

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Anže Brvar

**Strojno učenje in razvoj sistema za
avtomatsko trgovanje
na valutnem trgu**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Blaž Zupan

Ljubljana 2012

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00044/2012

Datum: 13.04.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANŽE BRVAR**

Naslov: **STROJNO UČENJE IN RAZVOJ SISTEMA ZA AVTOMATSKO TRGOVANJE NA VALUTNEM TRGU**
A MACHINE LEARNING-BASED AUTOMATIC CURRENCY TRADING SYSTEM

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

V diplomski nalogi preverite, ali je za trgovanje na valutnem trgu moč uporabiti modele strojnega učenja. V te namen izdelajte avtomatski trgovalni sistem, ki bo omogočal njegovo uporabo v okolju Metatrader4. Sistem naj sproti pridobiva podatke o trgu, te predstavi z ustreznimi primeri v atributnem zapisu, iz njih gradi napovedne modele in te uporabi v avtomatskem trgovanju. Sistem ovrednotite s tehniko vzorčenja in pri dejanskem trgovanju. Poročajte o njegovi uspešnosti.

Mentor:

prof. dr. Blaž Zupan



Dekan:

prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Anže Brvar, z vpisno številko **63090011**, sem avtor diplomskega dela z naslovom:

Strojno učenje in razvoj sistema za avtomatsko trgovanje na valutnem trgu

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Blaža Zupana,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 18. septembra 2012

Podpis avtorja:

Zahvaljujem se mentorju prof. dr. Blažu Zupanu za pomoč pri nastajanju diplomskega dela. Zahvaljujem se tudi Mihi Groharju in Andreju Krevlu, ki sta mi omogočila uporabo virtualnih računalnikov v laboratoriju za računalniške komunikacije. Prav tako se zahvaljujem vsem, ki ste mi kakorkoli pripomogli pri doseganju tega cilja.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Valutni trg in trgovanje	3
2.1	Opis valutnega trga	3
2.1.1	Finančni vzvod	7
2.2	Trgovalna platforma <i>Metatrader4</i>	7
2.2.1	Prikazi gibanja tečajev	8
2.2.2	Tehnični indikatorji	8
2.2.3	Funkcionalnosti platforme	16
3	Opis razvitega sistema in uporabljenih metod	21
3.1	Učni podatki	21
3.2	Opis sistema	23
3.2.1	Avtomatski trgovalni sistem	25
3.2.2	Klasifikacija	28
3.2.3	Površina pod krivuljo ROC	34
4	Eksperimentalni rezultati	37
4.1	Preverjanje z vzorčenjem	37
4.2	Ocenjevanje uspešnosti sistema “v živo”	41

KAZALO

5	Sklepne ugotovitve	47
A	<i>Python</i> skripta za napovedovanje gibanja vrednosti valutnih parov	51
B	Seznam uporabljenih atributov	55
C	Eksperimentalni rezultati preverjanja z vzorčenjem	59

Povzetek

Cilj te diplomske naloge je bil razviti avtomatski trgovalni sistem za trgovanje na valutnem trgu, ki bi odločanje o načinu trgovanja prepustil algoritmom strojnega učenja in z njimi razvitim napovednim modelom. Množico učnih podatkov smo pridobili iz vrednosti valutnih tečajev in tehničnih indikatorjev, ki opisujejo stanje na valutnem trgu. Izbrane algoritme strojnega učenja in njihove parametre smo ocenili z metodo preverjanja z vzorčenjem. Pripravili smo množico avtomatskih trgovalnih sistemov z različnimi nastavitvami in ocenili njihovo uspešnost na učnih podatkih. Najboljših devet avtomatskih trgovalnih sistemov smo nato preizkusili v trgovanju “v živo” na pravem trgu z namišljenim denarjem. Rezultati preverjanja z vzorčenjem so bili precej dobri, avtomatsko trgovanje “v živo” pa se je izkazalo za nekoliko težji problem, saj nam je uspelo izdelati le en dobičkonosen avtomatski trgovalni sistem. Na koncu smo predstavili možne izboljšave.

