*International Journal of Internet, Broadcasting and Communication Vol.15 No.3 203-210 (2023)*
*http://dx.doi.org/10.7236/IJIBC.2023.15.3.203*

IJIBC 23-3-25

# Development of a Stock Auto-Trading System using Condition-Search

Gyu-Sang Cho

*Dept. of Computer and Software, Dongyang University*
*cho@dyu.ac.kr*

## *Abstract*

*In this paper, we develope a stock trading system that automatically buy and sell stocks in Kiwoom Securities' HTS system. The system is made by using Kiwoom Open API+ with the Python programming language. A trading strategy is based on an enhanced system query method called a Condition-Search. The Condition-Search script is edited in Kiwoom Hero 4 HTS and the script is stored in the Kiwoom server. The Condition-Search script has the advantage of being easy to change the trading strategy because it can be modified and changed as needed. In the HTS system, up to ten Condition-Search scripts are supported, so it is possible to apply various trading methods. But there are some restrictions on transactions and Condition-Search in Kiwoom Open API+. To avoid one problem that has transaction number and frequency are restricted, a method of adjusting the time interval between transactions is applied and the other problem that do not support a threading technique is solved by an IPC(Inter-Process Communication) with multiple login IDs.*

***Keywords:*** Stock Auto-Trading System, Condition-Search, Kiwoom Securities, Open API+

## 1. Introduction

Since the spread of the Internet, securities companies have developed a HTS(Home Trading System) system to provide a new platform for stock trading, and with the spread of smartphones, they are providing the MTS(Mobile Trading System) system for use in a mobile environment. Currently, most people who trade stocks use HTS or MTS stock trading system. In addition, each securities company provides API(Application Program Interface) functions so that customers can trade directly through customer-generated programs. Using these API functions, it is possible to implement the function of trading based on the price predicted by the artificial intelligence or the automatic transaction by the trading rule written by the user [1][2][3]. For quant investors, a system that can automatically trade by a user-created program must be used to apply information not provided by securities companies [4].

Several securities companies provide their own API systems. The systems provided by each securities company are listed as follows: Kiwoom Securities: Open API+ (OCX) only for Windows [5], Daeshin Securities: CYBOS Plus(COM) compatible for mobile Android and iOS [6], eBest Investment Securities: xing API compatible for PC and mobile [7]. Yuanta Securities: Open API (DLL, COM), NH Futures: API (DLL,

Manuscript Received: July. 16, 2023 / Revised: July. 21, 2023 / Accepted: July. 26, 2023
Corresponding Author: cho@dyu.ac.kr
Tel: +82-54-630-1119, Fax: +82-54-630-1179
Professor, Department of Computer & Software, Dongyang University, Korea

OCX), NH Investment & Securities: QV Open API (DLL), Hana Financial Investment: 1Q Pro (OCX), Daehan Investment & Securities: eFriend Expert Open API (OCX), Eugene Investment & Futures: API (DLL, OCX), and so on. There are not many cases where APIs are appropriately provided for easy access to open source. Three companies, Kiwoom Securities [5], Daishin Securities [6], and eBest Securities [7], are providing appropriate services [8].

In this paper, we develop a system that can automatically trade stocks using Open API+ provided by Kiwoom Securities. As a trading strategy, we implement an automatic trade strategy by using the Condition-Search script. There are restrictions on transactions of Open API+ of Kiwoom Securities, so we apply ways of taking time intervals between transactions and utilizing an IPC technique with multiple login IDs to avoid a threading technique problem.

## 2. Features of Kiwoom Open API+

### 2.1 System files

Kiwoom open api+ is a service that connects investment strategies produced by customers to modules provided by Kiwoom Securities that serve market price inquiries, balance inquiries, order etc. Kiwoom Open API+ supports OCX control considering the extensibility of program development, which depending on the user's convenience, programs can be produced with VB, Excel, web-based, MFC, Python PyQt5, etc. Data requests and receptions are performed by performing the transaction service name and calling the API function through OCX [5].

The supplied system library files are as follows: 1) KHOpenAPI.ocx: OpenApi Execution module, 2) OPCommApi.dll: communication module, 3) OPComms.dll: TCP/IP connection module to serve, 4) OPSecurity.dll: communication encryption module, 5) librsadlx.dll nsldap32v11.dll, 6) ZIPDLL.DLL: compression module.

