

Automatizirano trgovanje financijskim instrumentima temeljeno na Ichimoku Cloud metodi

Zavišić, Dario

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Department of Mathematics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:126:598102>

Rights / Prava: [In copyright / Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-04-27**



Repository / Repozitorij:

[Repository of School of Applied Mathematics and Computer Science](#)



Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike i računarstva

Dario Zavišić

**Automatizirano trgovanje financijskim
instrumentima temeljeno na Ichimoku Cloud
metodi**

Završni rad

Osijek, 2021.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike i računarstva

Dario Zavišić

**Automatizirano trgovanje financijskim
instrumentima temeljeno na Ichimoku Cloud
metodi**

Završni rad

Mentor: izv. prof. dr. sc. Nenad Šuvak

Osijek, 2021.

English paper title

Sažetak

U ovome radu ćemo se baviti metodom tehničke analize pod nazivom Ichimoku cloud. Rad je podijeljen na dva dijela. U prvom dijelu ovog rada nalazi se povijest metode te detaljna analiza tehničkih aspekata i matematičkih izračuna unutar metode. Drugi dio sastoji se od implementacije metode Ichimoku cloud u programskom jeziku Python te primjene iste na tržište s kriptovalutama.

Ključne riječi

Ichimoku cloud, algoritam, trgovanje, kriptovaluta, analiza.

Abstract

In this paper we will discuss technical analysis method named Ichimoku cloud. The paper is separated into two sections. In the first section of this paper, we will read about history of the method and a detailed analysis of technical aspects and mathematical computations inside of the method. The second section contains implementation of the Ichimoku cloud method in Python programming language and also its application to the cryptocurrency market.

Key words

Ichimoku cloud, algorithm, trading, cryptocurrency, analysis.

Sadržaj

Uvod	1
1 Ichimoku cloud	2
1.1 Povijest	2
1.2 Indikatori i njihov izračun	2
1.2.1 Tenkan-sen	2
1.2.2 Kijun-sen	3
1.2.3 Senkou Span A	5
1.2.4 Senkou Span B	5
1.2.5 Chikou Span	6
1.3 Analiza indikatora	7
1.3.1 Identifikacija trenda	7
1.3.2 Identifikacija signala za trgovanje	7
2 Implementacija u programskom jeziku Python	10
2.1 Dohvaćanje i priprema podataka	10
2.2 Obrada podataka i računanje	10
2.2.1 Tenkan-sen	10
2.2.2 Kijun-sen	10
2.2.3 Senkou Span A	11
2.2.4 Senkou Span B	11
2.2.5 Chikou Span	11
2.3 Računanje signala za trgovanje	11
2.3.1 Pomoćne funkcije	11
2.3.2 Funkcija ichimoku_cloud_buy_sell()	13
2.4 Grafički prikaz podataka	15
2.5 Testiranje metode Ichimoku Cloud	17
2.5.1 Web stranica	17
2.5.2 Testiranje	20
Literatura	24

Uvod

Metoda Ichimoku cloud je svestrani pokazatelj koji služi kako bi pokazao podršku i otpor, te kako bi odredio kretanje trenutnog trenda, njegov momentum i signale za kupnju, odnosno prodaju promatranog financijskog instrumenta. Način prikazivanja izračuna metode Ichimoku cloud je korištenjem grafičkog prikaz. Iako je naizgled grafički prikaz metode kompliciran, njezini pokazatelji su poprilično jasni te laki za razumjeti. Grafički prikaz ove metode nam omogućuje bolji uvid u trenutno stanje trenda i mogućih signala za ulazak u trgovanje.



Slika 1. Grafički prikaz Ichimoku cloud-a nad Bitcoin kriptovalutom

1 Ichimoku cloud

1.1 Povijest

Metoda Ichimoku cloud, nadalje u ovom radu samo metoda, je objavljena 1969. godine u knjizi njezinog stvaratelja, Japanskog novinara, Goichi Hosoda. Novinar Goichi Hosoda prijateljima poznatiji kao '*Ichimoku Sanjin*' što u prijevodu znači '*Što čovjek s planine vidi*' je iznimno inteligentan, a samozatajan čovjek. Svoje dane je provodio proučavajući tržište riže. Imao je odlično razumijevanje kako se cijene kreću te je shvaćao kako bi se one ponašale u određenim trenutcima na grafu. Stoga je odlučio napraviti svoju metodu koja bi objedinila različite pokazatelje u jedan grafički prikaz. S velikom grupom studenata proveo je 20 godina testirajući različite metode dok nije došao do svoje konačne Ichimoku cloud metode koja je vrlo brzo postala jedna od najkorištenijih metoda u Japanu. Na japanskom ova metoda nosi naziv '*Ichimoku Kinko Hyo*' što u prijevodu znači '*grafikon ravnoteže jednim pogledom*'.

1.2 Indikatori i njihov izračun

Metoda se sastoji od pet linija grafičkih podataka. Četiri od pet linija unutar metode su bazirane na prosjeku između najviše i najniže vrijednosti unutar odabranog vremenskog perioda.

1.2.1 Tenkan-sen

Tenkan-sen je linija koja je poznatija pod nazivom pretvorbena linija (*eng. conversion line*) skraćeno CL. Ona je sredina devetodnevnog perioda raspona od najviše do najniže vrijednosti financijskog instrumenta, što znači da se proteže u rasponu nešto manjem od dva tjedna. Izračun:

$$CL_i = \frac{9_PNV + 9_PNN}{2}, \quad (1)$$

$$9_PNV = \max\{v_9, v_8, \dots, v_1\} \quad (1.1)$$

$$9_PNN = \min\{n_9, n_8, \dots, n_1\} \quad (1.2)$$

gdje je:

CL_*i* - vrijednost Tekan-sen linije u *i*-tom danu

v_*j* - najviša postignuta vrijednost financijskog instrumenta *j*-tog dana

n_*j* - najniža postignuta vrijednost financijskog instrumenta *j*-tog dana

9_PNV - najviša postignuta vrijednost financijskog instrumenta unutar zadnjih 9 dana

9_PNN - najniža postignuta vrijednost financijskog instrumenta unutar zadnjih 9 dana

Date	Open	High	Low	Close	Adj Close	Volume	9_PNV	9_PNN	Tekan-sen
31.12.2019	7,2944E+14	7,3353E+13	7,1698E+14	7,1936E+14	7,1936E+14	2,117E+10			
1.1.2020	7,1949E+14	7,2543E+14	7,1749E+13	7,2002E+14	7,2002E+14	1,857E+10			
2.1.2020	7,2026E+14	7,2122E+13	6,9353E+14	6,9855E+14	6,9855E+14	2,08E+10			
3.1.2020	6,9844E+13	7,4137E+14	6,915E+11	7,3449E+14	7,3449E+14	2,811E+10			
4.1.2020	7,3454E+14	7,4274E+13	7,3095E+14	7,4107E+14	7,4107E+14	1,844E+10			
5.1.2020	7,4105E+14	7,5445E+13	7,4005E+14	7,4113E+13	7,4113E+13	1,973E+10			
6.1.2020	7,4105E+13	7,7819E+10	7,4093E+11	7,7692E+14	7,7692E+14	2,328E+10			
7.1.2020	7,7687E+14	8,1782E+13	7,7682E+13	8,1637E+13	8,1637E+13	2,877E+10			
8.1.2020	8,1619E+12	8,3967E+11	7,9568E+13	8,0799E+14	8,0799E+14	3,167E+10	7,414E+14	6,915E+11	3,71032E+14
9.1.2020	8,0823E+13	8,0823E+13	7,8424E+14	7,8791E+13	7,8791E+13	2,405E+10	7,414E+14	6,915E+11	3,71032E+14
10.1.2020	7,8783E+13	8,1666E+14	7,7268E+14	8,1666E+14	8,1666E+14	2,871E+10	8,167E+14	6,915E+11	4,08673E+14
11.1.2020	8,1622E+14	8218359375	8,0296E+14	8,0375E+14	8,0375E+14	2,552E+10	8,167E+14	6,915E+11	4,08673E+14
12.1.2020	8,0333E+11	8,2001E+13	8,0091E+14	8,1925E+12	8,1925E+12	2,29E+10	8,167E+14	7,409E+11	4,08698E+14
13.1.2020	8,1898E+14	8,1978E+13	8,0797E+14	8,1442E+13	8,1442E+13	2,248E+10	8,167E+14	7,409E+11	4,08698E+14
14.1.2020	8,1409E+14	8,8795E+11	8,1409E+14	8,8278E+13	8,8278E+13	4,484E+10	8,167E+14	7,409E+11	4,08698E+14
15.1.2020	882534375	8,8901E+10	86571875	8,807E+13	8,807E+13	4,01E+10	8,167E+14	86571875	4,08328E+14
16.1.2020	8,8125E+13	8,8465E+10	8,6121E+12	8,7238E+13	8,7238E+13	3,131E+10	8,167E+14	86571875	4,08328E+14
17.1.2020	8,7252E+13	8,9581E+13	8,6773E+11	8,929E+13	8,929E+13	3,637E+10	8,167E+14	86571875	4,08328E+14
18.1.2020	8,9272E+13	9,0122E+13	8,8273E+11	8,9428E+11	8,9428E+11	3,234E+10	8,167E+14	86571875	4,08328E+14
19.1.2020	8,9414E+10	9,1644E+13	8,6201E+12	8,7062E+13	8,7062E+13	3,422E+10	9,164E+13	86571875	4,58219E+13
20.1.2020	8,7046E+13	8,7456E+13	8,5605E+13	8,6576E+12	8,6576E+12	2,642E+10	9,164E+13	86571875	4,58219E+13
21.1.2020	8,659E+13	8,7557E+13	8,5445E+13	8,7459E+11	8,7459E+11	2,41E+10	9,164E+13	86571875	4,58219E+13