Ključne besede

valutno trgovanje, avtomatsko trgovanje, strojno učenje, časovni nizi, napovedovanje gibanja valutnih parov

Abstract

The main goal of this thesis was to develop an automated trading system for Forex trading, which would use machine learning methods and their prediction models for deciding about trading actions. A training data set was obtained from exchange rates and values of technical indicators, which describe conditions on currency market. We estimated selected machine learning algorithms and their parameters with validation with sampling. We have prepared a set of automated trading systems with various settings and evaluated their performance on training data sets. The best nine automated trading systems were tested in live trading on real market with test trading accounts. Experimental results were rather good. Automated live trading was proved to be more difficult problem, because we implemented only one profitable automated trading system. We also presented possible improvements.

Keywords

foreign exchange, automated trading, machine learning, time series, foreign exchange rates movement prediction

Poglavje 1

Uvod

Valutni trg (ali Forex) je sestavljen iz trgovanja, ki obsega celotno svetovno gospodarstvo in je sistem, ki deluje vse delovne dni, 24 ur na dan. Z dnevnim prometom okoli 4000 milijard USD je valutni trg najbolj likviden in svetovno razpršen trg, ki omogoča visoke dobičke in izgube. Vlagateljem je trgovanje omogočeno preko trgovalnih platform, ki v vsakem trenutku ponujajo vpogled v trenutno stanje vseh valutnih parov na valutnem trgu. Trgovalne platforme prikazujejo vrednosti tečajev valutnih parov in vrsto tehničnih indikatorjev. Večina vlagateljev uporablja ročno oddajo naročil za nakup ali prodajo valutnih parov, vedno bolj priljubljeno pa je avtomatsko trgovanje, kjer namesto človeka trguje program - avtomatski trgovalni sistem. Avtomatske trgovalne sisteme je mogoče napisati v orodju za pisanje programske kode v jeziku MQL4, ki je del platforme *Metatrader4*. Mnogo ljudi se že več kot desetletje ukvarja s problemom izdelave samostojnega, zanesljivega in dobičkonosnega avtomatskega trgovalnega sistema.

V tem diplomskem delu smo si zastavili cilj izdelati avtomatski trgovalni sistem, ki bi odločanje o načinu trgovanja prepustil algoritmom strojnega učenja. Želeli smo poiskati primerne učne algoritme in potrebne nastavitve avtomatskega trgovalnega sistema, ter jih ovrednotiti z uspešnostjo trgovanja. Prav tako je cilj diplomskega dela preizkusiti delovanje razvitega avtomatskega trgovalnega sistema v trgovanju v živo.

Razvili smo sistem, ki uporablja trgovalno platformo *Metatrader4* zgolj za izdelavo učnih podatkov in za oddajo naročil za nakup ali prodajo valutnih parov. Za učenje in izdelavo napovedi smo uporabili orodje za podatkovno analitiko in strojno učenje Orange [2] (dostopno na <http://orange.biolab.si>). Implementirali smo različne načine pridobivanja atributov, saj smo iz vrednosti 30-ih tehničnih indikatorjev en primer opisali s 195 atributi in diskretnim razredom, ki opisuje nadaljnje gibanje tečaja v pozitivno oz. negativno smer. Razvili smo skripto v programskem jeziku *Python*, ki, s pomočjo orodja Orange, za učenje in napovedovanje gibanja valutnih tečajev uporablja logistično regresijo, naključna drevesa ali naučeno kombiniranje z meta-učenjem. Metodi naključnih gozdov smo na učnih podatkih določili najoptimalnejše parametre, naučenem kombiniranju z meta-učenjem pa smo izbrali optimalen nabor osnovnih in meta-algoritmov, tako da so na učnih podatkih s preverjanjem z vzorčenjem dosegli čim boljšo točnost.

