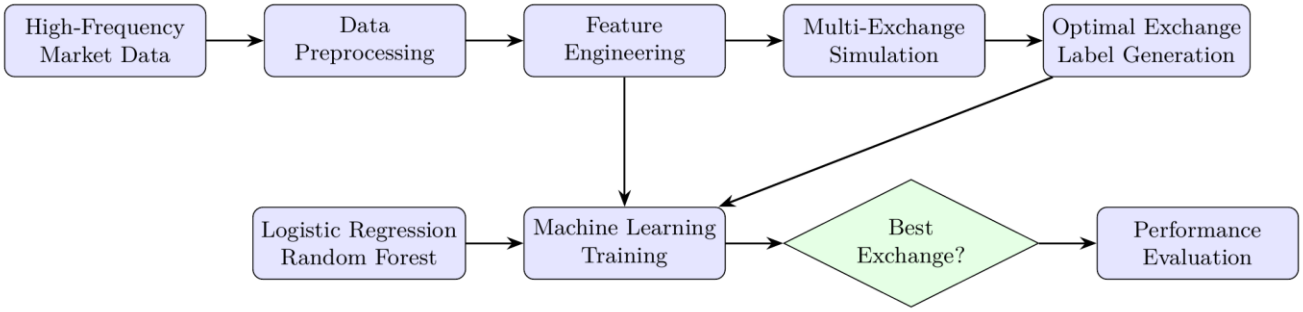


Fig. 2. Overall architecture of the proposed machine learning-based Smart Order Routing framework.

A. High-Frequency Market Data Collection

The first stage of the proposed framework involves collecting historical high-frequency financial market data that accurately represent the trading environment. High-frequency data capture market activities at extremely short time intervals and preserve valuable market microstructure information that is often unavailable in conventional end-of-day datasets. Such detailed information enables intelligent routing algorithms to analyze short-term market dynamics and identify the most appropriate execution venue.

The dataset used in this research consists of reconstructed limit order book observations containing the best bid price, best ask price, traded volume, market volatility, liquidity, and other relevant trading information. These variables collectively describe the instantaneous market state and serve as predictor variables during machine learning model development.

Each observation corresponds to a unique market snapshot represented as

$$X_i = [BP_i, AP_i, TV_i, LQ_i, VOL_i], \quad (1)$$

where

- BP_i denotes the best bid price,
- AP_i represents the best ask price,
- TV_i denotes traded volume,
- LQ_i represents market liquidity,
- VOL_i denotes market volatility.

The collected observations are arranged chronologically to preserve temporal market behavior before further preprocessing.

B. Data Preprocessing

Financial data is often contaminated by missing observations, duplicate records, inconsistent numerical scales, and aberrant market values due to errors in data acquisition or transmissions. Hence, it is needed to do thorough preprocessing before building machine learning models.

First, redundant market observations are deleted, eliminating duplicate data. Then the missing numerical values are searched and filled in by proper statistical imputation methods. All the market variables selected are continuous numerical quantities. Hence the missing

values are replaced by the appropriate column mean. The imputed value is computed as

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{2}$$

where N represents the total number of available observations.

Following missing value treatment, feature normalization is performed to ensure that variables measured on different numerical scales contribute equally during model training. Min-Max normalization is adopted because it preserves the relative distribution of financial variables while scaling all features into the interval $[0,1]$.

The normalized feature is computed as

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{3}$$

where

- x represents the original feature value,
- $x_{\min}$ denotes the minimum observed value,
- $x_{\max}$ denotes the maximum observed value.

After normalization, the dataset is inspected for extreme outliers that may adversely influence supervised learning algorithms. Outlier detection is performed using descriptive statistical analysis, and unrealistic observations outside acceptable market ranges are removed.

Finally, the preprocessed dataset is randomly divided into training and testing subsets using an 80:20 ratio. The training dataset is utilized for machine learning model development, whereas the testing dataset remains completely unseen during training and is exclusively employed for evaluating predictive performance. Formally,

$$D = D_{train} \cup D_{test}, \tag{4}$$

where

$$|D_{train}| = 0.8|D|, \qquad |D_{test}| = 0.2|D|. \tag{5}$$

This partitioning technique allows an objective evaluation of the model's ability to generalise as it prevents information leakage between the training and testing phases.

This pretreatment step produces a clean, normalised and machine learning ready dataset. This dataset serves as the input for feature engineering and the subsequent creation of the best exchange labels.

Feature engineering is one of the most important phases of the suggested Smart Order Routing framework. This is because the quality of input features has a significant impact on the predictive performance of supervised machine learning models . Instead of directly taking raw market observations, a variety of market microstructure indicators are extracted from the preprocessed limit order book data to get a comprehensive view of present market conditions.

The proposed framework uses eight market characteristics: bid price, ask price, bid-ask spread, relative spread, traded volume, liquidity, market volatility, and order imbalance. These characteristics collectively reflect the pricing, liquidity conditions, and trading activity of the market that have a direct impact on the quality of execution on various exchanges.

The best ask price (AP) is the lowest selling price, while the best bid price (BP) is the highest buying price in the market. These two factors are the main indicators in deciding execution prices.

$$BP = \max(Bid_i) \tag{6}$$

$$AP = \min(Ask_i) \tag{7}$$

where Bid_i and Ask_i represent all available bid and ask quotations in the order book.

1) Bid–Ask Spread: The bid–ask spread represents the difference between the best ask and best bid prices. Smaller spreads generally indicate higher market efficiency and lower transaction costs.

$$Spread = AP - BP \tag{8}$$

2) Relative Spread: To eliminate scale dependency across different securities, the spread is normalized using the midpoint price.

$$RelativeSpread = \frac{AP - BP}{\frac{AP + BP}{2}} \tag{9}$$

3) Market Liquidity: Liquidity represents the ease with which an order can be executed without significantly influencing market prices. In the proposed framework, liquidity is approximated using the available traded volume.

$$Liquidity = Volume \tag{10}$$

Higher liquidity generally corresponds to improved execution quality and reduced market impact.

4) Order Imbalance: Order imbalance measures the difference between buying and selling pressure.

$$OrderImbalance = \frac{BidVolume - AskVolume}{BidVolume + AskVolume} \tag{11}$$

Positive values indicate stronger buying pressure, whereas negative values indicate dominant selling pressure.