Slika 2. Prikaz izračunatih vrijednosti za pretvorbenu liniju u Excelu

1.2.2 Kijun-sen

Kijun-sen je linija koja je poznatija pod nazivom bazne linije (*eng. base line*) skraćeno BL. Ona je sredina perioda od 26 dana gdje gledamo raspon od najviše do najniže vrijednosti financijskog instrumenta. Izračun:

$$BL_i = \frac{26_PNV + 26_PNN}{2}, \quad (2)$$

$$26_PNV = \max\{v_26, v_25, \dots, v_1\} \quad (2.1)$$

$$26_PNN = \min\{n_26, n_25, \dots, n_1\} \quad (2.2)$$

gdje je:

BL_i - vrijednost Kijun-sen linije u i-tom danu

v_j - najviša postignuta vrijednost financijskog instrumenta j-tog dana

n_j - najniža postignuta vrijednost financijskog instrumenta j-tog dana

26_PNV - najviša postignuta vrijednost financijskog instrumenta unutar zadnjih 26 dana

26_PNN - najniža postignuta vrijednost financijskog instrumenta unutar zadnjih 26 dana

Date	Open	High	Low	Close	Adj Close	Volume	26_PNV	26_PNN	Kijun-sen
31.12.2019	7,2944E+14	7,3353E+13	7,1698E+14	7,1936E+14	7,1936E+14	2,117E+10			
1.1.2020	7,1949E+14	7,2543E+14	7,1749E+13	7,2002E+14	7,2002E+14	1,857E+10			
2.1.2020	7,2026E+14	7,2122E+13	6,9353E+14	6,9855E+14	6,9855E+14	2,08E+10			
3.1.2020	6,9844E+13	7,4137E+14	6,915E+11	7,3449E+14	7,3449E+14	2,811E+10			
4.1.2020	7,3454E+14	7,4274E+13	7,3095E+14	7,4107E+14	7,4107E+14	1,844E+10			
5.1.2020	7,4105E+14	7,5445E+13	7,4005E+14	7,4113E+13	7,4113E+13	1,973E+10			
6.1.2020	7,4105E+13	7,7819E+10	7,4093E+11	7,7692E+14	7,7692E+14	2,328E+10			
7.1.2020	7,7687E+14	8,1782E+13	7,7682E+13	8,1637E+13	8,1637E+13	2,877E+10			
8.1.2020	8,1619E+12	8,3967E+11	7,9568E+13	8,0799E+14	8,0799E+14	3,167E+10			
9.1.2020	8,0823E+13	8,0823E+13	7,8424E+14	7,8791E+13	7,8791E+13	2,405E+10			
10.1.2020	7,8783E+13	8,1666E+14	7,7268E+14	8,1666E+14	8,1666E+14	2,871E+10			
11.1.2020	8,1622E+14	8218359375	8,0296E+14	8,0375E+14	8,0375E+14	2,552E+10			
12.1.2020	8,0333E+11	8,2001E+13	8,0091E+14	8,1925E+12	8,1925E+12	2,29E+10			
13.1.2020	8,1898E+14	8,1978E+13	8,0797E+14	8,1442E+13	8,1442E+13	2,248E+10			
14.1.2020	8,1409E+14	8,8795E+11	8,1409E+14	8,8278E+13	8,8278E+13	4,484E+10			
15.1.2020	882534375	8,8901E+10	86571875	8,807E+13	8,807E+13	4,01E+10			
16.1.2020	8,8125E+13	8,8465E+10	8,6121E+12	8,7238E+13	8,7238E+13	3,131E+10			
17.1.2020	8,7252E+13	8,9581E+13	8,6773E+11	8,929E+13	8,929E+13	3,637E+10			
18.1.2020	8,9272E+13	9,0122E+13	8,8273E+11	8,9428E+11	8,9428E+11	3,234E+10			
19.1.2020	8,9414E+10	9,1644E+13	8,6201E+12	8,7062E+13	8,7062E+13	3,422E+10			
20.1.2020	8,7046E+13	8,7456E+13	8,5605E+13	8,6576E+12	8,6576E+12	2,642E+10			
21.1.2020	8,659E+13	8,7557E+13	8,5445E+13	8,7459E+11	8,7459E+11	2,41E+10			
22.1.2020	8,7442E+10	8,793E+12	8,6367E+13	8,6809E+13	8,6809E+13	2,26E+10			
23.1.2020	8,6807E+12	8,6877E+13	8,3336E+13	8406515625	8406515625	2,577E+10			
24.1.2020	8,4056E+13	8,5147E+13	8,2668E+13	8,4454E+13	8,4454E+13	2,44E+10			
25.1.2020	8,4401E+12	8458453125	829621875	8,3678E+11	8,3678E+11	1,965E+10	8,167E+14	86571875	4,0833E+14
26.1.2020	8,3644E+11	8,6024E+13	8,3255E+12	8,5968E+12	8,5968E+12	2,218E+10	8,167E+14	86571875	4,0833E+14
27.1.2020	8,5973E+11	8,9777E+10	8,5973E+11	8,9098E+13	8,9098E+13	2,865E+10	8,167E+14	86571875	4,0833E+14
28.1.2020	8,9125E+13	9,3586E+11	8,9084E+12	9,3586E+11	9,3586E+11	3,44E+10	8,167E+14	86571875	4,0833E+14
29.1.2020	9,3575E+12	9,4064E+12	9,2695E+13	9,3166E+13	9,3166E+13	3,068E+10	8,167E+14	86571875	4,0833E+14

Slika 3. Prikaz izračunatih vrijednosti za baznu liniju u Excelu

1.2.3 Senkou Span A

Senkou Span A je poznatija kao vodeći raspon A (*the leading span A*), koja se računa kao prosjek vrijednosti prikazanih pretvorbenom baznom linijom. Ona predstavlja sredinu između dvije linije, te formira gornju liniju oblaka koji je omeđen s Senkou Span A te Senkou Span B linijama. Izračun:

$$\text{Vodeći raspon A} = \frac{CL + BL}{2}, \quad (3)$$

gdje je:

CL - pretvorbena linija

BL - bazna linija.