Izbrali smo nabore različnih parametrov in valutnih parov, ter na 108-ih avtomatskih trgovalnih sistemih ocenili uspešnost s preverjanjem z vzorčenjem. Nekaj najboljših avtomatskih trgovalnih sistemov smo preizkusili tudi v trgovanju na pravem trgu v realnem času na trgovalnih računih, ki so namenjeni preizkušanju trgovanja. Trgovalne sisteme, ki so trgovali z istimi valutnimi pari smo pognali vzporedno in s tem zagotovili, da so imeli vsi iste razmere in dobili realno sliko o uspešnosti uporabljenih metod strojnega učenja in nastavljenih parametrov sistema.

Poglavje 2

Valutni trg in trgovanje

Valutni trg (tudi Forex, FX) je največji in najlikvidnejši finančni trg na svetu. Kot že beseda "Forex" pove, gre za mednarodno menjavo med valutami (ang. *Foreign Exchange*). Posli na valutnem trgu so zgolj menjava ene valute v drugo.

2.1 Opis valutnega trga

Če želimo zamenjati valuto xxx v valuto yyy, to pomeni, da prodajamo valutni par xxxyyy, če pa želimo valuto yyy zamenjati za valuto xxx, pa pomeni, da kupujemo valutni par xxxyyy. Razmerje menjave ene enote valute xxx v valuto yyy se imenuje menjalni tečaj. Menjalni tečaj sestavljata dve vrednosti, prodajna cena (ang. *bid*) in nakupna cena (ang. *ask*). Najmanjša sprememba tečajev na valutnem trgu je ena točka (ang. *pip*). Večina tečajev se zapisuje na eno deset-tisočino natančno, zato je ena točka 0.0001.

Poglejmo si primer na valutnem paru EURUSD. Tečaj valutnega para je 1.1234/1.2346, prva je prodajna cena, druga pa nakupna cena. Če bi želeli zamenjati 1 EUR v USD, bi prodali valutni par po menjalnem tečaju $1 \text{ EUR} = 1.1234 \text{ USD}$. V primeru, da bi želeli zamenjati 1 USD v EUR, pa bi kupili valutni par EURUSD po menjalnem tečaju $1 \text{ EUR} = 1.1236 \text{ USD}$. V tem primeru je razlika med prodajno in nakupno ceno dve točki.

Leta 1973 so pri organizaciji ISO uvedli standardni enotni zapis valut ISO-4217 za uporabo v ekonomiji in bančništvu. Standard ISO-4217 predvideva tri-črkovni zapis valute in dve vrsti šest-črkovnega zapisa valutnega para: xxxyyy ali xxx/yyy. Prvi valuti pravimo osnovna (xxx), drugi pa obratna (yyy). Osnovna valuta je izražena relativno na obratno valuto. Celoten seznam valut je proti plačilu dostopen na spletni strani organizacije ISO, brezplačen seznam valut pa je na voljo na spletni strani http://www.commondatahub.com/live/geography/currency/iso_4217_currency_codes. 100,000 enot osnovne valute predstavlja osnovno mersko enoto za količino transakcij izbranega valutnega para (ang. *lot*).

Trgovanje na valutnem trgu poteka pet delovnih dni, od nedelje 20:00 GMT do petka 20:00 GMT. Za lažjo predstavo je en trgovalni dan prikazan na sliki 2.1. Trgovanje se prične v nedeljo zvečer ob 20:00 GMT v Wellingtonu (Nova Zelandija) in Sydneyu (Avstralija). Dve uri kasneje se odprejo finančne institucije v Tokiu, takrat prične delovati azijski finančni trg. Eno uro pred zaprtjem azijskega finančnega trga se odpre evropski finančni trg, pet ur kasneje pa še ameriški finančni trg z začetkom delovnega dne v New Yorku. Vsi finančni trgi so običajno odprti 8 ur. Med seboj se finančni trgi stalno prekrivajo, zato trgovanje poteka vse delovne dni, 24 ur na dan.