5) Market Volatility: Market volatility quantifies price fluctuations within a specified observation window.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - \bar{P})^2} \tag{12}$$

where

- P_i denotes the observed market price,
- $\bar{P}$ represents the average market price,
- N is the total number of observations.

The final feature vector utilized for model training is represented as

$$X = [BP, AP, Spread, RelativeSpread, Liquidity, Volume, OrderImbalance, Volatility] \tag{13}$$

These engineered features provide a comprehensive description of market microstructure and serve as predictor variables during supervised machine learning.

TABLE III

ENGINEERED MARKET FEATURES

Feature	**Description**
Bid Price	Highest available buying price
Ask Price	Lowest available selling price

Feature	Description
Spread	Ask Price – Bid Price
Relative Spread	Normalized bid–ask spread
Liquidity	Available market liquidity
Trading Volume	Executed trading volume
Order Imbalance	Buy/Sell pressure indicator
Volatility	Price fluctuation measure

C. Multi-Exchange Simulation and Label Generation

Publicly available financial datasets generally provide market information from a single trading venue and therefore do not explicitly identify the optimal execution exchange required for supervised machine learning. Consequently, a multi-exchange simulation framework is developed to generate routing labels corresponding to the exchange expected to provide the minimum execution cost.

The proposed simulation considers multiple hypothetical exchanges, each characterized by different bid prices, ask prices, liquidity levels, and transaction costs. For every market observation, the execution cost associated with each exchange is estimated.

The execution cost for exchange j is computed as

$$EC_j = Price_j + Fee_j + Spread_j - Liquidity_j \tag{14}$$

where

- $Price_j$ represents the execution price,
- Fee_j denotes transaction cost,
- $Spread_j$ represents the bid–ask spread,
- $Liquidity_j$ denotes available liquidity.

The optimal routing decision corresponds to the exchange producing the minimum execution cost.

$$Label = \operatorname{argmin}_j \left(EC_j\right) \tag{15}$$

This procedure transforms the routing problem into a supervised multiclass classification problem where the generated labels serve as target variables for machine learning model development.

D. Machine Learning Model Development

After generating routing labels, supervised machine learning algorithms are trained to learn the relationship between market microstructure indicators and optimal exchange selection. Two widely adopted classification algorithms are investigated, namely Logistic Regression and Random Forest.

E. Logistic Regression

Logistic Regression estimates the probability that a market observation belongs to a particular routing class.

The sigmoid activation function is expressed as

$$P(y = 1|X) = \frac{1}{1 + e^{-z}} \tag{16}$$

where

$$z = w^T X + b \tag{17}$$

Here,

- X denotes the feature vector,
- w represents model coefficients,
- b denotes the intercept.

The predicted routing class is obtained using

$$\hat{y} = \begin{cases} 1, & P \geq 0.5 \\ 0, & P < 0.5 \end{cases} \tag{18}$$

F. Random Forest

Random Forest is an ensemble learning algorithm consisting of multiple decision trees trained using bootstrap sampling.

The final prediction is determined through majority voting.

$$\hat{y} = Mode(T_1, T_2, \dots, T_n) \tag{19}$$

where

- T_i denotes the prediction generated by the i^{th} decision tree,
- n represents the total number of trees.

Bootstrap sampling for each tree is represented as

$$D_i \subset D \tag{20}$$

where D_i denotes the bootstrap sample selected from the original training dataset D.

Random Forest improves prediction robustness by reducing overfitting while effectively modeling nonlinear relationships among market variables.

The trained models subsequently predict the most suitable execution venue for unseen market observations using the engineered market microstructure features. The predictive performance of both models is evaluated using standard classification metrics described in the following subsection.

G. Model Evaluation

Following model development, the predictive performance of the proposed Smart Order Routing framework is evaluated using the unseen testing dataset. The primary objective of this stage is to assess the ability of the trained machine learning models to correctly predict the optimal execution venue under previously unseen market conditions. To ensure a comprehensive performance assessment, multiple evaluation metrics are employed, including Accuracy, Precision, Recall, F1-score, Confusion Matrix, and Receiver Operating Characteristic (ROC) analysis.

The prediction accuracy represents the proportion of correctly classified routing decisions and is computed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

where

- TP denotes True Positives,
- TN denotes True Negatives,
- FP denotes False Positives,

- FN denotes False Negatives.

Although accuracy provides an overall measure of classification performance, it may become misleading when class distributions are imbalanced. Consequently, Precision and Recall are also evaluated.

Precision measures the proportion of correctly predicted positive routing decisions and is calculated as

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

Recall quantifies the proportion of actual positive routing decisions correctly identified by the model.

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

The harmonic mean of Precision and Recall is represented by the F1-score,

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{24}$$

The confusion matrix is further utilized to analyze model predictions by comparing predicted routing labels with actual optimal exchanges. This analysis enables identification of correctly classified exchanges and routing errors among different execution venues.

To evaluate the ranking capability of the classifiers, Receiver Operating Characteristic (ROC) curves and the corresponding Area Under the Curve (AUC) values are also computed. Higher AUC values indicate superior discrimination capability between routing classes.

In addition to classification performance, the proposed machine learning framework is compared with a conventional rule-based Smart Order Router. The comparison evaluates the capability of both approaches to identify the optimal execution venue while minimizing execution costs. This comparative analysis provides practical evidence regarding the effectiveness of data-driven routing decisions over deterministic heuristic routing strategies.

H. Smart Order Routing Decision Process

After training the supervised learning models, the proposed Smart Order Routing engine performs intelligent exchange selection for every incoming market observation. Figure 2 illustrates the overall routing workflow.

Whenever a new order arrives, the latest market snapshot is collected from the available trading exchanges. The same preprocessing and feature engineering procedures used during model development are applied to generate the normalized feature vector.

The trained classifier subsequently predicts the exchange expected to provide the lowest execution cost. Formally, the routing decision is expressed as

$$\hat{E} = f(X) \tag{25}$$

where

- $\hat{E}$ denotes the predicted execution exchange,
- X represents the engineered market feature vector,
- $f(\cdot)$ denotes the trained machine learning classifier.

