A Study on Comparison of Response Time using Open API of Daishin Securities Co. and eBest Investment and Securities Co.

Gui Yeol Ryu

Professor, Department of Software, Seokyeong University, Seoul 02713, Korea
e-mail: gyryu@skuniv.ac.kr

Abstract

Securities and investment services have and use large data. Investors started to invest through their own analysis methods. There are 22 major securities and investment companies in Korea and only 6 companies support open API. Python is effective for requesting and receiving, analyzing text data from open API. Daishin Securities Co. is the only open API that officially supports Python, and eBest Investment & Securities Co. unofficially supports Python. There are two important differences between CYBOS plus of Daishin Securities Co. and xingAPI of eBest Investment & Securities Co. First, we must log in to CYBOS plus to access the server of Daishin Securities Co. And the python program does not require a logon. However, to receive data using xingAPI, users log on in an individual Python program. Second, CYBOS plus receives data in a Request/Reply method, and zingAPI receives data through events. It can be thought that these points will show a difference in response time.

Response time is important to users who use open APIs. Data were measured from August 5, 2021, to February 3, 2022. For each measurement, 15 repeated measurements were taken to obtain 420 measurements. To increase the accuracy of the study, both APIs were measured alternately under same conditions. A paired t-test was performed to test the hypothesis that the null hypothesis is there was no difference in means. The p-value is 0.2961, we do not reject null hypothesis. Therefore, we can see that there is no significant difference between means. From the boxplot, we can see that the distribution of the response time of eBest is more spread out than that of Cybos, and the position of the center is slightly lower. CYBOS plus has no restrictions on Python programming, but xingAPI has some limits because it indirectly supports Python programming. For example, there is a limit to receiving more than one current price.

Keywords: Open API, Python, Securities and investment services, Stock data, Response Time

1. Introduction

The McKinsey Global Institute verifies that securities and investment services sector used the most data [1]. The home trading system (HTS) supported by DBMS is a system of users for the data of Securities and

investment companies. Many investors started to invest through their own analysis methods [2]. The data in the HTS is not suitable for users having their own analysis methods. Therefore, securities and investment companies should provide stock-related data suitable for data analysis quickly. They adopt Open Application Programming Interface (API) as the solution [3]. Investors can store vary large data obtained by open API. And big data is the basis for machine learning and artificial intelligence [4, 5].

There are 22 major securities and investment companies in Korea such as SK Securities Co., Hanwha Investment & Securities Co., Shinyoung Securities Co., Bookook Securities Co., Hanyang Securities Co. Samsung Securities, Kyobo Securities Co., KB Securities Co., Hyundai Motor Securities, Daishin Securities Co., Hi Investment & Securities Co., NH Investment & Securities Co., Kiwoom Securities Co., eBest Investment & Securities Co., Mirae Asset Daewoo Co., Yuanta Securities Co., Eugene Investment Co., Korea Investment & Securities Co., KTB Investment & Securities Co., Hana Financial Investment Co., IBK Securities Co., and Cape Investment & Securities Co. They support Home Trading System (HTS) which allows individuals to trade stocks anytime, anywhere by online. But only 6 companies support open API for big data. Table 1 shows the HTS and open API support status of 22 securities and investment companies in Korea [6].

Table 1. Open API supported by 22 major securities and investment companies in Korea

Securities and investment services	HTS	Open API
SK Securities Co.	Yes	No
Hanwha Investment & Securities Co.	Yes	No
Shinyoung Securities Co.	Yes	No
Bookook Securities Co.	Yes	No
Hanyang Securities Co.	Yes	No
Samsung Securities	Yes	No
Kyobo Securities Co.	Yes	No
KB Securities Co.	Yes	No
Hyundai Motor Securities	Yes	No
Daishin Securities Co.	Yes	Yes
Hi Investment & Securities Co.	Yes	No
NH Investment & Securities Co.	Yes	No
Kiwoom Securities Co.	Yes	Yes
eBest Investment & Securities Co.	Yes	Yes
Mirae Asset Daewoo Co.	Yes	No
Yuanta Securities Co.	Yes	Yes
Eugene Investment Co.	Yes	No
Korea Investment & Securities Co.	Yes	No
KTB Investment & Securities Co.	Yes	No
Hana Financial Investment Co.	Yes	Yes
IBK Securities Co.	Yes	Yes
Cape Investment & Securities Co.	Yes	No
Total	22	6

We receive data in text format using the open API. Python is effective and convenient for requesting and receiving, analyzing text data. Ryu investigated the status of Korean securities companies' open API and compared how to receive stock data through open API using Python. It is founded that Daishin Securities Co.

is the only open API that officially supports Python, and eBest Investment & Securities Co. unofficially supports Python. Table 2 shows software supported by open API [7].