The supplied program data files are as follows:1) OPTxxxxx.enc 2) OPWxxxxx.enc : data files for a communication(about 100 files). 3) jongacc.dat: account related file, 4) OPComms.ini : communication attribute related data file.

### 2.2 Query Restrictions of Kiwoom Open API+

Kiwoom server transaction query has the following restrictions: 1) If 5 queries of a second in successive 1 times: 17 seconds blocked 2) If 5 queries of a second in successive 5 times: 90 seconds blocked 3) If 5 queries per a second in successive 10 times: 180 seconds blocked 4) Un-official successive query restriction : 1,000 queries in an hour [9].

Condition-Search query restrictions are as follows: 1) 5 queries of a second by a Condition-Search, 2) 1 query in a minute for the identical Condition-Search, 3) Up to 10 types of Condition-Search are possible 4) If the Condition-Search result exceeds 100 items, the condition cannot receive a real-time Condition-Search signal.

Usage of a threading function restriction: because Open API+ does not support a threading in functions and events, calling Open API+ functions or events from the thread cannot be guaranteed. In fact, when a transaction is called multiple times using a thread function, the system is found to be shut down frequently. In addition, when a Condition-Search is performed and information is received by performing a transaction within a loop, system often shut down.

Avoid restrictions: to avoid the problem of limiting the number of such queries, we maintain the interval between transactions for more than 0.4 seconds. In addition, we adopt a method of counting the total number of queries and waiting until a set time elapses when the maximum number is reached.

In this paper, we avoid a threading function restriction problem by applying an IPC method between the two processes using two IDs as depicted in Figure 1.

## 3. Implementation of an Auto-Trading System using Condition-Search of Kiwoom Open API+

### 3.1 Overview

Stock auto-trading system using a Condition-Search for Kiwoom Securities is composed of 3 parts; a transaction request part roles to query stock information, a stock trading part to buy and sell stocks, and the Condition-Search part to manage trading strategy as depicted in Figure 1.

The advantage of auto-trading is that it is possible to carry out several strategies simultaneously. Setting up several strategies and using multiple computers or cloud servers greatly reduces the time required to develop a profit model. The Condition-Search is a service of Kiwoom Securities that searches for stocks corresponding to the conditions specified by the user. It provides functions such as market analysis, technical analysis, pattern analysis, financial analysis, ranking analysis, and provides pre-specified Condition-Search script under the name “Kiwoom Condition”. This feature allows users to program complex queries, allowing them to create and use very convenient features for trading. In addition, using Condition-Search and using the functions of HTS can greatly reduce the amount of code in automatic system development. Condition setting is done by the user through the Kiwoom HTS system, and real-time Condition-Search items are transmitted through the API.

In this study, we implement a system called “auto-trading system for Kiwoom Securities using Open API+” that can execute an auto-trading strategy, which Kiwoom Securities’ Condition-Search is used to search for trading stock items, and judging buy and sell.

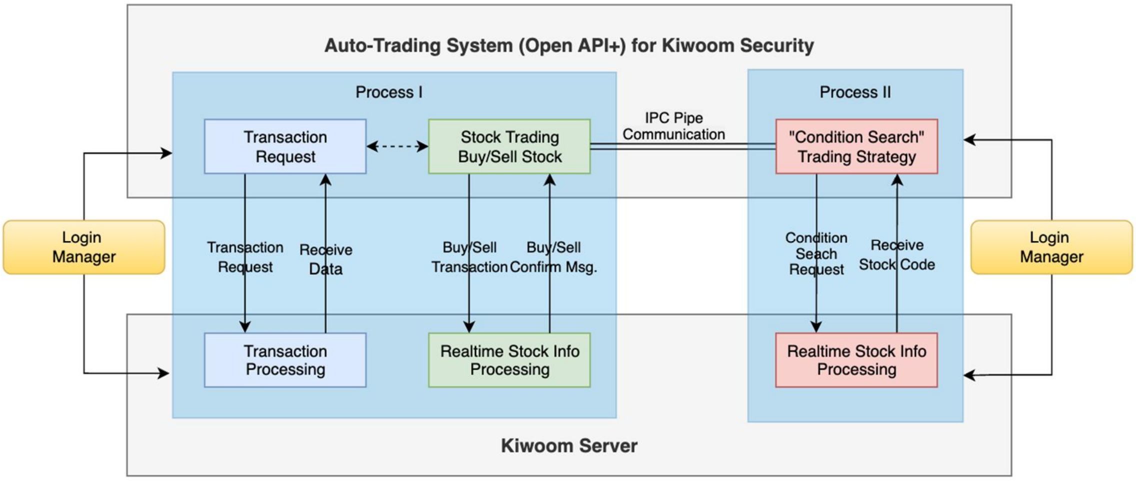


**Figure 1. Overall configuration of stock auto-trading using Condition-Search for Kiwoom Securities**

### 3.2 Overall System Configuration

Stock auto-trading system using Condition-Search for Kiwoom Securities is composed of 3 parts; a transaction request part roles to query stock information, a stock trading part to buy and sell stocks, and Condition-Search part to manage trading strategy as depicted in Figure 1.