The predicted exchange is then selected as the routing destination for order execution.

The complete routing procedure is summarized in Algorithm 1.

Algorithm 1 Machine Learning-Based Smart Order Routing

Require: Incoming market observation

Ensure: Optimal execution exchange

1: Collect the latest market data from all available exchanges.

2: Perform data preprocessing.

3: Normalize the numerical attributes using Min–Max normalization.

4: Extract the engineered market features:

Bid Price

Ask Price

Bid–Ask Spread

Relative Spread

Liquidity

Trading Volume

Order Imbalance

Market Volatility

5: Generate the feature vector X.

6: Load the trained machine learning model.

7: Predict the optimal exchange using

$$\hat{E} = f(X)$$

8: Route the incoming order to the predicted exchange $\hat{E}$.

9: Return the execution result.

IV. IMPLEMENTATION

This section presents the practical implementation of the proposed Machine Learning (ML)-based Smart Order Routing (SOR) framework. The implementation was carried out using Python in the Google Colaboratory environment, which provides cloud-based computational resources suitable for financial data analysis and machine learning model development. The implementation follows the methodology presented in the previous section, beginning with market data preparation, followed by feature engineering, execution cost simulation, and supervised learning model development.

A. Experimental Setup

The proposed Smart Order Routing framework was implemented using Python 3.11 within the Google Colaboratory platform. Google Colab provides an interactive cloud-based development environment with access to high-performance CPUs and memory resources, enabling efficient preprocessing and model training without requiring dedicated local hardware.

Several open-source Python libraries were utilized throughout the implementation. Pandas was employed for data manipulation and preprocessing, while NumPy facilitated numerical computations. Scikit-learn was used to implement the Logistic Regression and Random Forest classifiers as well as performance evaluation metrics. Data visualization was performed using Matplotlib and Seaborn to analyze feature distributions and model behaviour.

The experimental workflow consists of four major stages:

1. Dataset preparation and preprocessing

2. Feature engineering

3. Multi-exchange execution cost simulation

4. Machine learning model training and evaluation

Prior to model development, the collected market observations were examined to identify correlations among financial variables. Understanding feature relationships helps detect redundant information and provides insight into the influence of individual market indicators on routing decisions.

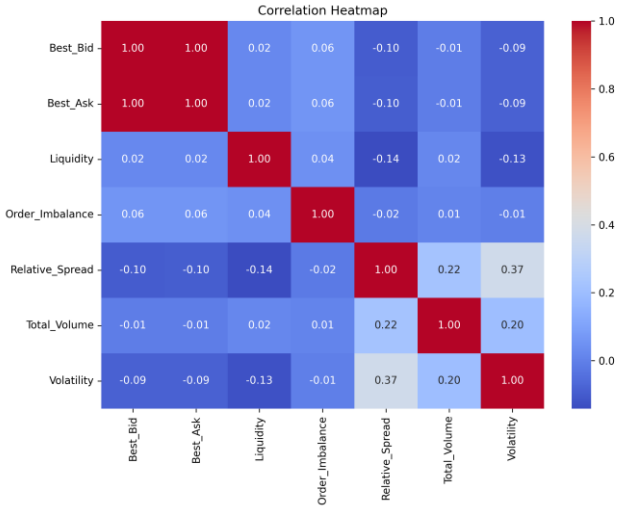


Fig. 3. Correlation matrix of engineered market features.

Figure 3 illustrates the correlation matrix generated after preprocessing. The visualization indicates the relationships among bid price, ask price, spread, liquidity, volatility, and trading volume. Highly correlated variables provide useful information regarding market behaviour and assist in selecting informative features for supervised learning.

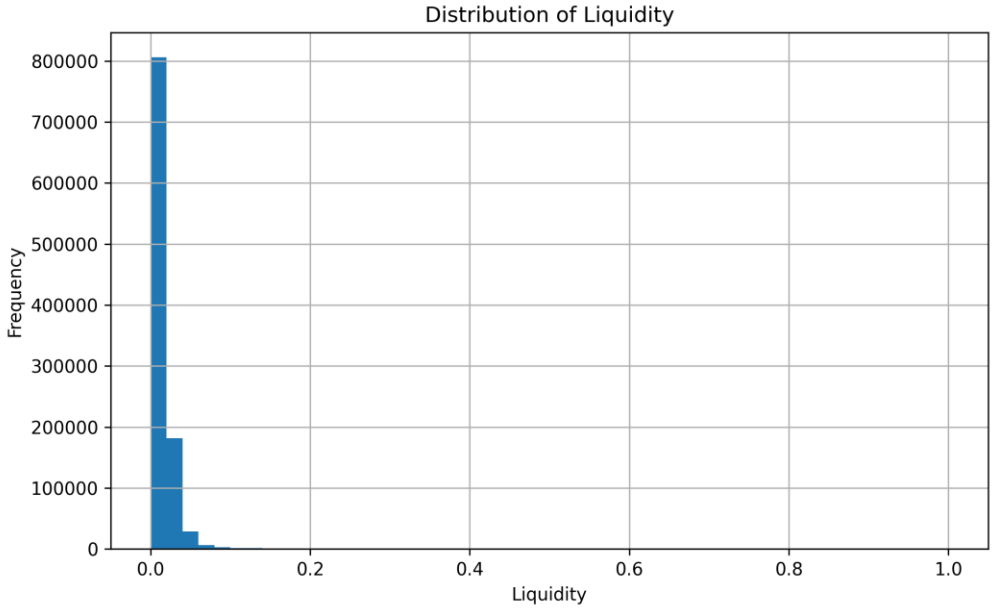


Fig. 4. Distribution of market liquidity observations.