	Date	Open	High	Low	Close	Adj Close	Volume	Tekan-sen	Kijun-sen	Senkou Span A
46	13.2.2020	1,0324E+11	1,0458E+13	1,0116E+14	1,0214E+14	1,0214E+14	4,936E+10	4,98677E+13	4,9384E+13	
47	14.2.2020	1,0212E+12	1,0322E+12	1,0126E+14	1,0312E+14	1,0312E+14	4,334E+10	4,98677E+13	4,9384E+13	
48	15.2.2020	1,0314E+14	1,0342E+14	9,8744E+12	9,8894E+13	9,8894E+13	4,387E+10	5,21917E+13	5,1708E+13	
49	16.2.2020	9,8892E+10	1005396875	9,7224E+11	9,9344E+11	9,9344E+11	4,337E+10	5,21917E+13	5,1708E+13	
50	17.2.2020	9,9366E+12	9,9388E+13	9,5076E+13	9,6901E+12	9,6901E+12	4,6E+10	5,21939E+13	5,1708E+13	
51	18.2.2020	9,6912E+11	1,0162E+13	9,6324E+10	1,0142E+12	1,0142E+12	4,727E+10	5,17559E+13	5,1708E+13	
52	19.2.2020	1,0144E+13	1,0192E+12	9,6112E+13	9,6334E+11	9,6334E+11	4,699E+10	5,17559E+13	5,1708E+13	
53	20.2.2020	9,6293E+13	9,6432E+12	9,5079E+12	9,6085E+13	9,6085E+13	4,493E+10	5,17559E+13	5,1756E+13	5,17559E+13
54	21.2.2020	9,6118E+13	9,723E+13	9,5897E+13	9,6864E+11	9,6864E+11	4,093E+10	5,17559E+13	5,1756E+13	5,17559E+13
55	22.2.2020	9,6877E+11	9,6982E+13	9,6007E+12	9,6632E+12	9,6632E+12	3,584E+10	5,17559E+13	5,1756E+13	5,17559E+13
56	23.2.2020	9,6633E+12	9,9374E+12	9,6578E+12	9924515625	9924515625	4,119E+10	5,17559E+13	5,1756E+13	5,17559E+13
57	24.2.2020	9,9216E+13	9,9517E+11	9,537E+11	9,6502E+13	9,6502E+13	4,508E+10	4,97422E+13	5,1756E+13	5,07491E+13
58	25.2.2020	96513125	9,6527E+13	9,305E+12	9,3417E+12	9,3417E+12	4,252E+10	4,97422E+13	5,1756E+13	5,07491E+13
59	26.2.2020	9,3383E+13	9,3548E+13	8,7044E+13	8,8205E+13	8,8205E+13	5,042E+10	4,86632E+13	5,1756E+13	5,02096E+13
60	27.2.2020	882509375	8,9329E+12	8,5772E+11	8,7845E+12	8,7845E+12	4,547E+10	4,90439E+13	5,1756E+13	5,03999E+13
61	28.2.2020	8,7887E+12	8,8905E+13	8,4929E+13	8,6725E+12	8,6725E+12	4,461E+10	4,90439E+13	5,1756E+13	5,03999E+13
62	29.2.2020	8,6712E+12	8,7756E+13	8,5995E+13	8,5995E+13	8,5995E+13	3,579E+10	4,90439E+13	5,1756E+13	5,03999E+13
63	1.3.2020	8,5998E+13	8726796875	8,4712E+12	8,5625E+13	8,5625E+13	3,535E+10	4,892E+13	5,1756E+13	5,0338E+13
64	2.3.2020	8,5633E+13	8,9213E+11	8,5326E+12	8,8697E+12	8,8697E+12	4,286E+10	4,86925E+13	5,1756E+13	5,02242E+13
65	3.3.2020	8,8654E+13	8,9016E+13	8,705E+12	8,7878E+13	8,7878E+13	4,239E+10	4,86925E+13	5,1756E+13	5,02242E+13
66	4.3.2020	8,7885E+13	8,8434E+13	8,7124E+12	8,7552E+11	8,7552E+11	3,475E+10	4,86925E+13	5,1756E+13	5,02242E+13
67	5.3.2020	8,7603E+11	9,1421E+10	8,7573E+11	9,0788E+13	9,0788E+13	3,97E+10	4,72028E+13	5,1756E+13	4,94793E+13

Slika 4. Prikaz izračunatih vrijednosti za vodeći raspon A u Excelu

1.2.4 Senkou Span B

Senkou Span B je poznatija kao vodeći raspon B (*the leading span B*), koja se računa kao prosjek najviše vrijednosti i najniže vrijednosti unutar zadnjih 52 dana. Ona čini donju granicu oblaka između Senkou Span A i Senkou Span B linija kao sredina perioda od 52 dana. Izračun:

$$(\text{Vodeći raspon B})_i = \frac{52_PNV + 52_PNN}{2}, \quad (4)$$

$$52_PNV = \max\{v_{52}, v_{51}, \dots, v_1\} \quad (4.1)$$

$$52_PNN = \min\{n_{52}, n_{51}, \dots, n_1\} \quad (4.2)$$

gdje je:

(Vodeći raspon B)_i - vrijednost Senkou Span B linije u i-tom danu

v_j - najviša postignuta vrijednost financijskog instrumenta j-tog dana

n_j - najniža postignuta vrijednost financijskog instrumenta j-tog dana

52_PNV - najviša postignuta vrijednost financijskog instrumenta unutar zadnjih 52 dana

52_PNN - najniža postignuta vrijednost financijskog instrumenta unutar zadnjih 52 dana

	Date	Open	High	Low	Close	Adj Close	Volume	S2_PNV	S2_PNN	Senkou Span B
75	13.3.2020	5,0178E+13	5,8381E+14	4,107E+14	5,5637E+11	5,5637E+11	7,416E+10	8,178E+14	76428125	
76	14.3.2020	5,5731E+14	5,6252E+10	5,1251E+13	5,2004E+13	5,2004E+13	3,615E+10	8,178E+14	76428125	
77	15.3.2020	5,2011E+14	5,8366E+14	5,1693E+12	5,3923E+14	5,3923E+14	3,4E+10	8,178E+14	76428125	
78	16.3.2020	5,3852E+13	5,3852E+13	4,5754E+14	5,0145E+14	5,0145E+14	4,537E+10	8,178E+14	76428125	
79	17.3.2020	5002578125	5,3713E+13	4,9819E+13	5,2256E+14	5,2256E+14	3,862E+10	8,178E+14	76428125	4,08328E+14
80	18.3.2020	5,2271E+14	5,3318E+12	5,0693E+10	5,2384E+13	5,2384E+13	3,788E+10	8,178E+14	76428125	4,08328E+14
81	19.3.2020	5,2454E+14	6,3297E+14	523696875	6,1912E+14	6,1912E+14	5,1E+10	8,178E+14	76428125	4,08328E+14
82	20.3.2020	6,1917E+14	6,8443E+11	5,8658E+14	6,1988E+13	6,1988E+13	5,444E+10	8,178E+14	76428125	4,08328E+14
83	21.3.2020	6,2065E+12	6,3781E+14	5,9328E+13	6,1851E+11	6,1851E+11	4,249E+10	8,178E+14	76428125	4,08328E+14
84	22.3.2020	6,1856E+14	6,3597E+12	5,8237E+13	5,8303E+13	5,8303E+13	4,01E+10	8,178E+14	76428125	4,08328E+14
85	23.3.2020	5,8314E+14	6,4439E+13	5,785E+14	6,4163E+14	6,4163E+14	4,649E+10	8,178E+14	76428125	4,08328E+14
86	24.3.2020	6,4366E+12	6,789E+14	6,4111E+11	6,7348E+13	6,7348E+13	4,822E+10	8,178E+14	76428125	4,08328E+14
87	25.3.2020	6,7387E+12	6,8925E+14	6,5369E+14	6,6811E+14	6,6811E+14	4,459E+10	8,178E+14	76428125	4,08328E+14
88	26.3.2020	6,6752E+13	6,7355E+13	6,591E+12	6,7164E+13	6,7164E+13	3,532E+10	8,178E+14	76428125	4,08328E+14
89	27.3.2020	6,7194E+14	6,7938E+14	6,4667E+14	6,4698E+14	6,4698E+14	3,459E+10	8,178E+14	76428125	4,08328E+14
90	28.3.2020	6,4673E+11	6,4675E+13	6,1178E+12	6,2422E+14	6,2422E+14	3,489E+10	8,178E+14	76428125	5,17078E+13
91	29.3.2020	6,2456E+14	6,2505E+14	5,9201E+10	5,922E+11	5,922E+11	2,837E+10	8,178E+14	76428125	5,17078E+13
92	30.3.2020	5,9255E+14	6,5172E+14	5903234375	6,4298E+12	6,4298E+12	3,71E+10	8,178E+14	76428125	5,17078E+13
93	31.3.2020	6,4306E+13	6,5045E+14	6,3742E+12	6,4386E+11	6,4386E+11	3,279E+10	8,178E+14	76428125	5,17078E+13
94	1.4.2020	6,4373E+13	6,6126E+14	6,2024E+14	6,6068E+13	6,6068E+13	4,035E+10	8,178E+14	76428125	5,17078E+13
95	2.4.2020	6,6068E+13	7,0882E+14	6,5959E+14	6,7936E+14	6,7936E+14	4,766E+10	8,178E+14	76428125	5,17082E+13
96	3.4.2020	6,7974E+12	7,0032E+12	6,6733E+10	6,7334E+14	6,7334E+14	3,898E+10	8,178E+14	76428125	5,17082E+13
97	4.4.2020	6,7384E+10	6,879E+14	6,6965E+14	6,8675E+11	6,8675E+11	3,319E+10	8,178E+14	76428125	4,0889E+14
98	5.4.2020	6,8625E+14	6,8834E+10	6,7159E+14	6,7911E+14	6,7911E+14	2,951E+10	8,178E+14	76428125	4,0889E+14
99	6.4.2020	6,788E+13	727178125	6,7829E+13	727178125	727178125	4,69E+10	8,178E+14	76428125	4,0889E+14
100	7.4.2020	7,2736E+14	7,4279E+12	7,1367E+14	7,1764E+14	7,1764E+14	4,424E+10	8,178E+14	76428125	4,0889E+14