The entire system is driven by two processes; one is a process for Condition-Search and the other is a process for a stock trading transaction and for stock information search. The reason for separating the system into these two processes is that the Kiwoom API does not support a threading, which would cause a failure in performing real-time Condition-Search, so this configuration is made to avoid this.

To do this, two IDs are used for the Kiwoom Securities. To communicate between each ID, Process I installs the Python library "from multiprocessing.connection import Listener" and waits to receive the stock code sent by Process II with the recv() function. The client function is installed on the Process II side, "from multiprocessing.connection import Client", and use the send() function to send the Condition-Search stock code to the Process I side. A two-way communication is used to confirm the send and receive of both sides [10].

Process I side

```
from multiprocessing.connection import Listener
address = ('localhost', 12345)
listener = Listener(address, authkey=b'hello')
conn =listener.accept()
while True:
    print('waiting condition msg...')
    msg = conn.recv()
    conn.send(msg+'received')
```

Process II side

```
from multiprocessing.connection import Client
address=('localhost', 12345)
conn=Client(address, authkey=b'hello')
conn.send(msg)
```

Process I consist of two functions: a transaction to obtain the current price of a stock and a buy/sell function. The transaction to obtain the present price is used to obtain the present price of the code after receiving the code of the item Condition-Search in Process II. After performing the transaction, real-time stock information is automatically received, which is used to continuously update the current price until the stock is sold. If any of the conditions searched stocks meet the conditions set in the strategy, they will quickly place an order and buy it. After the buy order, real-time information related to the item is automatically received. This real-time information is used as important data to update the current information of stocks.

### 3.3 Description on Transaction Request

Figure 2 shows the procedure for querying data in the Kiwoom Securities HTS system. Chart information such as the current price, asking price, daily chart, weekly chart and monthly chart of stocks for stock trading can be received by requesting a transaction from the server. There are some essential functions required for transactions. SetInputValue() is a function used to enter information necessary for a transaction request immediately before a transaction request is made. For example, stock code, account number, transaction number, etc. are used as arguments for this function. SendOrder() is used as a transaction command to transmit stock trade orders to the server, which has nine arguments, and inputs information such as buy/sell information, stock code, order quantity, and order unit price etc. CommGetData () is a common function used to receive data by transaction request, but each function uses a dedicated function for its purpose instead of this function; GetCommData() is for a query information request, GetCommRealData() is for a real-time information request, and GetChejanData() is for a buy/sell confirm request. Functions such as OnReceiveTrData(), OnReceiveRealData(), OnReceiveMsg(), OnReceiveChejanData(), OnEventConnect(), OnReceiveRealCondition(), and OnReceiveTrCondition() are used to receive transaction inquiries and real-time events.

### 3.4 Description on Condition-Search

Condition-Search is a function of finding stocks that meet certain conditions, and this function allows a wide variety of settings. Market analysis, technical analysis, pattern analysis, financial analysis, ranking analysis are provided for a pre-specified Condition-Search script. The Condition-Search is produced by the user in the Kiwoom Securities HTS system, and the script is stored in the Kiwoom server. Open API + functions are provided to use it. Figure 3. shows the procedures and functions of the Condition-Search.