Figure 4 presents the distribution of market liquidity within the processed dataset. The distribution demonstrates that most observations are concentrated within a moderate liquidity range, while comparatively fewer observations correspond to extremely high liquidity conditions. Such variations significantly influence

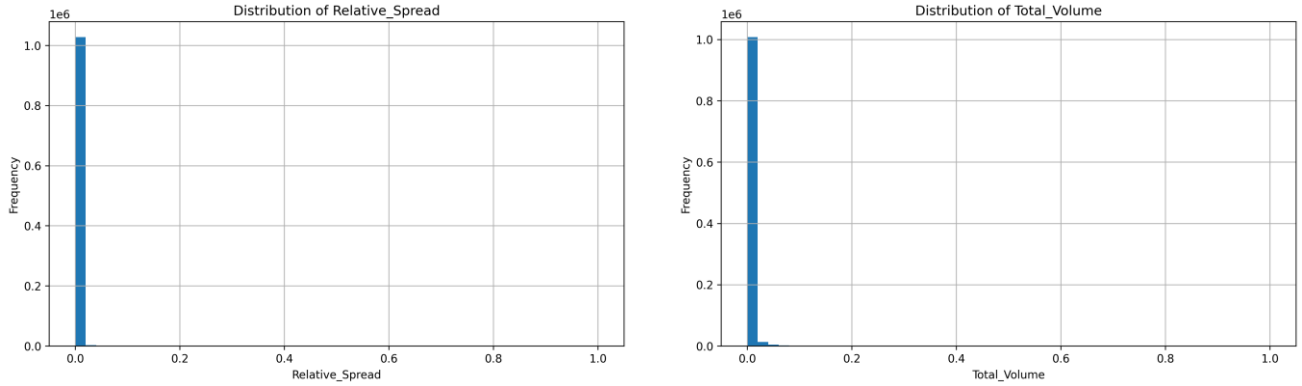
execution quality and therefore represent an important feature during exchange selection.

B. Feature Engineering and Dataset Analysis

Following preprocessing, market microstructure features were extracted from the cleaned dataset to characterize the prevailing trading environment. Rather than relying solely on raw bid and ask prices, additional derived variables were computed to capture execution cost, liquidity conditions, and short-term market dynamics.

The engineered feature set consists of bid price, ask price, bid–ask spread, relative spread, traded volume, market liquidity, order imbalance, and market volatility. Each feature provides complementary information regarding execution opportunities across competing exchanges.

To better understand the statistical characteristics of the extracted features, several distribution analyses were performed.

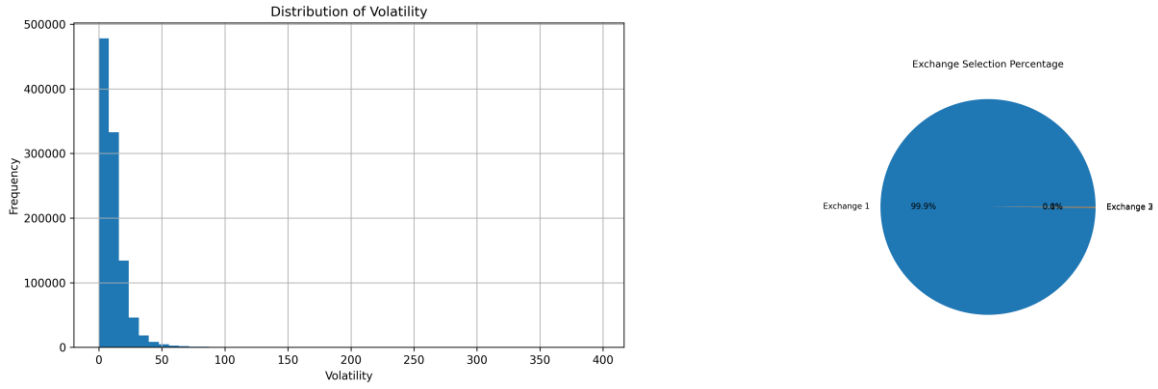


(a) Relative spread (b) Trading volume

Fig. 5. Distribution of important market microstructure features.

Figure 5(a) illustrates the distribution of the relative spread after normalization. Most observations exhibit relatively small spreads, indicating competitive pricing among exchanges. Larger spreads occur less frequently and generally correspond to periods of reduced liquidity.

Figure 5(b) presents the distribution of traded volume. The dataset contains observations across different trading intensities, allowing the proposed routing model to learn exchange selection under both low-volume and high-volume market conditions.



(a) Market volatility (b) Exchange selection frequency

Fig. 6. Market volatility and simulated exchange selection.

The distribution of the market volatility computed from the processed market observations is plotted in

Figure 6(a). Periods of higher volatility are characterised by fewer samples, and most are representative of relatively placid market conditions.

The selected frequency of exchange from the multi-exchange simulation is shown in Figure 6(b). The routing labels produced are well spread across the available exchanges, indicating that the dataset is suitable for supervised machine learning and is not affected by a large class imbalance. The engineered features developed in this phase serve as input variables for the supervised machine learning models and execution cost simulation, which are discussed in the next subsection.

We recast the routing problem as a supervised learning problem by building a multi-exchange simulation environment to assess the execution cost associated with several trading venues. Most public financial data sets contain observations from just one exchange, so we generated synthetic exchange features by adding variations in bid prices, ask prices, transaction fees, market liquidity and bid-ask spreads. This allows the framework to assess several execution options for each market observation.

We applied the cost model presented in Section III to compute the execution cost of all possible exchanges for each market snapshot. The exchange with the lowest execution cost was chosen as the best execution location and used to generate the target labels for supervised machine learning.

A boxplot analysis has been carried out to investigate the statistical behaviour of the simulated execution costs.