Slika 4. Prikaz izračunatih vrijednosti za vodeći raspon B u Excelu

1.2.5 Chikou Span

Chikou Span je poznatija kao raspon zaostajanja (*the lagging span*). Ona prikazuje zadnje vrijednosti (*eng. closing price*) tijekom dnevne faze trgovanja unutar perioda od 26 dana.

	Date	Open	High	Low	Close	Adj Close	Volume	Chikou Span
	31.12.2019	7,2944E+14	7,3353E+13	7,1698E+14	7,1936E+14	7,1936E+14	2,117E+10	8,36785E+11
	1.1.2020	7,1949E+14	7,2543E+14	7,1749E+13	7,2002E+14	7,2002E+14	1,857E+10	8,59683E+12
	2.1.2020	7,2026E+14	7,2122E+13	6,9353E+14	6,9855E+14	6,9855E+14	2,08E+10	8,90982E+13
	3.1.2020	6,9844E+13	7,4137E+14	6,915E+11	7,3449E+14	7,3449E+14	2,811E+10	9,35859E+11
	4.1.2020	7,3454E+14	7,4274E+13	7,3095E+14	7,4107E+14	7,4107E+14	1,844E+10	9,31663E+13
	5.1.2020	7,4105E+14	7,5445E+13	7,4005E+14	7,4113E+13	7,4113E+13	1,973E+10	9,50899E+13
	6.1.2020	7,4105E+13	7,7819E+10	7,4093E+11	7,7692E+14	7,7692E+14	2,328E+10	9,35053E+12
	7.1.2020	7,7687E+14	8,1782E+13	7,7682E+13	8,1637E+13	8,1637E+13	2,877E+10	9392875
	8.1.2020	8,1619E+12	8,3967E+11	7,9568E+13	8,0799E+14	8,0799E+14	3,167E+10	9,34437E+12
	9.1.2020	8,0823E+13	8,0823E+13	7,8424E+14	7,8791E+13	7,8791E+13	2,405E+10	9,29352E+12
	10.1.2020	7,8783E+13	8,1666E+14	7,7268E+14	8,1666E+14	8,1666E+14	2,871E+10	9,18096E+12
	11.1.2020	8,1622E+14	8218359375	8,0296E+14	8,0375E+14	8,0375E+14	2,552E+10	9,61342E+12
	12.1.2020	8,0333E+11	8,2001E+13	8,0091E+14	8,1925E+12	8,1925E+12	2,29E+10	9,7298E+13
	13.1.2020	8,1898E+14	8,1978E+13	8,0797E+14	8,1442E+13	8,1442E+13	2,248E+10	9,79594E+12
	14.1.2020	8,1409E+14	8,8795E+11	8,1409E+14	8,8278E+13	8,8278E+13	4,484E+10	9,86512E+12
	15.1.2020	882534375	8,8901E+10	86571875	8,807E+13	8,807E+13	4,01E+10	1,01167E+13
	16.1.2020	8,8125E+13	8,8465E+10	8,6121E+12	8,7238E+13	8,7238E+13	3,131E+10	9,85661E+12
	17.1.2020	8,7252E+13	8,9581E+13	8,6773E+11	8,929E+13	8,929E+13	3,637E+10	1,02082E+13
	18.1.2020	8,9272E+13	9,0122E+13	8,8273E+11	8,9428E+11	8,9428E+11	3,234E+10	1,03261E+11
	19.1.2020	8,9414E+10	9,1644E+13	8,6201E+12	8,7062E+13	8,7062E+13	3,422E+10	1,02144E+14
	20.1.2020	8,7046E+13	8,7456E+13	8,5605E+13	8,6576E+12	8,6576E+12	2,642E+10	1,03121E+14
	21.1.2020	8,659E+13	8,7557E+13	8,5445E+13	8,7459E+11	8,7459E+11	2,41E+10	9,88942E+13
	22.1.2020	8,7442E+10	8,793E+12	8,6367E+13	8,6809E+13	8,6809E+13	2,26E+10	9,93443E+11
	23.1.2020	8,6807E+12	8,6877E+13	8,3336E+13	8406515625	8406515625	2,577E+10	9,69014E+12
	24.1.2020	8,4056E+13	8,5147E+13	8,2668E+13	8,4454E+13	8,4454E+13	2,44E+10	1,0142E+12
	25.1.2020	8,4401E+12	8458453125	829621875	8,3678E+11	8,3678E+11	1,965E+10	9,63339E+11

Slika 6. Prikaz izračunatih vrijednosti za raspon zaostajanja u Excelu

1.3 Analiza indikatora

Ova metoda se često koristi za identifikaciju područja podrške i otpora ovisno o relativnoj poziciji vrijednosti. S obzirom da oblak (*eng. cloud*) pruža razine podrške i otpora koji se mogu prikazati za budućnost, ova metoda se time razlikuje od mnogih drugih metoda s tehničkim indikatorima koji pružaju razine podrške i otpora samo za trenutačno vrijeme i stanje.

1.3.1 Identifikacija trenda

Za identifikaciju trenda se primarno koristi oblak koji se dobije između linije vodećeg raspona A i linije vodećeg raspona B. Kako se vodeći raspon A (zeleni linija na slici 7.) računa kao prosjek vrijednosti prikazanih pretvorbenom i baznom linijom, a koje se računaju u periodima od 9 i 26 dana, češće mijenja svoje vrijednosti od vodećeg raspona B (crvene granice oblaka na slici 7.), koji se računa kao prosjek najviših i najnižih vrijednosti u prethodna 52 dana.

Postoje dva načina kako možemo odrediti sveukupni trend. Prvi način nam kaže da je trend rastući ako se vrijednosti nalaze iznad oblaka, padajući ako se nalaze ispod oblaka i stabilan ako se vrijednosti financijskog instrumenta nalaze unutar samog oblaka. Drugi način kaže da uzlazni trend jača kada vodeći raspon A raste i kada je iznad vodećeg raspona B, što nam stvara zeleni oblak na slici 7. Dok kada vodeći raspon A pada ispod vodećeg raspona B, nastaje crveni oblak na slici 7. koji nam ukazuje na padajući trend. S obzirom da je naš oblak napravljen tako da je pomjeren 26 dana unaprijed, njime dobijemo i mali pogled u budućnost podrške i otpora.



Slika 7. Identifikacija trenda pomoću oblaka nad Cardano kriptovalutom

1.3.2 Identifikacija signala za trgovanje

Cijena, pretvorbeni linija i bazna linija koriste se kako bi brže i češće identificirali signale za trgovanje. Treba uzeti u obzir da su bullish signali pojačani kada je cijena iznad oblaka

i oblak je zelen. Bearish signali su pojačani kada je cijena ispod oblaka i oblak je crven. Odnosno, bullish signali su poželjni kada je veći trend gore (cijene su iznad zelenog oblaka), dok su bearish signali poželjni kada je veći trend dolje (cijene su ispod crvenog oblaka). Signali koji su u suprotnosti s postojećim trendom smatraju se slabima, poput kratkoročnih bullish signala unutar dugoročno silaznog trenda ili kratkoročnih bearish signala unutar dugoročno uzlaznog trenda.

Signali pretvorbene-bazne linije Tijekom rastućeg trenda, bullish signal je okinut kada pretvorbena linija postigne vrijednosti veće od vrijednosti bazne linije. Slično, kada pretvorbena linija postigne vrijednosti manje od vrijednosti bazne linije tijekom padajućeg trenda, onda je to bearish signal.