Table 2. Software supported by open API

Securities and investment services	Supported Software	Supporting Python officially
Daishin Securities Co.	Microsoft Visual Studio (VB, C#, C++, .Net etc.), Microsoft Office (Word, Excel, Powerpoint, Access etc.), Microsoft Internet Explorer Dynamic HTML, Python, and Borland Delphi	Yes
Kiwoom Securities Co.	VB, C/C++, C#, Excel, and Delphi	No
eBest Investment & Securities Co.	VB, C/C++, C#, and Delphi	No
Yuanta Securities Co.	VB, C/C++, C#, and Delphi	No
Hana Financial Investment Co.	MFC, C#, VB, Delphi, Excel	No
IBK Securities Co.	Microsoft Visual Studio (C++, VB, J++, etc.), Microsoft Office (Word, Excel, Powerpoint, Access etc.), Microsoft Internet Explorer Dynamic HTML, Borland Delphi	No

Daishin Securities Co. provides CYBOS plus for open API. eBest Investment & Securities Co. provides xingAPI for open API. There are two important differences between CYBOS plus and xingAPI. First, we must log in to CYBOS plus to access the server of Daishin Securities Co. And the python program does not require a logon. However, to receive data using xingAPI, users log on in an individual Python program every time when they connect. Second, CYBOS plus receives data in a Request/Reply method, and zingAPI receives data through events. It is thought that these points will lead to a difference in response time between two open API. Response time is important to users who use open APIs. Therefore, the study on the difference in response time would have an important meaning. In this paper, we will study the difference in response time between two open APIs. Data were measured from August 5, 2021, to February 3, 2022. For each measurement, 15 repeated measurements were taken to obtain 420 measurements. To increase the accuracy of the study, both APIs were measured alternately under same condition.

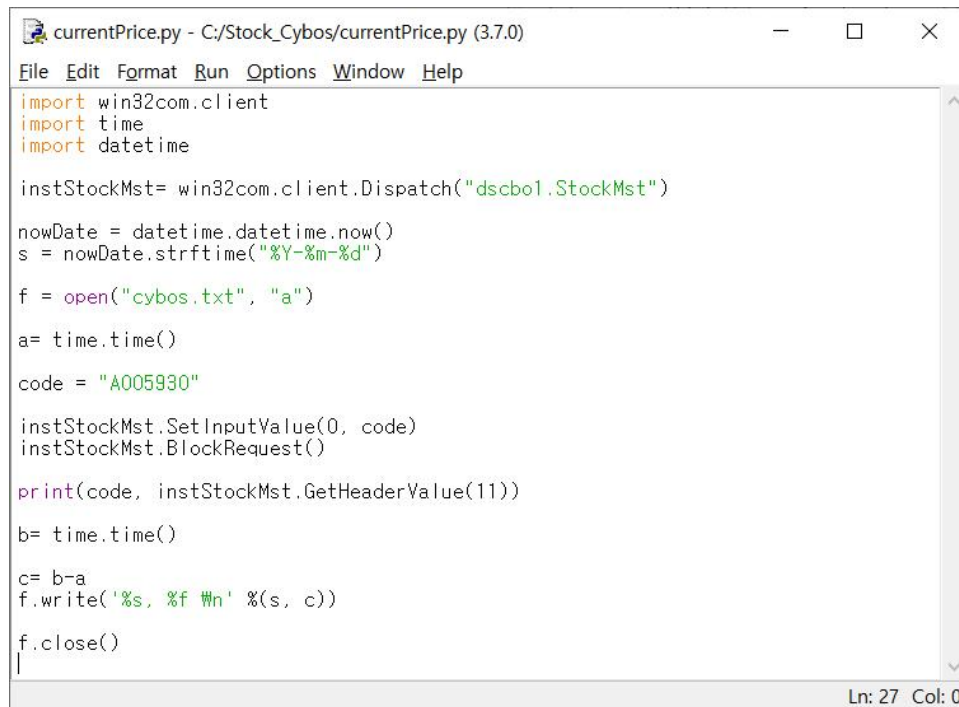## 2. Comparison of Response Time of Two Open APIs using Python

2.1 Response Time of Current Price by open API of Daishin Securities Co.

The website of Daishin Securities Co. explains how to access open API. CYBOS plus, the name of open API must be run in the background to access the server. We code in Python and request CYBOS plus to receive data. We can compute response time. The login information is not required in the Python code, because CYBOS plus checks the login permission.