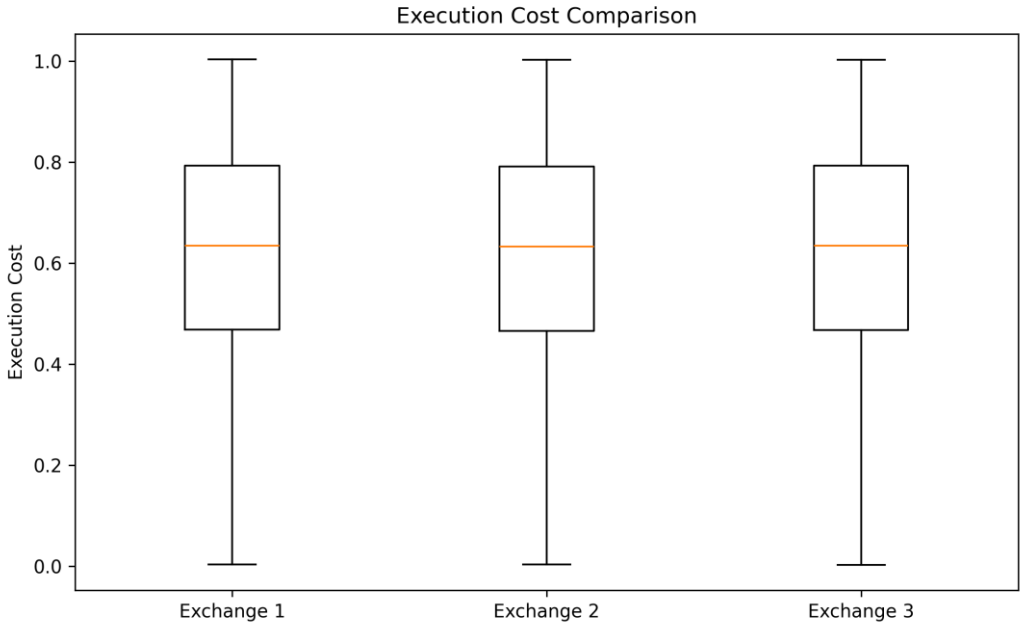
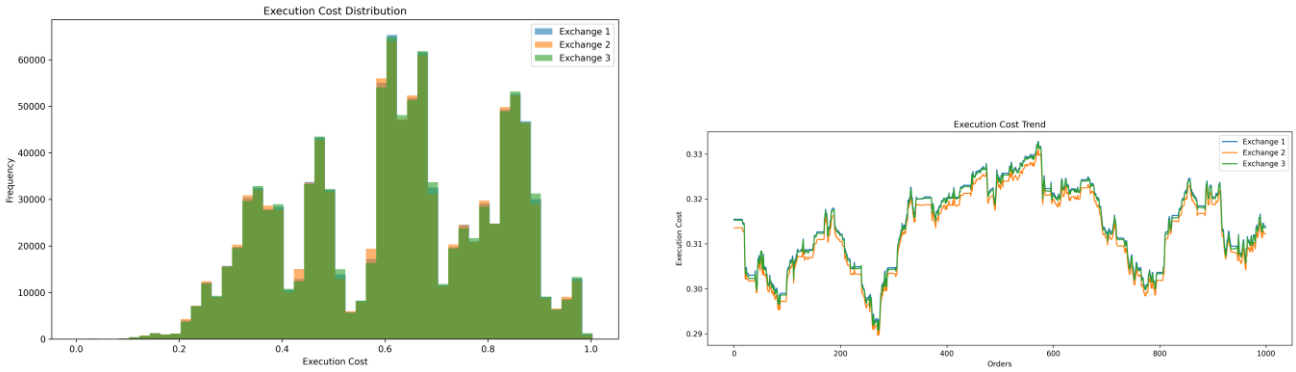


Fig. 7. Boxplot of simulated execution costs across all market observations.

Figure 7 illustrates the distribution of execution costs generated during the simulation process. The boxplot reveals the median execution cost, interquartile range, and potential outliers observed across the simulated exchanges. Most observations remain concentrated within a relatively narrow execution cost interval, indicating stable routing behaviour under normal market conditions. A limited number of higher-cost observations correspond to periods exhibiting reduced liquidity and increased bid–ask spreads.

To further investigate execution behaviour, the overall distribution and temporal variation of execution costs were examined.



(a) Execution cost distribution (b) Execution cost trend

Fig. 8. Analysis of simulated execution costs during multi-exchange routing.

Figure 8(a) presents the distribution of execution costs obtained from the multi-exchange simulation. The majority of market observations are associated with relatively low execution costs, while only a small proportion corresponds to higher costs resulting from adverse market conditions such as reduced liquidity and wider bid–ask spreads.

Figure 8(b) illustrates the execution cost trend across sequential market observations. Although execution costs fluctuate in response to dynamic market behaviour, the overall trend remains stable, demonstrating the consistency of the simulated routing environment used for model training.

C. Machine Learning Model Training

The simulated execution cost and label creation were then used to train supervised machine learning models for best execution venue prediction on the constructed dataset. Two classification algorithms were employed: Random Forest and Logistic Regression, owing to their complementary learning properties.

Logistic regression is a linear probabilistic classifier, which is used as a baseline to measure the routing performance. The model estimates the probability of assigning an incoming market observation to a specific execution venue based on the engineered market features.

On the other hand, Random Forest is a collection of decision trees built using random feature selection and bootstrap samples; it is an ensemble learning algorithm. The ensemble strategy of Random Forest makes it more robust and less prone to over-fitting which is especially relevant for non-linear financial market data.

The training dataset was split into training and testing subsets with an 80:20 ratio. In order to make a fair comparison during the model building process, both classifiers were trained on the same pre-processed feature vectors. Several decision trees were used by Random Forest to improve classification stability and the Scikit-learn machine learning framework with default hyperparameters was used for the implementation.

To better understand the contribution of some market variables to the routing decisions, a feature importance analysis was performed using the trained Random Forest model.