Po ocenah Bank for International Settlements je leta 2010 dnevni promet znašal 3980 milijard USD in se še povečuje [6]. Najbolj prometni tečaji v letu 2010 so bili EURUSD (28% dnevnih transakcij), USDJPY (14%) in GBPUSD (9%). Največ transakcij se opravi v povezavi z USD (85%) in EUR (39%). Največ transakcij je opravljenih v Združenem Kraljestvu (37%), sledijo ji Združene Države (18%) in Japonska (6%). Visoko likvidnost zagotavljajo največje svetovne banke, ki prispevajo 53% vseh transakcij. To so Deutsche Bank, Citi Bank, Barclays Capital, UBS AG in HSBC. Na trgu so prisotne tudi centralne banke, ki uravnavajo vrednost domače valute. Te banke so edina institucija, ki ni prisotna na trgu zaradi dobička, temveč zaradi kontrole nad inflacijo in zalogami denarja.

Ker ni centralnega središča, vrednost tečaja določajo posredniki trgovanja



Slika 2.1: Prikaz 24-urnega valutnega trgovanja in največjih trgovalnih obdobj na valutnem trgu.

(ang. *brokers*) med seboj, glede na trenutno ponudbo in povpraševanje. Trgovanje z valutami je razdeljeno na več nivojev, ki se razlikujejo v razliki med nakupno in prodajno ceno za vlagatelja (ang. *spread*). Posredniki med seboj določijo ceno, ki ji potem dodajo še dogovorjeno število točk. Višina razlike med nakupno in prodajno ceno je odvisna od kupne moči vlagatelja. Medbančni nivo, ki ga sestavlja le par največjih bank ima najnižjo razliko med nakupno in prodajno ceno (zaupen podatek, ki ni znan javnosti). Bolj ko se pomikamo proti malim vlagateljem, večja je ta razlika. Za malega vlagatelja ta navadno znaša med eno in tremi točkami (*pips*).

Vlagatelje lahko razdelimo na ne-špekulativne in špekulativne. Med ne-špekulativne spadajo centralne banke, ki uravnavaajo inflacijo in pretok denarja, ter mednarodne korporacije, ki potrebujejo tujo valuto za plačevanje storitev ali blaga. Špekulativni vlagatelji (približno 80% vseh vlagateljev) pa so vsi, ki kupujejo in prodajajo valute z namenom ustvariti dobiček. Med špekulativne vlagatelje spadajo tvegani skladi (ang. *hedge funds*), naložbena podjetja in mali vlagatelji. Trgovanje z valutami se neprestano povečuje zaradi pojava spletnih orodij, ki omogočajo elektronsko oddajo in obdelavo naročil. Ta orodja so znižala stroške transakcij, povečala število malih vlagateljev in s tem tudi povečala likvidnost in dnevni promet. Začelo se je množično uporabljati avtomatske trgovalne sisteme, kar je povečalo število hitrih (ang. *high-frequency*) transakcij, ki so odprte tudi samo par sekund.

Vrednost tečaja se določa med bankami na podlagi ponudbe in povpraševanja po bazni in obratni valuti. Veliko vlogo na ponudbo in povpraševanje imajo novice (politične ali ekonomske) in objave različnih poročil, ki se nanašajo na območje, kjer se valuta uporablja (indeksi brezposelnosti, BDP, inflacija, mesečno število prodanih hiš, mesečno število prodanih vozil, ...). Največje banke imajo v času objave novice ali nastopa političnega voditelja pomembno prednost in sicer vidijo reakcijo njihovih klientov na novico in to lahko izkoristijo.