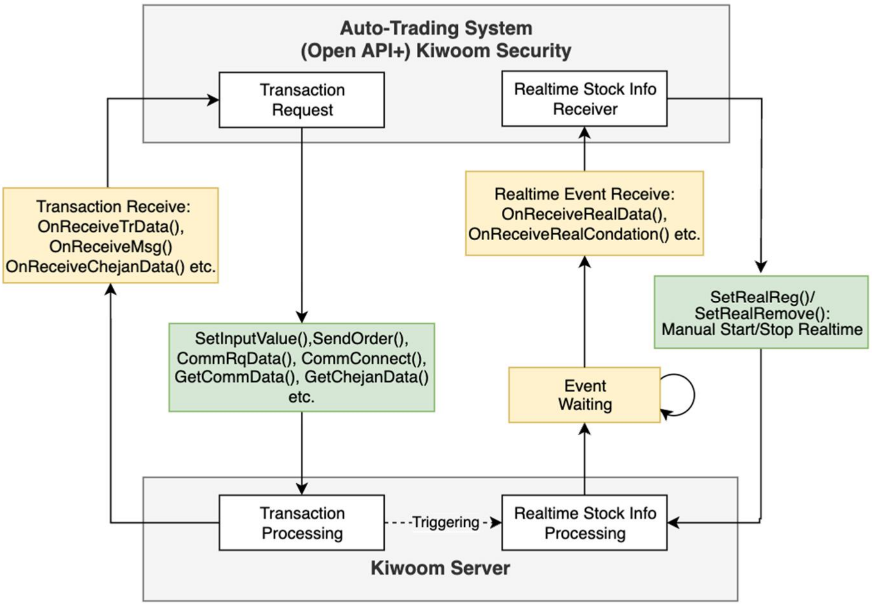

Auto-Trading System
(Open API+) Kiwoom Security
Transaction Request
Realtime Stock Info Receiver
Transaction Receive: OnReceiveTrData(), OnReceiveMsg() OnReceiveChejanData() etc.
Realtime Event Receive: OnReceiveRealData(), OnReceiveRealCondation() etc.
SetRealReg()/ SetRealRemove(): Manual Start/Stop Realtime
SetInputValue(),SendOrder(), CommRqData(), CommConnect(), GetCommData(), GetChejanData() etc.
Event Waiting
Transaction Processing
Triggering
Realtime Stock Info Processing
Kiwoom Server

**Figure 2. Procedure of transaction for data query and real-time event reception**

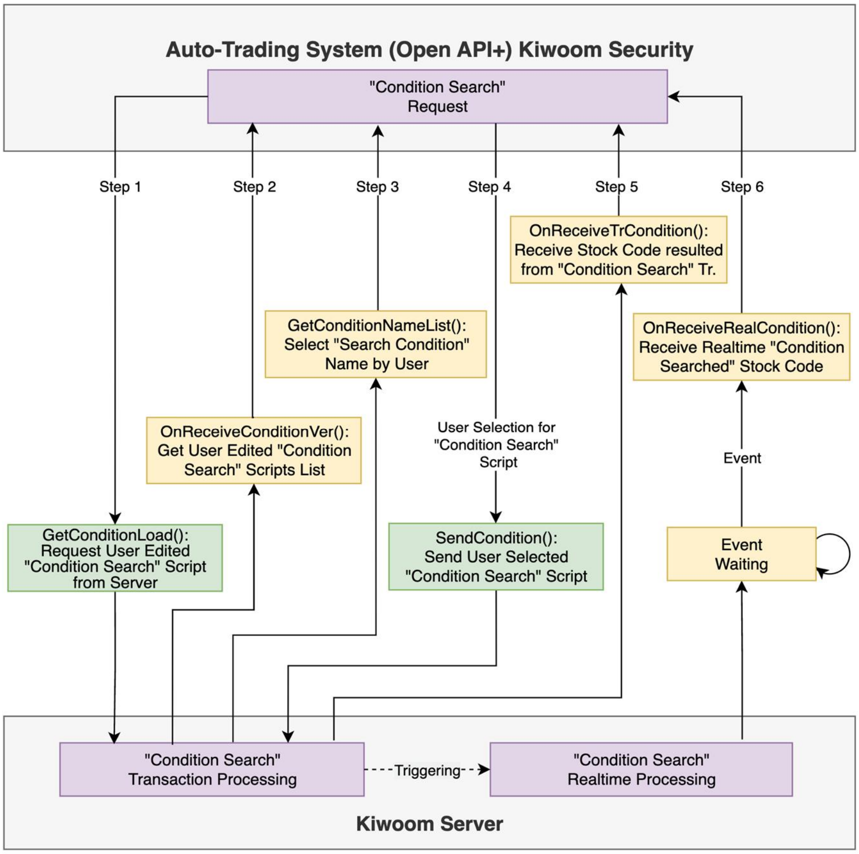
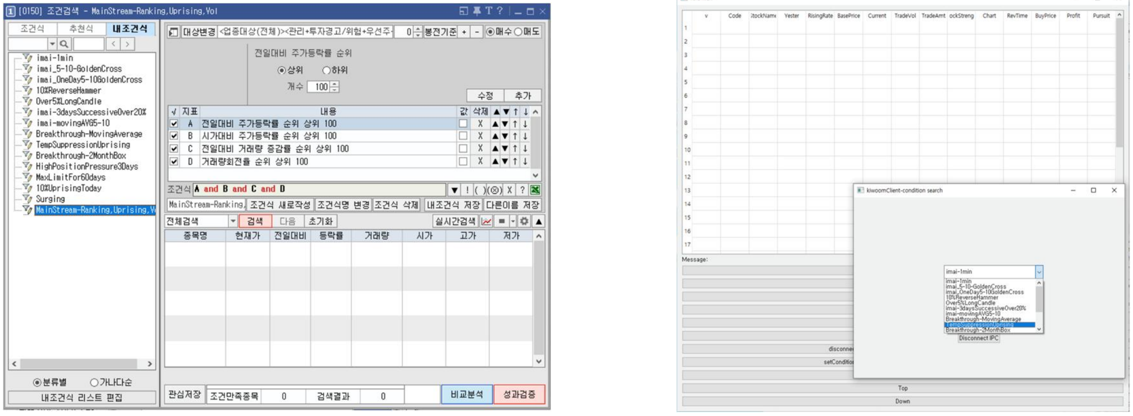