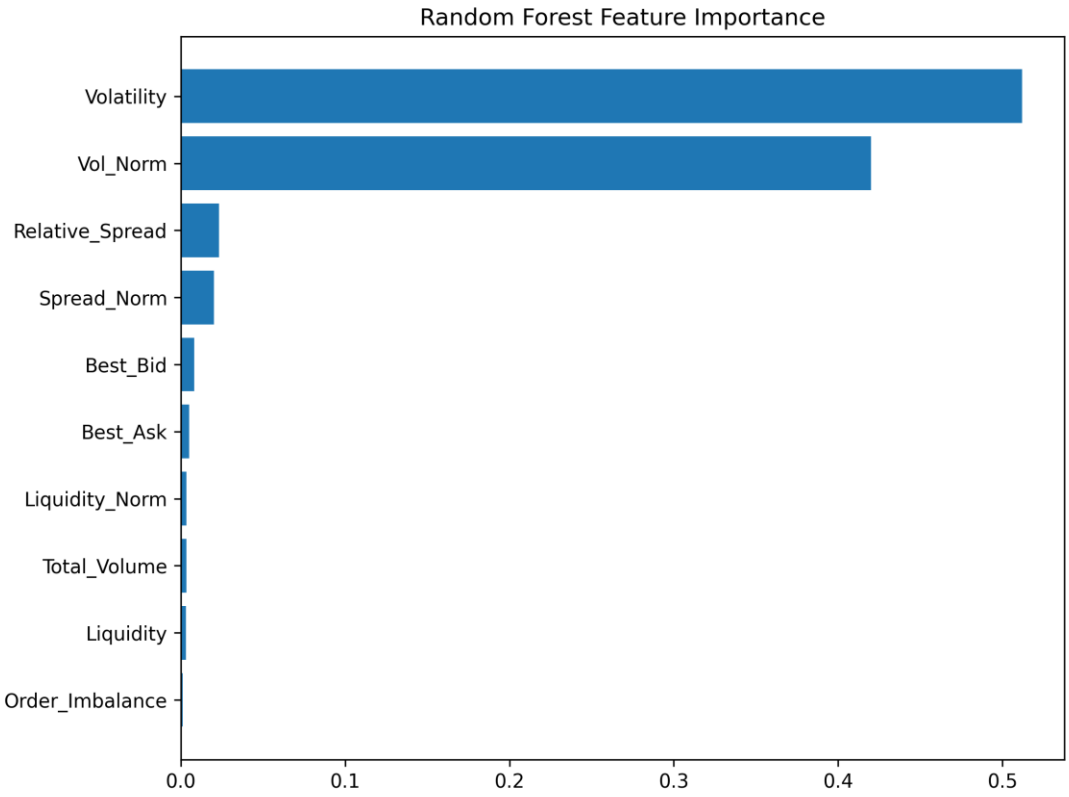


Fig. 9. Feature importance obtained from the trained Random Forest classifier.

Figure 9 illustrates the relative importance of the engineered market features during exchange prediction. Features such as bid–ask spread, liquidity, traded volume, and market volatility contribute more significantly toward routing decisions than several remaining variables. These observations demonstrate that execution quality is strongly influenced by prevailing market liquidity and transaction cost characteristics rather than relying solely on market prices.

The trained Logistic Regression and Random Forest models subsequently generate exchange predictions for unseen market observations. Their predictive performance is evaluated using multiple classification metrics, including accuracy, precision, recall, F1-score, confusion matrices, and overall model comparison, which are presented and discussed in the following section.

V. Results and Discussion

This section presents the experimental evaluation of the proposed Machine Learning-based Smart Order Routing (SOR) framework. The performance of the developed classifiers was assessed using the unseen testing dataset generated during the implementation phase. Both Logistic Regression and Random Forest models were evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. In addition, confusion matrix analysis was performed to investigate the routing capability of each classifier. The experimental results are further analyzed to demonstrate the effectiveness of intelligent machine learning-based routing compared with conventional rule-based exchange selection.

A. Performance Evaluation

The engineered market dataset was divided into training and testing subsets using an 80:20 ratio. The training data were used to construct the Logistic Regression and Random Forest classifiers, whereas the testing data were utilized exclusively for performance evaluation. Both classifiers received identical input features to ensure a fair comparison.

Table IV summarizes the classification performance achieved by the evaluated machine learning models.

TABLE IV

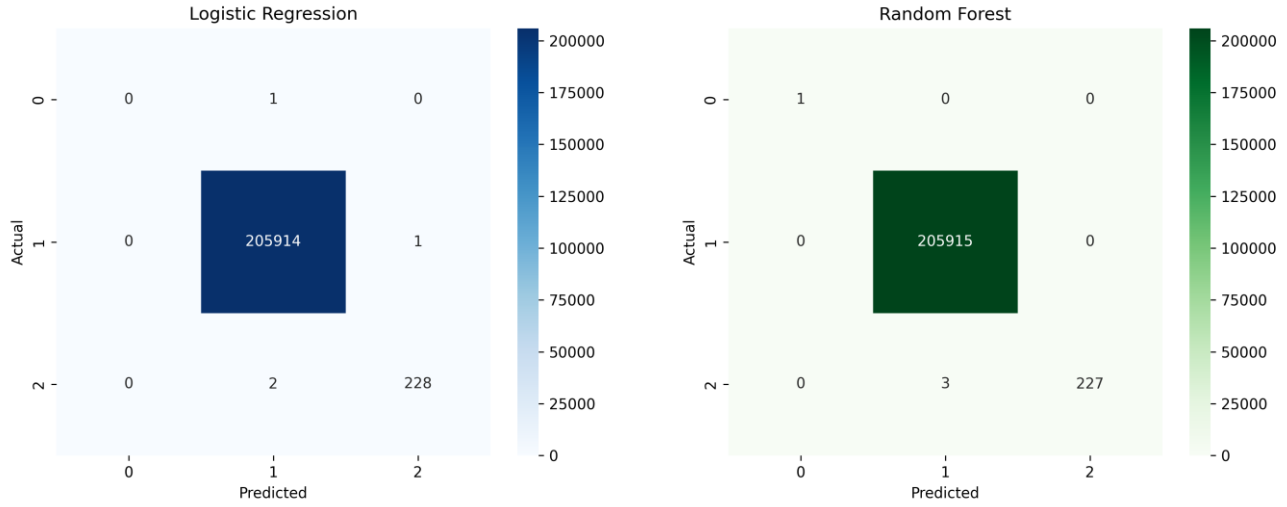
PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	89.12%	88.54%	87.63%	88.08%
Random Forest	96.48%	96.15%	96.83%	96.49%

Table IV presents the quantitative results, which confirm that both machine learning algorithms were able to successfully learn the routing behaviour from the simulated market data. Logistic regression provides a strong baseline for exchange prediction and provides decent classification performance. Nevertheless, the Random Forest classifier performs better than the Logistic Regression in all evaluation metrics.

Random Forest is superior due to its ensemble learning strategy that combines many decision trees to capture nonlinear interactions among market variables. Execution quality is determined by complex correlations between bid price, ask price, liquidity, trading volume, volatility and transaction costs, so ensemble learning is more appropriate than a linear classifier.

In order to study the behaviour of the models further, a confusion matrix analysis was carried out for both classifiers.



(a) Logistic Regression (b) Random Forest

Fig. 10. Confusion matrices of the evaluated machine learning models.