Figure 1 shows the CYBOS plus program for computing response time of a current price of the code "A005935", which is the code of SAMSUNG Electronics Co., Ltd. Figure 1 refers to Jo [8]. Lines 1 imports the win32com.client module. Line 2 imports the time module. Line 3 import the datetime module for managing time. Line 5 creates an instance of the StockMst class by using the Dispach method. Line 7 generates the instance of time. Line 8 generates a string of time. Line 10 opens the cybos.txt that records the response time. Line 12 creates variable "a" for recording the time before receiving a response. Line 14

creates a code variable for SAMSUNG Electronics Co., Ltd. Line 16 sets input variables. Line 17 requests the current value. Line 19 prints the code and the current price. Line 21 create variable "b" for recording the time after receiving a response. Line 23 creates variable "c" for computing response time. Line 24 saves the code and the response time to file. Line 26 closes file.

Table 3 is the result of response time of Daishin Securities Co. The mean is 1.93 seconds, the median is 1.89 seconds, and the standard deviation is 0.19 seconds. Since the difference between the mean and the median is not large and the standard deviation is small, we could see that the distribution of response time is dense with the center． The maximum value is 3.45 seconds，and the minimum value is 1.61 seconds, so the range is 1.84 seconds, which is not large.

```python
import win32com.client
import time
import datetime

instStockMst= win32com.client.Dispatch("dscbo1.StockMst")

nowDate = datetime.datetime.now()
s = nowDate.strftime("%Y-%m-%d")

f = open("cybos.txt", "a")

a= time.time()

code = "A005930"

instStockMst.SetInputValue(0, code)
instStockMst.BlockRequest()

print(code, instStockMst.GetHeaderValue(11))

b= time.time()

c= b-a
f.write('%s, %f \n' %(s, c))

f.close()
```

Figure 1. Python program of Daishin Securities Co. for response time of current price

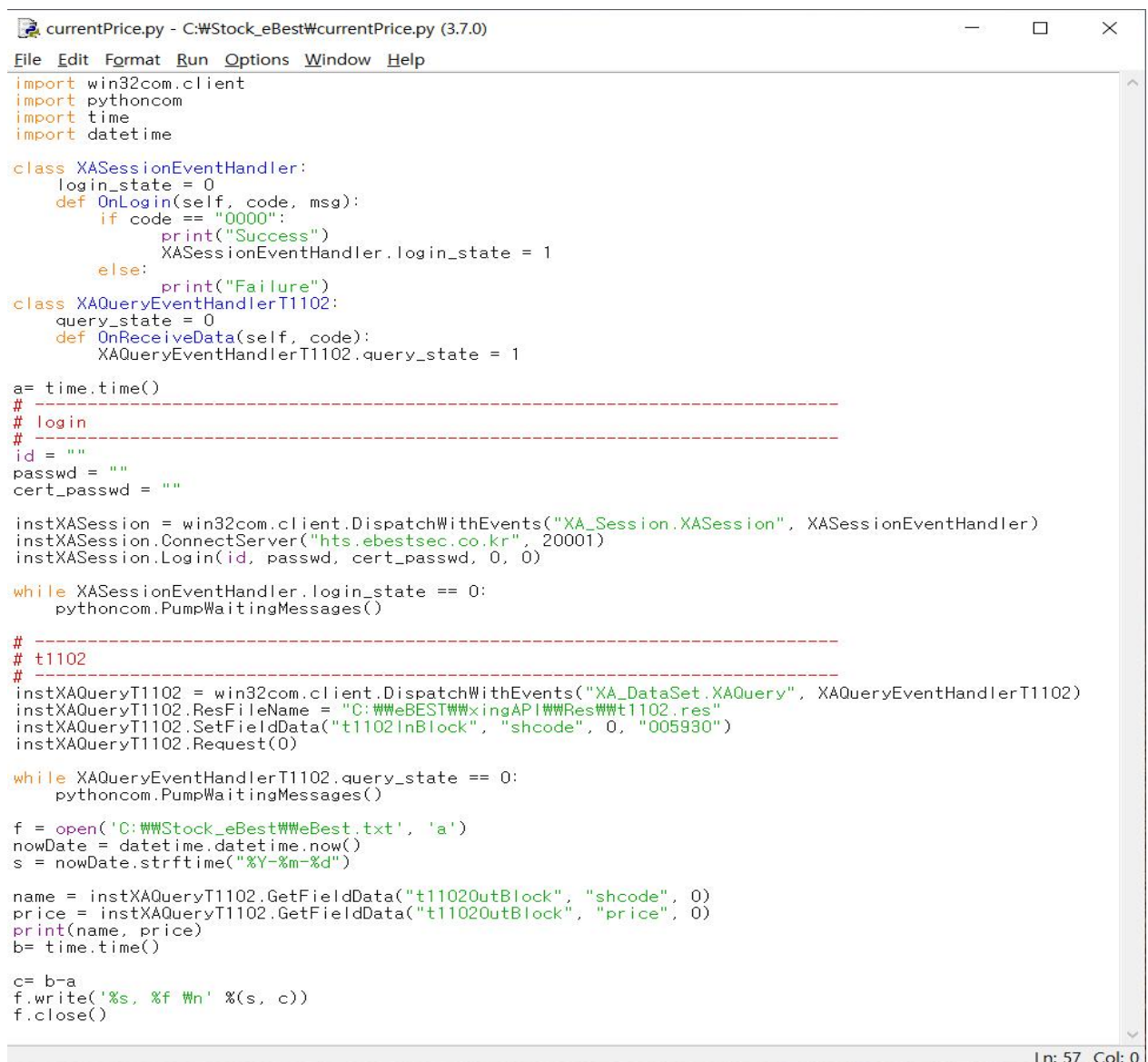**Table 3. Result of Response Time of Daishin Securities Co.**