Slika 8. Primjer signala za kupnju pomoću pretvorbene i bazne linije nad Cardano kriptovalutom



Slika 9. Primjer signala za prodaju pomoću pretvorbene i bazne linije nad Cardano kriptovalutom

Signali cijene-bazne linije Tijekom rastućeg trenda, bullish signal je okinut kada cijena financijskog instrumenta bude veća od vrijednosti bazne linije. Slično, kada cijena financijskog instrumenta bude manja od vrijednosti bazne linije tijekom padajućeg trenda dobivamo bearish signal.



Slika 10. Primjer signala za kupnju pomoću cijene i bazne linije nad Cardano kriptovalutom



Slika 11. Primjer signala za prodaju pomoću cijene i bazne linije nad Cardano kriptovalutom

2 Implementacija u programskom jeziku Python

2.1 Dohvaćanje i priprema podataka

Potrebne biblioteke:

```
1 import numpy as np
2 import pandas as pd
3 import yfinance as yf
4 import requests
5 from datetime import datetime
6 import sys
```

Za dohvaćanje podataka potrebno je definirati sljedeće varijable i definirati funkciju `get_data(name_, start_, end_)`.

```
1 time = datetime.now()
2 dt_string = time.strftime("%Y-%m-%d")
3 id_ = sys.argv[1] #wallet_id
4 name = sys.argv[2] #ime kriptovalute/dionice
5 start = sys.argv[3] #datum pocetka povijesnih podataka
6 stop_loss = sys.argv[4] #dopusten gubitak
7 take_profit = sys.argv[5] #dovoljan profit
8 end = str(dt_string) #trenutni datum i vrijeme
9 last_date_with_value = get_last_date(end)
10
11 def get_data(name_, start_, end_):
12     data = yf.download(name_, start=start_, end=end_)
13     return data
14
15 #Dodatna priprema podataka koristeći biblioteku Pandas
16 index = pd.date_range(end, periods=25, freq='D')
17 columns = data.columns
18 data_pred = pd.DataFrame(index=index, columns=columns)
19 data = pd.concat([data, data_pred])
20
21 data = get_data(name, start, end) #preuzima podatke s yahoo finance api-a
```

2.2 Obrada podataka i računanje

2.2.1 Tenkan-sen

Stupac pod nazivom "tenkan_sen" popunjavamo po formuli (1).

```
1 data['tenkan_sen'] = (data['High'].rolling(window = 9).max() + data['Low']
    ].rolling(window = 9).min()) / 2
```

2.2.2 Kijun-sen

Stupac pod nazivom "kijun_sen" popunjavamo po formuli (2).

```
1 data['kijun_sen'] = (data['High'].rolling(window = 26).max() + data['Low']
    ].rolling(window = 26).min()) / 2
```


2.2.3 Senkou Span A

Stupac pod nazivom "senkou_span_A" popunjavamo po formuli (3).

```
1 data['senkou_span_A'] = ((data['tenkan_sen'] + data['kijun_sen']) / 2).shift(26)
```

2.2.4 Senkou Span B

Stupac pod nazivom "senkou_span_B" popunjavamo po formuli (4).

```
1 data['senkou_span_B'] = ((data['High'].rolling(window=52).max() + data['Low'].rolling(window=52).min()) / 2).shift(26)
```

2.2.5 Chikou Span

Stupac pod nazivom "chikou_span" popunjavamo ubacivanjem vrijednosti prije 26 dana.

```
1 data['chikou_span'] = data['Close'].shift(-26)
```

2.3 Računanje signala za trgovanje

Za računanje signala za kupnju i prodaju definiramo pomoćne funkcije te glavnu funkciju za određivanje signala *ichimoku_cloud_buy_sell()*.

2.3.1 Pomoćne funkcije

```
1 def get_last_date(end):
2     last_date = end
3     day = str(last_date[-2] + last_date[-1])
4     month = str(last_date[-5] + last_date[-4])
5     year = str(last_date[2] + last_date[3])
6     if day=='01' and month=='01':
7         year=str(int(year)-1)
8         day='31'
9         month='12'
10    else:
11        if day=='01':
12            if month=='03':
13                if int(year)%4==0:
14                    month='02'
15                    day='29'
16                else:
17                    month='02'
18                    day='28'
19            elif month=='02' or month=='04' or month=='06' or month=='08'
or month=='09' or month=='11':
20                month=str(int(month)-1)
21                day='31'
22            else:
23                month=str(int(month)-1)
24                day='30'
25        else:
26            if int(day)<=10:
27                day = '0' + str(int(day)-1)
28            else:
```

```

29         day = str(int(day)-1)
30
31
32     last_date_with_value = str('20' + year + '-' + month + '-' + day)
33     return last_date_with_value
34
35 def stoppers_check(id_, stop_loss, take_profit, curr_close_price):
36
37     curr_close = curr_close_price
38     balance = get_wallet_balance(id_, "ICHIMOKU CLOUD")
39     if float(balance) == 0:
40         quantity = get_wallet_quantity(id_, "ICHIMOKU CLOUD")
41         if (float(quantity) * float(curr_close)) >= float(take_profit) or
42         (float(quantity) * float(curr_close)) <= float(stop_loss):
43             print("TERMINATE TRADING BOT")
44         else:
45             print("RUNNING 1")
46     else:
47         if float(balance) >= float(take_profit) or float(balance) <= float(
48         stop_loss):
49             print("TERMINATE TRADING BOT")
50         else:
51             print("RUNNING 2")
52
53 def get_latest_data(data, last_date_with_value):
54     curr_close_price = data.loc[last_date_with_value]['Close']
55     curr_span_a = data.loc[last_date_with_value]['senkou_span_A']
56     curr_span_b = data.loc[last_date_with_value]['senkou_span_B']
57     curr_kijun = data.loc[last_date_with_value]['kijun_sen']
58     curr_tenkan = data.loc[last_date_with_value]['tenkan_sen']
59     return curr_close_price, curr_span_a, curr_span_b, curr_kijun,
60     curr_tenkan
61
62 def create_wallet(id_, name, balance, method):
63     requests.post('http://localhost:3014/api/createWallet', json={"id": id_,
64     "name": name, "balance": balance, "method": method})
65
66 def get_wallet_balance(id_, method):
67     r = requests.get('http://localhost:3014/api/getWallet', json={"id": id_,
68     "method": method})
69     return r.json()['data'][0]['balance']
70
71 def buying_rebalance(id_, buying_price, quantity, method):
72     balance = float(get_wallet_balance(id_, method))
73     rebalance1 = balance/buying_price
74     rebalance = float(balance - (rebalance1*float(buying_price)))
75     if rebalance < 1:
76         rebalance = "0"
77     requests.post('http://localhost:3014/api/rebalance', json={"id": id_,
78     "rebalance": rebalance, "quantity": quantity, "method": method})
79
80 def selling_rebalance(id_, revenue, method):
81     balance = get_wallet_balance(id_, method)
82     rebalance = float(balance) + float(revenue)
83     requests.post('http://localhost:3014/api/rebalance', json={"id": id_,
84     "rebalance": rebalance, "quantity": "0", "method": method})

```

```

80 def get_wallet_quantity(id_,method):
81     r = requests.get('http://localhost:3014/api/getWallet', json={"id":id_
82     , "method":method})
83     return r.json()['data'][0]['quantity']
84
85 def hodling_check(id_,method):
86     r = requests.get('http://localhost:3014/api/getWallet', json={"id":id_
87     , "method":method})
88     if int(r.json()['data'][0]['quantity']) == 0 and int(r.json()['data'
89     ][0]['balance'])!=0:
90         return False
91     else:
92         return True

```

2.3.2 Funkcija `ichimoku_cloud_buy_sell()`

Signal za kupnju je implementiran na način da se kupnja obavi, u slučaju ako nismo već kupili, kada closing cijena bude iznad oblaka (vodećeg raspona A i vodećeg raspona B) te kada se bazna linija nalazi ispod pretvorbene linije, pri čemu njihova razlika nije veća od 2.5.

```

1 if curr_close_price > curr_span_a and curr_close_price > curr_span_b:
2     if curr_tenkan-curr_kijun >= 0 and curr_tenkan-curr_kijun <= 2.5:
3         if hodling==False:

```

Signal za prodaju je implementiran na način da se prodaja obavi, u slučaju ako smo pret-hodno obavili kupnju, kada closing cijena bude ispod oblaka te kada se bazna linija nalazi iznad pretvorbene linije, pri čemu njihova razlika nije veća od 2.5.

```

1 if hodling==True:
2     if curr_close_price < curr_span_a and curr_close_price < curr_span_b:
3         if curr_kijun-curr_tenkan >= 0 and curr_kijun-curr_tenkan <= 2.5:

```

Implementacija cijele funkcije `ichimoku_cloud_buy_sell()`:

```

1 def ichimoku_cloud_buy_sell(id_,stop_loss, take_profit, name,
2     curr_close_price, curr_span_a, curr_span_b, curr_kijun, curr_tenkan):
3     hodling = hodling_check(id_,"ICHIMOKU CLOUD")
4     stoppers_check(id_,stop_loss, take_profit, curr_close_price)
5     if curr_close_price > curr_span_a and curr_close_price > curr_span_b:
6         if curr_tenkan-curr_kijun >= 0 and curr_tenkan-curr_kijun <= 2.5:
7             if hodling==False:
8                 print("BUY SIGNAL TRIGGERED!")
9                 print('Bought at the price of:',curr_close_price,'$')
10                time = datetime.now()
11                r = requests.post('http://localhost:3014/api/order', json
12                ={
13                    "wallet_id": id_,
14                    "timestamp": str(time),
15                    "type":"BUY",
16                    "name": name,
17                    "quantity": str(float(get_wallet_balance(id_,"
18                    ICHIMOKU CLOUD"))/curr_close_price),
19                    "price":float(curr_close_price),
20                    "method": "ICHIMOKU CLOUD"
21                })
22                buying_rebalance(id_,curr_close_price, str(float(
23                get_wallet_balance(id_,"ICHIMOKU CLOUD"))/curr_close_price), "ICHIMOKU
24                CLOUD")

```



```

20         print(r.json())
21         hodling = True
22     else:
23         print("ALREADY HODLING!")
24     else:
25         print("Watch for conversion and base line! Might buy soon!")
26     else:
27         if hodling==True:
28             if curr_close_price < curr_span_a and curr_close_price <
curr_span_b:
29                 if curr_kijun-curr_tenkan >= 0 and curr_kijun-curr_tenkan
<= 2.5:
30                     print("SELL SIGNAL TRIGGERED!")
31                     print('Sold at the price of:',curr_close_price,'$')
32                     time = datetime.now()
33                     quantity = get_wallet_quantity(id_,"ICHIMOKU CLOUD")
34                     r = requests.post('http://localhost:3014/api/order',
json={
35                         "wallet_id": id_,
36                         "timestamp": str(time),
37                         "type": "SELL",
38                         "name": name,
39                         "quantity": str(quantity),
40                         "price": float(curr_close_price),
41                         "method": "ICHIMOKU CLOUD"
42                     })
43                     selling_rebalance(id_,str(float(quantity)*float(
curr_close_price)), "ICHIMOKU CLOUD")
44                     hodling=False
45                 else:
46                     print("Watch for conversion and base line! Might sell
soon!")
47             else:
48                 print("Doing nothing, still above cloud!")
49         else:
50             print("Still waiting for a good opportunity!")

```

2.4 Grafički prikaz podataka

Implementacija grafičkog prikaza metode i podataka je dana na sljedeći način:

```

1 def plot_data(self, data):
2     fig = go.Figure()
3     fig.add_trace(go.Candlestick(x=data.index,
4                                 open=data['Open'],
5                                 high=data['High'],
6                                 low=data['Low'],
7                                 close=data['Close'], name = 'market data'))
8
9     fig.update_layout(
10         title='Live share price evolution',
11         yaxis_title='Coin/Stock Price (US Dollars)'
12     )
13     fig.update_xaxes(
14         rangeslider_visible=True,
15         rangeselector=dict(
16             buttons=list([
17                 dict(step="all")
18             ])
19         )
20     )
21     fig.show()
22 def plot_ichimoku(self, data):
23     fig = go.Figure()
24
25     fig.add_trace(go.Candlestick(
26         x=data.index,
27         open=data['Open'],
28         high=data['High'],
29         low=data['Low'],
30         close=data['Close'],
31         name = 'Market Data'
32     ))
33
34     fig.add_trace(go.Scatter(
35         x=data.index,
36         y=data['tenkan_sen'],
37         line=dict(
38             color='royalblue',
39             width=.8
40         ),
41         name = 'Tenkan Sen'
42     ))
43
44     fig.add_trace(go.Scatter(
45         x=data.index,
46         y=data['kijun_sen'],
47         line=dict(
48             color='red',
49             width=.8
50         ),
51         name = 'Kijun Sen'
52     ))
53
54     fig.add_trace(go.Scatter(
55         x=data.index,

```

```

56     y=data['senkou_span_A'],
57     line=dict(
58         color='black',
59         width=.8
60     ),
61     name = 'Senkou Span A'
62 )
63 )
64 fig.add_trace(go.Scatter(
65     x=data.index,
66     y=data['senkou_span_B'],
67     line=dict(
68         color='purple',
69         width=.8
70     ),
71     name = 'Senkou Span B',
72     fill = 'tonexty',
73     fillcolor= 'lightgreen',
74     opacity=0.01
75 )
76 )
77 fig.add_trace(go.Scatter(
78     x=data.index,
79     y=data['chikou_span'],
80     line=dict(
81         color='orange',
82         width=.8
83     ),
84     name = 'Chikou Span'
85 )
86 )
87 fig.show()
88 fig.write_html("./ichimoku.html")
89 ichimoku.plot_ichimoku(data)

```

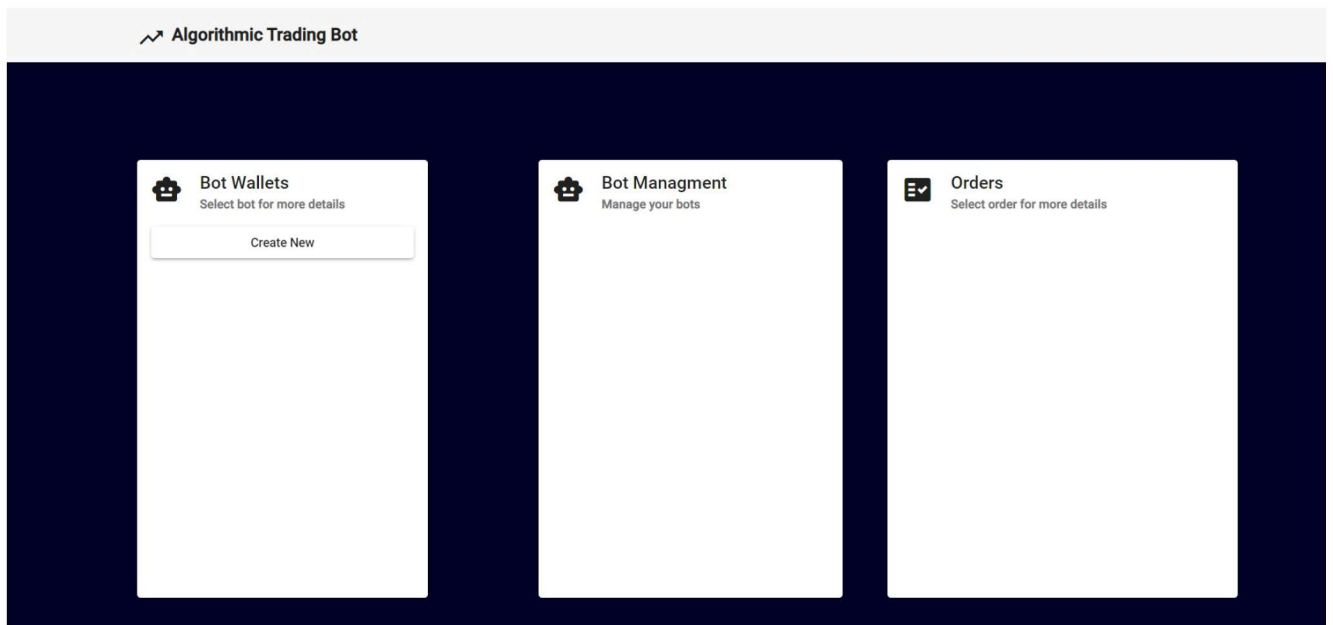


Prikaz izlazne datoteke s grafičkim prikazom, *ichimoku.html*

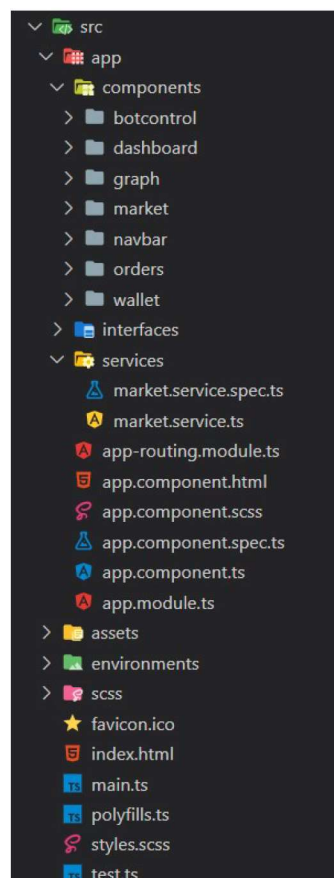
2.5 Testiranje metode Ichimoku Cloud

2.5.1 Web stranica

Za kreiranje simulacije crypto exchange-a koji ne koristi pravi novac, korišten je Angular. Glavni dio Angular projekt je market service koji služi za logiku povezivanja interakcije korisnika sa serverom.