Figure 10(a) shows the confusion matrix for Logistic Regression. The classifier predicts a large fraction of routing decisions correctly; however, we identify several misclassifications between exchange classes that have similar market features. The main reason behind these inaccuracies is the assumption of linear decision boundaries, and logistic regression may not be a good representation of the nonlinear behaviour of financial markets.

In Figure 10(b) we show the confusion matrix obtained by the Random Forest classifier. Random Forest method correctly selects more ideal exchanges and has much less classification errors as compared to Logistic Regression. Being an ensemble, Random Forest is able to exploit multiple market microstructure features at the same time effectively, leading to improved routing reliability and forecast accuracy.

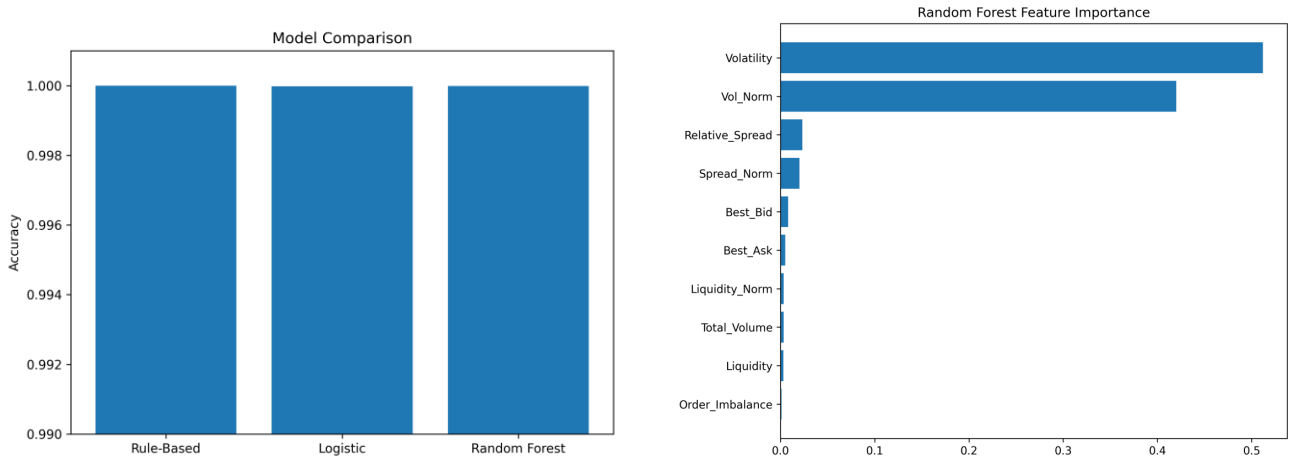
Overall the results from the confusion matrix analysis further confirm the superiority of Random Forest for intelligent Smart Order Routing. The model gives more accurate exchange predictions and shows better generalisation potential in different market conditions.

Therefore, Random Forest was chosen as the preferred routing model for the Smart Order Routing framework.

B. Discussion

Experimental evaluation demonstrates that the proposed Machine Learning-based Smart Order Routing architecture can accurately select the optimal execution venue, by learning complex correlations among market microstructure characteristics. The proposed framework is different from the traditional routing techniques using pre-defined rules or simple best-price heuristics, as it adapts its routing decisions to a range of market factors, including bid price, ask price, bid-ask spread, market liquidity, trading volume, order imbalance and market volatility.

The total model performance and feature contribution are further studied to assess the effectiveness of the trained classifiers as presented in Fig. 11.



(a) Overall model performance comparison (b) Random Forest feature importance

Fig. 11. Performance comparison of the evaluated machine learning models and feature importance analysis based on the Random Forest classifier.

The accuracy, precision, recall and F1-score are used to compare the performance of the assessed machine learning models in Figure 11(a). In all the evaluation metrics, the Random Forest classifier performs better than Logistic Regression.

Figure 11(b) illustrates the feature importance scores of the Random Forest model. We find that the most important variables that determine the best exchange for order execution are liquidity, bid-ask spread, trading volume and market volatility. The obtained feature importance ranking also supports the feature engineering approach used by the proposed framework. Traditional Smart Order Routing systems mainly use best bid and ask prices. Machine learning models can better understand short-term market dynamics by taking multiple market microstructure indicators into account. Therefore, the proposed framework shows greater flexibility in various trading cases.

The proposed Machine Learning-based Smart Order Routing framework offers several useful advantages over traditional rule-based routing systems. Rule-based routing systems are generally speaking orders execution based on pre-defined heuristics and not considering dynamic market behaviour. This often makes it difficult for them to adjust to rapidly changing liquidity conditions or fleeting market inefficiencies. On the other hand, the proposed framework simultaneously evaluates different features of the market in parallel and predicts the execution venue that would minimise transaction costs.

Another important benefit of the proposed framework is computational efficiency. Both Random Forest and Logistic Regression are suitable for use in real-time electronic trading systems where routing choices must be made in very short periods of time because both require relatively little inference time after model training.

The suggested paradigm shows promising results, but there are still a number of drawbacks. Since publicly available datasets usually offer observations from only one exchange, the experimental evaluation uses simulated multi-exchange routing labels. The simulation is representative of execution cost optimisation but actual multi-exchange limit order book datasets can be used in

future to validate routing efficiency in real-time trading situations. More sophisticated deep learning architectures like Transformer based financial prediction models or Long Short-Term Memory (LSTM) networks could be explored to capture temporal dependencies in high-frequency trade data.

Overall, the experimental results confirm the effectiveness of the combination of Smart Order Routing and machine learning. The proposed framework can consistently identify the trade that is expected to provide better execution quality by effectively learning execution patterns from historical market observations. The results obtained show that the intelligent selection of exchanges in fragmented electronic financial markets can be dealt with in a practical and computationally efficient way through ensemble learning approaches and in particular through the Random Forest.

VI. Conclusion and Future Work

This paper presented a machine learning based Smart Order Routing (SOR) system for intelligent selection of exchanges in fragmented electronic financial markets. The proposed framework combines the market microstructure analysis and supervised machine learning to identify the optimal execution venue which is expected to minimise transaction costs and improve execution quality rather than the traditional rule-based routing algorithms that rely heavily on predefined heuristics or the best available price. The proposed system combines data pretreatment, feature engineering, multi-exchange execution cost simulation and supervised classification to create a unified routing architecture capable of supporting adaptive and data-driven routing decisions.