2.1.1 Finančni vzvod

Najmanjša možna sprememba valutnega tečaja je kot že rečeno ena točka (0.0001). Trgovanje z vzvodom pomeni, da vlagatelj za nakup/prodajo nekega valutnega para plača samo nek delež vrednosti naročila. Na primer, če vlagatelj trguje z vzvodom 100:1, v primeru nakupa 10,000 enot dejansko plača samo 100 enot, ostalo mu posodi banka. V primeru, da je napravil dobiček, je dobiček 100-krat večji, kot bi bil sicer, vendar je morebitna izguba prav tako večja. Finančni vzvod se je uveljavil, ker imata prednosti oba, tako posrednik kot vlagatelj. Posrednik s 100-krat večjim poslom zasluži več z razliko med nakupno in prodajno vrednostjo valutnega para, vlagatelj pa pridobi možnost upravljati s 100-krat večjim kapitalom. Da sistem finančnega vzvoda deluje, potrebuje banka garancijo, da vlagatelj ne bo naredil izgube, ki je ne more pokriti, zato banka zahteva minimalno maržo (ang. *margin*). Marža je delež vrednosti odprtih poslov, ki mora biti manjši od vrednosti prostih sredstev na trgovalnem računu. Banka ta sredstva potrebuje za morebitno poplačilo negativnega stanja poslov. Takoj ko je prostih sredstev na trgovalnem računu manj, kot znaša marža, banka prične zapirati odprte posle in s tem sprostí sredstva za poplačilo odprtih negativnih poslov. Višina marže se razlikuje med bankami, prav tako se razlikuje med vrstami vlagateljev. Marža je ponavadi enaka količini sredstev, ki jih vlagatelj potrebuje pri finančnem vzvodu. Višino finančnega vzvoda si običajno vlagatelji lahko izberejo sami. V našem primeru smo uporabljali finančni vzvod 1:50.

2.2 Trgovalna platforma *Metatrader4*

Platforma za valutno trgovanje *Metatrader4* je produkt podjetja MetaQuotes Software Corp. Glavna naloga trgovalne platforme je povezava med vlagateljem in posrednikom. Posrednik preko platforme sporoča trenutne menjalne tečaje, vlagatelj pa spremlja dogajanje na valutnem trgu, upravlja s svojim trgovalnim računom, ročno oddaja naročila ali pa za to uporabi skripto ali avtomatski trgovalni sistem. V platformi je na voljo tudi urejevalnik

programske kode MQL4, ki je v pomoč pri pisanju skript in avtomatskih trgovalnih sistemov.

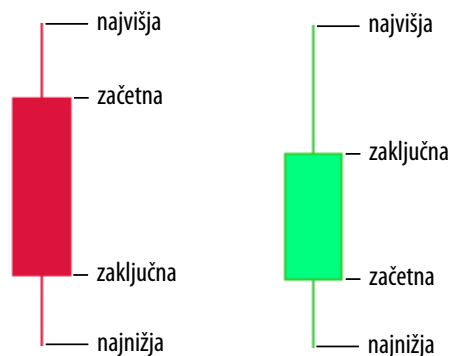
2.2.1 Prikazi gibanja tečajev

Gibanja tečajev pregledujemo v grafih, na katere lahko dodajamo tehnične indikatorje. Tovrstni grafični prikazi so razdeljeni na stolpce. Poznamo različne vrste prikazovanja valutnih tečajev. Najbolj razširjen je prikaz s svečkami, kjer je osnovni gradnik svečka (ang. *candlestick*). Svečka, ki predstavlja trenutno stanje in ni zaključena (se še oblikuje), je v platformi *Metatrader4* vedno označena kot *svečka[0]*. Vse ostale so označene kot odmik od trenutne svečke v preteklost. Vsaka svečka prikazuje štiri vrednosti: začetno, zaključno, najvišjo in najnižjo vrednost tečaja izbranega valutnega para v izbrani ločljivosti grafa. Na sliki 2.2 so prikazane te štiri vrednosti na primeru pozitivnega in negativnega premika tečaja. Ločljivost grafa je časovni razpon, ki ga prikazuje ena svečka. V platformi *Metatrader4* so na voljo ločljivosti 1, 5, 15 ali 30 minut, 1 ali 4 ure, 1 dan, 1 teden ali 1 mesec. Primer grafa, ki prikazuje valutni par EURUSD na enourni ločljivosti z dodanim tehničnim indikatorjem *Bollinger Bands*, je podan na sliki 2.3.