Izgled web stranice



Struktura Angular projekta

```

1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4 import { tap } from 'rxjs/operators';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class MarketService {
10
11   marketData$: Observable<any>;
12   orderData$: Observable<any>;
13   wallet$: Observable<any>;
14   bot$: Observable<any>;
15   constructor(private http: HttpClient) {
16     this.marketData$ = this.http.get('http://localhost:3014/');
17     this.orderData$ = this.http.get('http://localhost:3014/api/getAllOrder
18 /');
19     this.wallet$ = this.http.get('http://localhost:3014/api/getAllWallet');
20     this.bot$ = this.http.get('http://localhost:3014/api/bot/getAll');
21   }
22
23   initializeBot(crypto: string, method: string, balance: number):
24     Observable<any> {
25     // const date = new Date().toISOString().slice(0, 10);
26     return this.http.post('http://localhost:3014/api/${method}/start', {
27       name: crypto,
28       start_date: '2020-01-01'
29     });
30   }
31
32   // tslint:disable-next-line:typedef
33   createWallet(wallet_id:string,name: string, method: string, amount:
34     string){
35     return this.http.post<any>('http://localhost:3014/api/createWallet', {
36       wallet_id:wallet_id,
37       name: name,
38       balance: amount,
39       method:method
40     }).pipe(
41       tap(data => console.log('creating new wallet', data))
42     );
43   }
44
45   addBot(wallet_id:string,name:string,method:string, status:string){
46     return this.http.post<any>('http://localhost:3014/api/bot/add', {
47       wallet_id:wallet_id,
48       name:name,
49       method:method,
50       status:status
51     }).pipe(
52       tap(data => console.log('adding new bot', data))
53     );
54   }

```

```

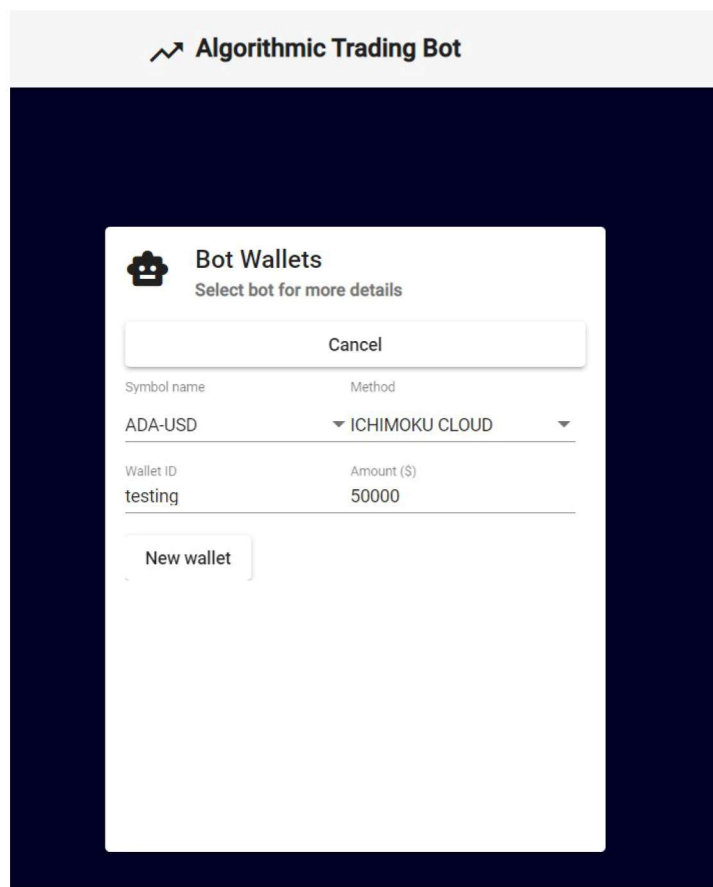
53
54 startBot(wallet_id:string, name:string, method: string, take_profit:
    string, stop_loss:string){
55     if(method==="ICHIMOKU CLOUD"){
56         method = "ichimoku";
57     }else if(method==="RSI"){
58         method="rsi";
59     }
60     else{
61         method="macd"
62     }
63     return this.http.post<any>('http://localhost:3014/api/${method}/start
        ', {
64         wallet_id:wallet_id,
65         name:name,
66         start_date:"2020-01-05", //date when the first data is fetched
67         take_profit:take_profit,
68         stop_loss:stop_loss
69     }).pipe(
70         tap(data => console.log('adding new bot', data))
71     );
72 }
73
74 stopBot(wallet_id:string, method:string){
75     if(method==="ICHIMOKU CLOUD"){
76         method = "ichimoku";
77     }else if(method==="RSI"){
78         method="rsi";
79     }
80     else{
81         method="macd"
82     }
83     return this.http.post<any>('http://localhost:3014/api/${method}/stop',
        {
84         wallet_id:wallet_id
85     }).pipe(
86         tap(data => console.log('stopping bot', data))
87     );
88 }
89 }

```

market.service.ts

2.5.2 Testiranje

Test 1 Pri kreiranju novčanika izabrana je kriptovaluta Cardano (ADA-USD), metoda Ichimoku Cloud, identifikacijska oznaka novčanika 'testing' te količina novaca 50000\$.



Symbol name	Method
ADA-USD	ICHIMOKU CLOUD

Wallet ID	Amount (\$)
testing	50000

Kreiranje novčanika

Nakon kreiranog novčanika unosimo vrijednosti zadovoljavajućeg iznosa zarade (*eng. take profit*) i prihvaćenog iznosa gubitka (*eng. stop loss*).

Algorithmic Trading Bot

Bot Walleets
Select bot for more details

Create New

ID: testing	ADA-USD	ICHIMOKU CLOUD	\$50000
-------------	---------	----------------	---------

Bot Managment
Manage your bots

ID: testing

Take profit (\$)
70000

Stop loss (\$)
30000

Start

Unos iznosa za take profit i stop loss u dolarima

Nakon pokretanja bota za trgovanje 31.5.2020., bot nam šalje obavijest da je prvu kupnju obavio dana 1.6.2020.

Algorithmic Trading Bot

Bot Walleets
Select bot for more details

Create New

ID: testing	ADA-USD	ICHIMOKU CLOUD	\$0
-------------	---------	----------------	-----