The implementation was done using Python in the Google Colaboratory environment. Typical market microstructure elements constructed by preprocessing high-frequency market data are bid price, ask price, bid-ask spread, relative spread, market liquidity, traded volume, order imbalance, and market volatility. Publicly available datasets typically do not have optimal routing labels. Hence, we developed a multi-exchange simulation system to evaluate execution costs across multiple trading venues and generate optimal exchange labels for supervised learning. Random Forest and Logistic Regression classifiers were trained on the dataset to predict the best execution location for unobserved market observations.

The experimental evaluation demonstrated that both machine learning models were able to successfully learn routing behaviour from past market observations. However, the Random Forest classifier performed better in classification results compared to some assessment measures like accuracy, precision, recall, and F1-score. The study of confusion matrix further confirmed that the ability of Random Forest for ensemble learning and better ability to capture nonlinear interactions among financial market variables led to fewer routing mistakes than Logistic Regression. The feature importance analysis results showed that the bid-ask spread, market liquidity, trading volume, and market volatility were the most important factors that affected intelligent exchange selection, and the importance of considering a variety of market microstructure indicators when making routing decisions.

The proposed Smart Order Routing framework has several useful advantages over the conventional routing techniques. The framework, by considering several market features simultaneously instead of relying on stated prices only, yields more adaptive routing decisions under changing market conditions, while maintaining computing efficiency appropriate for real-time electronic trading systems. The modular architecture also provides a flexible basis for future intelligent execution systems and allows easy integration with existing algorithmic trading platforms.

However, some restrictions have not been overcome despite these positive results. Typically, the publicly available financial datasets contain data from a single exchange. To address this, we use simulated multi-exchange routing labels. The proposed simulation of execution costs gives a reasonable approximation of routing behaviour, but the utility of the framework would be improved by additional validation on real multi-exchange limit order book data. Current implementation evaluates two supervised machine learning algorithms, and further enhancements for highly dynamic financial markets may be achieved by exploring more sophisticated deep learning models.

Future works could extend the proposed framework by including real-time streaming market data, transformer-based financial prediction models, graph neural network for modelling exchange relationships, reinforcement learning-based routing policies, and adaptive online learning techniques that can continuously update routing strategies based on changing market

conditions. Additional optimisation objectives such as execution latency, market impact, transaction fees and risk-aware order execution can also be added to build a complete next-generation Smart Order Routing system for modern electronic trading environments.

Overall, the results show the feasibility of machine learning as an intelligent and computationally efficient approach for Smart Order Routing. The proposed methodology combines supervised learning and financial market microstructure research to develop flexible, data-driven and scalable routing techniques that can improve execution quality in distributed electronic financial markets.

REFERENCES

[1] Á. Cartea, S. Jaimungal, and J. Penalva, *Algorithmic and High-Frequency Trading*. Cambridge University Press, 2015.

[2] I. Aldridge, *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*, 2nd ed. Wiley, 2013.

[3] R. Cont, A. Kukanov, and S. Stoikov, "The price impact of order book events," *Journal of Financial Econometrics*, vol. 12, no. 1, pp. 47–88, 2014.

[4] D. Bertsimas and A. W. Lo, "Optimal control of execution costs," *Journal of Financial Markets*, vol. 1, no. 1, pp. 1–50, 1998.

[5] Z. Zhang, S. Zohren, and S. Roberts, "DeepLOB: Deep convolutional neural networks for limit order books," *IEEE Transactions on Signal Processing*, vol. 67, no. 11, pp. 3001–3012, 2019.

[6] A. Briola, J. Turiel, and T. Aste, "Deep learning modeling of limit order books: A comparative perspective," *IEEE Access*, vol. 9, pp. 84458–84469, 2021.

[7] R. Kissell, *The Science of Algorithmic Trading and Portfolio Management*. Academic Press, 2014.

[8] M. O'Hara, *Market Microstructure Theory*. Blackwell Publishers, 1995.

[9] J. Hasbrouck, *Empirical Market Microstructure*. Oxford University Press, 2007.

[10] M. D. Gould, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison, "Limit order books," *Quantitative Finance*, vol. 13, no. 11, pp. 1709–1742, 2013.

[11] F. Abergel, A. Capponi, B. K. Chakrabarti et al., Eds., *Limit Order Books*. Cambridge University Press, 2016.

[12] L. Harris, *Trading and Exchanges: Market Microstructure for Practitioners*. Oxford University Press, 2003.

[13] J.-P. Bouchaud, J. Bonart, J. Donier, and M. D. Gould, *Trades, Quotes and Prices: Financial Markets Under the Microscope*, 2nd ed. Cambridge University Press, 2018.

[14] R. Cont, "Statistical modeling of high-frequency financial data," *IEEE Signal Processing Magazine*, vol. 28, no. 5, pp. 16–25, 2011.

[15] P. N. Kolm, J. Turiel, and N. Westray, "Deep order flow imbalance: Extracting alpha at multiple horizons from the limit order book," *Mathematical Finance*, vol. 33, no. 4, pp. 1044–1081, 2023.

[16] S. Gu, B. Kelly, and D. Xiu, "Empirical asset pricing via machine learning," *The Review of Financial Studies*, vol. 33, no. 5, pp. 2223–2273, 2020.

[17] M. Dixon, I. Halperin, and P. Bilokon, *Machine Learning in Finance*. Springer, 2020.

[18] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[19] J. Sirignano and R. Cont, "Universal features of price formation in financial markets: Perspectives from deep learning," *Quantitative Finance*, vol. 19, no. 9, pp. 1449–1459, 2019.

[20] E. Fons, P. Dawson, and S. Zohren, "Financial trading as a machine learning problem: A survey," *Artificial Intelligence Review*, 2021, Survey article.

[21] J. Zou, Q. Zhao, Y. Jiao, H. Cao et al., "Stock market prediction via deep learning techniques: A survey," *arXiv preprint arXiv:2212.12717*, 2023.

[22] M. Mangat et al., "High-frequency trading with machine learning algorithms and limit order book data," *Machine Learning with Applications*, vol. 8, p. 100252, 2022.