Number of experiments	Mean (second)	Standard Deviation (second)	Median (second)	Minimum (second)	Maximum (second)
420	1.93	0.19	1.89	1.61	3.45

2.2 Response Time of Current Price by open API of eBest Investment & Securities Co.

Figure 2 shows the xingAPI program for computing response time of a current price of the code "005935", which is the code of SAMSUNG Electronics Co., Ltd. Figure 2 refers to Jo [8]. Line 1 imports win32com.client module. Line 2 imports pythoncom module. Line 3 imports time module. Line 4 imports datetime module. Lines from 6 to 13 define XASessionEventHandler class. Lines from 14 to 17 defines OnReceiveData class. Line 19 creates variable "a" for recording the time before receiving a response. Lines from 20 to 32 are login codes. Lines from 34 to 40 are codes for the object t1102 of the code "005935".

Lines from 43 to 44 are codes for waiting messages. Line 45 opens the eBest.txt that records the response time. Line 46 creates the variable for current time. Line 47 creates the string for current time. Line 49 creates the variable for receiving the name. Line 50 creates the variable for receiving the current price. Line 51 prints the name and the current price. Line 52 create variable "b" for recording the time after receiving a response. Line 54 creates variable "c" for computing response time. Line 55 saves the code and the response time to file. Line 56 closes file.

Table 4 is the result of response time of eBest Investment & Securities Co. The mean is 1.95 seconds, the median is 1.82 seconds, and the standard deviation is 0.42 seconds. Since the difference between the mean and the median is not large and the standard deviation is small, we could see that the distribution of response time is dense with the center. The maximum value is 5.11 seconds, and the minimum value is 1.56 seconds. The range is 3.55 seconds, which is 1.9 times that of Daishin Securities Co. We can know the distribution of response time is less dense than that of Daishin Securities Co.

```
currentPrice.py - C:\Stock_eBest\currentPrice.py (3.7.0)                    —    □    ×
File  Edit  Format  Run  Options  Window  Help
import win32com.client
import pythoncom
import time
import datetime

class XASessionEventHandler:
    login_state = 0
    def OnLogin(self, code, msg):
        if code == "0000":
            print("Success")
            XASessionEventHandler.login_state = 1
        else:
            print("Failure")
class XAQueryEventHandlerT1102:
    query_state = 0
    def OnReceiveData(self, code):
        XAQueryEventHandlerT1102.query_state = 1

a= time.time()
# ----------------------------------------------------------------------
# login
# ----------------------------------------------------------------------
id = ""
passwd = ""
cert_passwd = ""

instXASession = win32com.client.DispatchWithEvents("XA_Session.XASession", XASessionEventHandler)
instXASession.ConnectServer("hts.ebestsec.co.kr", 20001)
instXASession.Login(id, passwd, cert_passwd, 0, 0)

while XASessionEventHandler.login_state == 0:
    pythoncom.PumpWaitingMessages()

# ----------------------------------------------------------------------
# t1102
# ----------------------------------------------------------------------
instXAQueryT1102 = win32com.client.DispatchWithEvents("XA_DataSet.XAQuery", XAQueryEventHandlerT1102)
instXAQueryT1102.ResFileName = "C:\\eBEST\\xingAPI\\Res\\t1102.res"
instXAQueryT1102.SetFieldData("t1102InBlock", "shcode", 0, "005930")
instXAQueryT1102.Request(0)

while XAQueryEventHandlerT1102.query_state == 0:
    pythoncom.PumpWaitingMessages()

f = open('C:\\Stock_eBest\\eBest.txt', 'a')
nowDate = datetime.datetime.now()
s = nowDate.strftime("%Y-%m-%d")

name = instXAQueryT1102.GetFieldData("t1102OutBlock", "shcode", 0)
price = instXAQueryT1102.GetFieldData("t1102OutBlock", "price", 0)
print(name, price)
b= time.time()

c= b-a
f.write('%s, %f \n' %(s, c))
f.close()
                                                                      Ln: 57  Col: 0
```