Auto-Trading System (Open API+) Kiwoom Security
"Condition Search" Request
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
OnReceiveTrCondition(): Receive Stock Code resulted from "Condition Search" Tr.
GetConditionNameList(): Select "Search Condition" Name by User
OnReceiveRealCondition(): Receive Realtime "Condition Searched" Stock Code
OnReceiveConditionVer(): Get User Edited "Condition Search" Scripts List
User Selection for "Condition Search" Script
Event
GetConditionLoad(): Request User Edited "Condition Search" Script from Server
SendCondition(): Send User Selected "Condition Search" Script
Event Waiting
"Condition Search" Transaction Processing
Triggering
"Condition Search" Realtime Processing
Kiwoom Server

**Figure 3. Procedure of Condition-Search for an auto-trading**

First, it is necessary to obtain a condition search script stored in the server. In Step 1, the GetConditionLoad() function retrieves the user condition expression from the server and temporarily stores it locally as a file. In Step 2, OnReceiveConditionVer() can check the success or failure of locally stored Condition-Search script and messages as an event function. In Step 3, GetConditionNameList() retrieves the list of conditional names from the temporarily saved user “Condition Search” file. In Step 4, SendCondition() selects the Condition-Search to be performed from the list by the user. In Step5, OnReceiveTrCondition() receives a searched stock list from the server. After a transaction phase of Condition-Search is completed, it triggers real-time Condition-Search mode, which real-time event happens continuously and Condition-Search one stock code is received in OnReceiveRealCondition() function at every event. The received stock code send to the Buy/Sell stock routine as depicted in Figure 1. Once a Condition-Search transaction is completed, it automatically enters the real-time event mode, which can be manipulated by SetRealReg() and SetRealRemove() by the user to resume or cease the real-time stock information receiving by the user.

Figure 4(a) shows the screen shot of the Condition-Search edit window, and Figure 4(b) show the screen shot of the proposed system window of a sending Condition-Search script.



[0150] 조건검색 - MainStream-Ranking,Uprising,Vol

- imai-1min
- imai_5-10-GoldenCross
- imai_OneDay5-10GoldenCross
- 10%ReverseHammer
- Over5%LongCandle
- imai-3daysSuccessiveOver20%
- imai-movingAVG5-10
- Breakthrough-MovingAverage
- TempSuppressionUprising
- Breakthrough-2MonthBox
- HighPositionPressure3Days
- MaxLimitFor60days
- 10%UprisingToday
- Surging
- MainStream-Ranking,Uprising,V

대상변경 <업종대상(전체)><관리+투자경고/위험+우선주- 0 봉전기준 ◉매수 ○매도

전일대비 주가등락률 순위 ◉상위 ○하위 개수 100

| √ | 지표 | 내용 | 값 | 삭제 |
|---|---|---|---|---|
| ☑ | A | 전일대비 주가등락률 순위 상위 100 | □ | X |
| ☑ | B | 시가대비 주가등락률 순위 상위 100 | □ | X |
| ☑ | C | 전일대비 거래량 증감률 순위 상위 100 | □ | X |
| ☑ | D | 거래량회전율 순위 상위 100 | □ | X |