Bot Managment
Manage your bots

ID: testing

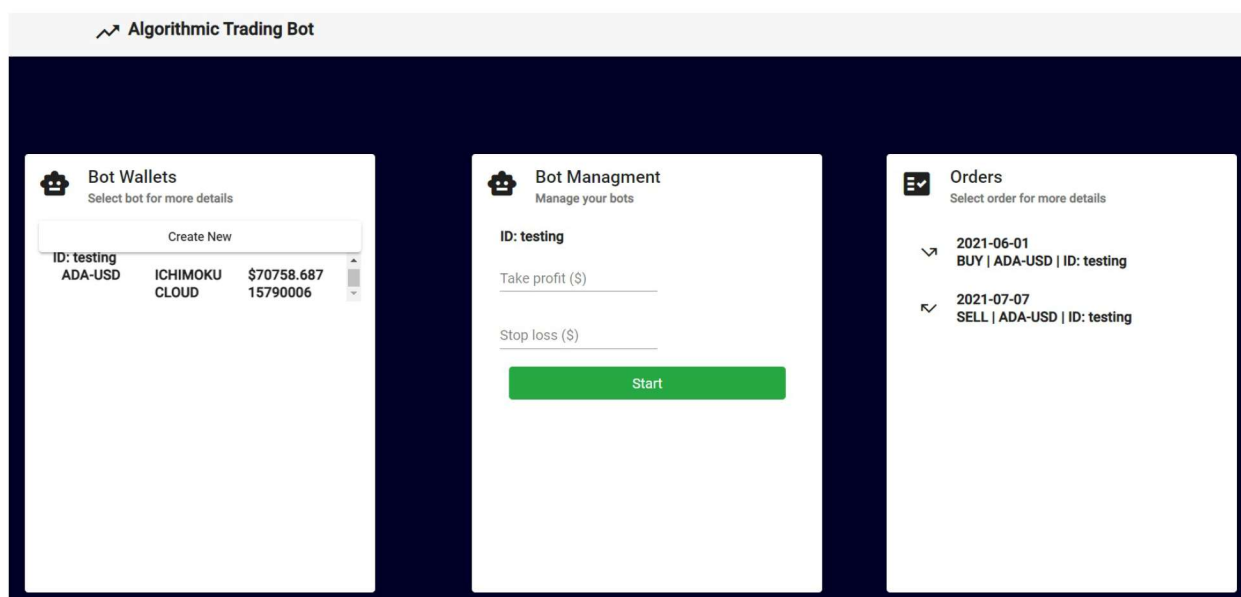
Stop

Orders
Select order for more details

2020-06-01
BUY | ADA-USD | ID: testing

Prikaz prve kupnje

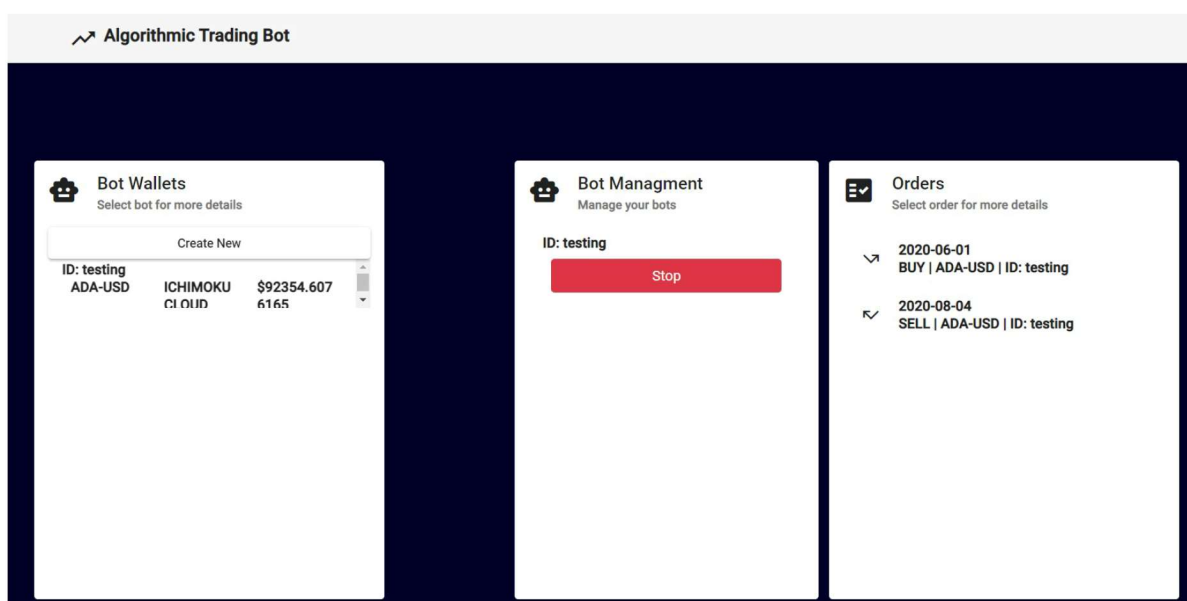
Zatim vidimo da je dana 7.7.2021. napravio prodaju kada je zadovoljio iznos zarade te time se zaustavio.



Prikaz prve prodaje

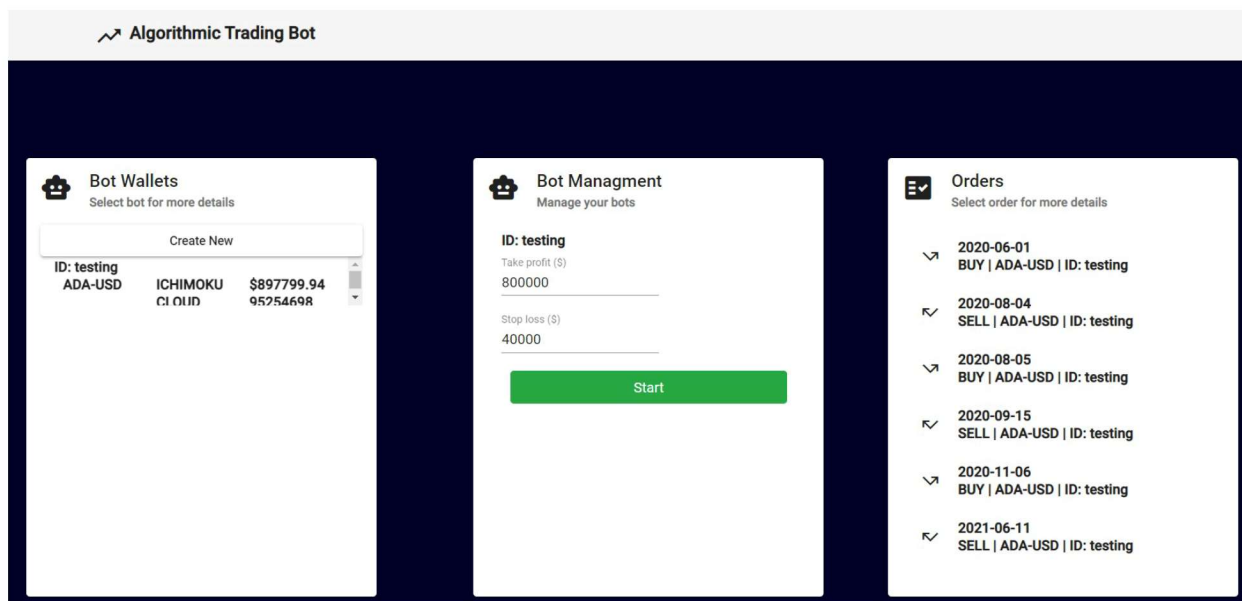
Test 2 Kreirani novčanik je jednak novčaniku iz Testa 1. Ali ovaj put je pokrenuti bot postavljen da uzme profit na 800000\$ i da prihvati gubitak na 40000\$.

Bot je obavio prvu kupnju 1.6.2020. pri čemu je kupio 672476.8339282782 ADA po cijeni od 0.07435200363397598\$, te je napravio prodaju po cijeni od 0.13733500242233276\$ čime je ostvario profit od 42354,6\$.



Prikaz trenutnog stanja u novčaniku nakon obavljene kupnje i prodaje kriptovalute Cardano

Zatim je bot obavio još 2 puta kupnje i prodaje, dok nije na zadnjoj prodaji ostvario profit veći od 800000\$.



Prikaz trenutnog stanja u novčaniku nakon dužeg perioda trgovanja bota nad kriptovalutom Cardano

Ako usporedimo rezultate Ichimoku cloud bota i mogućnost da nismo koristili bota, tj. da smo 1.6.2020. kupili s 500000\$ 672476.8339282782 ADA po cijeni od 0.07435200363397598\$ te da smo ih držali do 11.6.2021. ostvarili bismo 1030138.35461\$ što je veći profit nego što ga je bot ostvario.

Literatura

- [1] Cory Mitchel, *Ichimoku Cloud*
<https://www.investopedia.com/terms/i/ichimoku-cloud.asp>
- [2] Chris Nice, *History of Ichimoku*
<https://www.slideshare.net/ichi-mo-ku/history-of-ichimoku>
- [3] Web stranica IG: *Ichimoku Cloud definition*
<https://www.ig.com/uk/glossary-trading-terms/ichimoku-cloud-definition>
- [4] Web stranica StockCharts: *Ichimoku Clouds*.
https://school.stockcharts.com/doku.php?id=technical_indicators:ichimoku_cloud
- [5] Python documentation:
<https://docs.python.org/3/>
- [6] Pandas documentation:
<https://pandas.pydata.org/docs/>
- [7] Numpy documentation:
<https://numpy.org/doc/>