Figure 2. Python program of eBest Investment & Securities Co. for response time of the current price

Table 4. Result of Response Time of eBest Investment & Securities Co.

Number of experiments	Mean (second)	Standard Deviation (second)	Median (second)	Minimum (second)	Maximum (second)
420	1.95	0.42	1.82	1.56	5.11

2.3 Comparison of two response times

Daishin Securities Co. provides CYBOS plus for open API. eBest Investment & Securities Co. provides xingAPI for open API. There are two important differences between CYBOS plus and xingAPI. First, we must log in to CYBOS plus to access the server of Daishin Securities Co. And the python program does not require a logon. However, to receive data using xingAPI, users log on in an individual Python program. Second, CYBOS plus receives data in a Request/Reply method, and zingAPI receives data through events. It is thought that these points will lead to a difference in response time. Response time is important to users who use open APIs. Therefore, the study on the difference in response time would have an important meaning.

Since the response times of both APIs were alternately measured in the same environment, paired t-test is appropriate for the difference test as shown in Ryu [9]. We performed paired t-test by R. The results are shown in Figure 3. Null hypothesis is difference in means is equal to 0. Because p-value is 0.2961, we do not reject null hypothesis. Therefore, we can see that there is no significant difference between means. This result is unexpected considering the two differences of the two open APIs described above.
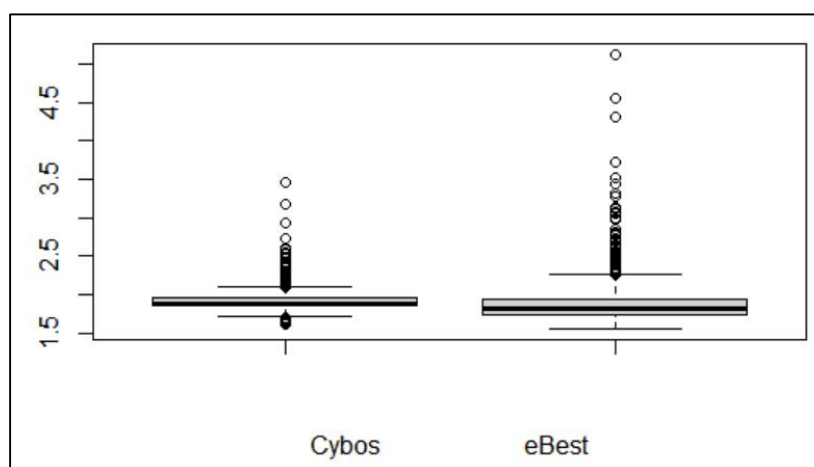
```
> t.test(response$Cybos, response$eBest, paired = TRUE)

        Paired t-test

data:  response$Cybos and response$eBest
t = -1.0461, df = 419, p-value = 0.2961
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.05291298  0.01615624
sample estimates:
mean of the differences
          -0.01837837
```

Figure 3. Result of Paired t-test

A boxplot is used to graphically see the difference between the two response distributions. Figure 4 is a boxplot comparing the two distributions. Looking at the boxplot, we can see that the distribution of the response time of eBest is more spread out than that of Cybos, and the position of the center is slightly lower. From the result of the hypothesis test, this difference is not significant.