조건식 A and B and C and D

| 종목명 | 현재가 | 전일대비 | 등락률 | 거래량 | 시가 | 고가 | 저가 |
|---|---|---|---|---|---|---|---|

조건만족종목 0 검색결과 0

(a) Condition-Search script edit window

Tera Rich

| | v | Code | StockName | Yester | RisingRate | BasePrice | Current | TradeVol | TradeAmt | ockStreng | Chart | RevTime | BuyPrice | Profit | Pursuit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | |

Message:

kiwoomClient-condition search

imai-1min

- imai-1min
- imai_5-10-GoldenCross
- imai_OneDay5-10GoldenCross
- 10%ReverseHammer
- Over5%LongCandle
- imai-3daysSuccessiveOver20%
- imai-movingAVG5-10
- Breakthrough-MovingAverage
- TempSuppressionUprising
- Breakthrough-2MonthBox

(b) Screen shot of the trading system

**Figure 4. Screen shots of Condition-Search and developed system for an auto-trading**

### 3.5 Development Environments

- OS: Windows 11
- Run Environment: Anaconda 32bit Virtual Environment
- Application Type: Python Script + PyQt5 GUI Application
- Programming Language: Python v3.9
- Library: PyQt5 v5.15.9
- Development Tools: PyCharm 2023.2 (Community Edition)

## 5. Conclusions

In this paper, we developed a stock trading system to automatically trade stocks at Kiwoom with the Python program. The method for automatic trading used Kiwoom Securities' stock search system, called Condition-Search. To buy and sell stock, we implemented the system by the Condition-Search script, the stocks were automatically traded by setting a certain range of profits. The Condition-Search script has the advantage of being easy to change the trading strategy because it can be modified and changed as needed. In the Kiwoom HTS system, up to ten Condition-Search scripts are supported, so it is possible to apply various trading methods. In addition, various cases were listed on the constraints of Kiwoom Securities' Open API+, and to avoid the problem, we solved a method of adjusting the time interval between transactions and a method problems that do not support a threading technique by an IPC(Inter-Process Communication) using a multi-login.

## Acknowledgement

"This study was supported by a grant from Dong Yang University in 2022"

## References

[1] H. Cho and H. J. Shin, "Trading Strategies Using Reinforcement Learning", Journal of the Korea Academia-Industrial Cooperation Society, Vol. 22, No. 1 pp. 123-130, 2021. DOI:https://doi.org/10.5762/KAIS.2021.22.1.123

[2] E. Yi and W. B. Lee, "A Study on Stock Trading Method based on Volatility Breakout Strategy using a Deep Neural Network", The Journal of the Korea Contents Association, '22 Vol. 22 No. 3, 2022. DOI: https://doi.org/10.5392/JKCA.2022.22.03.081

[3] B. H. Cho, “Analysis and Design of Stock Item Buy/Sell Recommend System using AI Machine Learning Technology”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 21, No. 4, pp.103-108, 2021.
DOI: https://doi.org/10.7236/JIIBC.2021.21.4.103

[4] S. W. Kim and H. S. Choi, “Estimation of GARCH Models and Performance Analysis of Volatility Trading System using Support Vector Regression”, Journal of Intelligence and Information Systems, Vol. 23, Issue 2, pp.107-122. 2017
DOI: https://doi.org/10.13088/jiis.2017.23.107

[5] Kiwoom Open API+ Developer’s Guide,
*https://download.kiwoom.com/web/openapi/kiwoom_openapi_plus_devguide_ver_1.1.pdf*

[6] *CYBOS Plus, Daishin Securities,*
*https://money2.daishin.com/E5/WTS/Customer/GuideTrading/DW_CybosPlus_Page.aspx?p=8812&v=8632&m=9508*

[7] *Open API Guide, eBEST INVESTMENT & SECURITIES, https://openapi.ebestsec.co.kr/intro*

[8] AiDA Lab, Open API of Securities, *https://aidalab.tistory.com/100*

[9] Information guide on restrictions of transactions(rev.), Kiwoom Open API+
https://www.kiwoom.com/h/common/bbs/VBbsNoticeNWOFZView

[10] Python Documentation, “multiprocessing — Process-based parallelism”,
*https://docs.python.org/3/library/multiprocessing.html*