Figure 4. Boxplot of two response times

In addition to the two major differences between the two APIs, CYBOS plus has no restrictions on Python programming, but xingAPI has some limits because it indirectly supports Python programming. For example, there is a limit to receiving more than one current price at once. This point is very important along with the response time.

3. Conclusion

Securities and investment services have and use large data. Investors started to invest through their own analysis methods. Therefore, securities and investment companies provide stock-related data to investors through open API. There are 22 major securities and investment companies in Korea. Only 6 companies such as Daishin Securities Co. and eBest Investment & Securities Co., Kiwoom Securities Co., Yuanta Securities Co., Hana Financial Investment Co., IBK Securities Co., support open API. The data received using the open API is in text format. Python is effective and convenient for requesting and receiving, analyzing text data.

Only Daishin Securities Co. supports Python officially, which name is CYBOS plus. eBest Investment & Securities Co. supports python unofficially, which name is xingAPI. There are two important differences between CYBOS plus and xingAPI. First, we must log in to CYBOS plus to access the server of Daishin Securities Co. And the python program does not require a logon. However, to receive data using xingAPI, users log on in an individual Python program. Second, CYBOS plus receives data in a Request/Reply method, and zingAPI receives data through events. It is thought that these points will lead to a difference in response time. Response time is important to investors who use open APIs.

Data were measured from August 5, 2021, to February 3, 2022. For each measurement, 15 repeated measurements were taken to obtain 420 measurements. To increase the accuracy of the study, both APIs were measured alternately under same conditions. A paired t-test was performed to test the hypothesis that the null hypothesis is there was no difference in means. Because p-value is 0.2961, we do not reject null hypothesis. Therefore, we can see that there is no significant difference between means. This result is unexpected considering the two differences of the two open APIs described above. A boxplot is used to graphically see the difference between the two response distributions. Looking at the boxplot, we can see that the distribution of the response time of eBest is more spread out than that of Cybos, and the position of the center is slightly lower. In addition to the two major differences between the two APIs, CYBOS plus has no restrictions on Python programming, but xingAPI has some limits because it indirectly supports Python programming. For example, there is a limit to receiving more than one current price at once. This point is

very important along with the response time.

Acknowledgement

References

[1] McKinsey Global Institute, *Big Data: The next frontier for innovation, competition, and productivity*, 2011.

[2] Chun, Se-Hak, "The Effect of Data Size on the k-NN Predictability: Application to Samsung Electronics Stock Market Prediction,*" JIntell Inform Syst(jiis)*, Vol. 25, No. 3, pp. 239~251, 2019.
DOI: http://dx.doi.org/10.13088/jiis.2019.25.3.239

[3] Jinho Byun, "Current Status and Perspectives of Fintech Innovation," *Journal of New Industry and Business*, Vol 51, No. 2, pp. 35-48, 2018
DOI: 10.30753/emr.2018.36.2.003

[4] Seonghoon Jang and Seung-Jung Shin, "Machine Learning-Based Programming Analysis Model Proposal: Based on User Behavioral Analysis," *International Journal of Advanced Smart Convergence(IJASC)*, Vol.9, No.4, pp. 179-183, 2020.
DOI: http://dx.doi.org/10.7236/IJASC.2020.9.4.179

[5] Soonduck Yoo, "Research on the Methodology for Policy Deriving to active Artificial Intelligence," *The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC)*, Vol. 20, No. 5, pp.187-193, 2020.
DOI: https://doi.org/10.7236/JIIBC.2020.20.5.187

[6] Gui Yeol Ryu, "A Study on Open API of Securities and Investment Companies for Activating Big Data," *International Journal of Advanced Smart Convergence(IJASC)*, Vol. 8, No. 2, pp. 102-108, 2019.
DOI: http://dx.doi.org/10.7236/IJASC.2019.8.2.102

[7] Gui Yeol Ryu, "A Study on Comparison of Open Application Programming Interface of Securities Companies Supporting Python," *International Journal of Advanced Smart Convergence(IJASC)*, Vol. 10, No. 1, pp. 97-104, 2021.
DOI: http://dx.doi.org/10.7236/IJASC.2021.10.1.97

[8] Dae Pyo Jo, *Algorithm Trading Learning in Python*, Wikibooks, 2019.

[9] Gui Yeol Ryu, *Introduction to Big Data Analysis R and Big Data Platform Hadoop*, 2019.