2.2.2 Tehnični indikatorji

Tehnični indikatorji so različni izračuni, ki temeljijo na vrednostih valutnih tečajev in količini opravljenih transakcij (tj. promet). Vrednosti tehničnih indikatorjev so v platformi *Metatrader4* prikazane na grafih, ki jih izrišemo na graf gibanja valutnega tečaja ali v nov graf, dejanske vrednosti tehničnih indikatorjev pa so shranjene v seznamih vrednosti. Vrednosti tehničnih indikatorjev vlagatelji uporabljajo kot signale za nakup ali prodajo valutnih parov. Vsi tehnični indikatorji uporabljeni v našem avtomatskem trgovalnem sistemu so na kratko opisani v nadaljevanju, podrobnosti pa so dostopne na <http://ta.mql4.com/indicators/>.

Drseča povprečja (ang. *Moving averages*) so najstarejša in najpopular-



Slika 2.2: Primer pozitivne in negativne svečke (ang. *candlestick*), ki prikazuje začetno, zaključno, najvišjo in najnižjo vrednost valutnega tečaja izbrane ločljivosti grafa.



Slika 2.3: Primer grafa, ki prikazuje valutni par EURUSD na enourni ločljivosti. Ena svečka v tem primeru predstavlja eno uro, torej začetno vrednost valutnega para ob npr. 12:00, zaključno vrednost ob 13:00, ter najvišjo in najnižjo vrednost valutnega para med 12:00 in 13:00. Na grafu je prikazan tudi tehnični indikator *Bollinger Bands*.

nejša metoda tehnične analize. Drseče povprečje je povprečna vrednost danega finančnega instrumenta (valutni tečaji, vrednostni papirji, borzni indeksi) v danem časovnem obdobju. Ko računamo drseče povprečje, moramo podati časovni okvir, v katerem računamo povprečno vrednost (npr. 10 ur) in izvor podatkov (npr. končna (ang. *close*) vrednost valutnega tečaja).

Acceleration/Deceleration (AC) prikazuje pospešek ali pojemek količine trgovanja in spremembe vrednosti izbranega valutnega para v izbranem času t . Podan je s spodnjim izrazom

$$AO_t - MA_t(AO, 5)$$

kjer je $MA_t(AO, 5)$ povprečje vrednosti indikatorja AO s časovnim intervalom 5 stolpcev.

Accumulation/Distribution (AD) prikazuje spremembe v vrednosti in prometu valutnega para. Vrednost v času t izračunamo z izrazom

$$(2 * zaključna_t - najnižja_t - najvišja_t) * \frac{\text{promet}_t}{\text{najvišja}_t - \text{najnižja}_t + AD_{t-1}}$$

kjer je AD_{t-1} vrednost indikatorja AD v času $t - 1$.

Average Directional Movement Index (ADX) je indikator, ki prikazuje trend gibanja vrednosti valutnega para. Vsebuje tri vrednosti (*main*, *plusdi* in *minusdi*). Vrednost *plusdi* in vrednost *minusdi* predstavljata najvišjo in najnižjo vrednost v času zadnjih t stolpcev. Vrednost *main* je razmerje povprečnih vrednosti *plusdi* in *minusdi*.

Alligator je indikator, ki v času t vsebuje tri vrednosti. To so povprečja na različnih časovnih okvirjih (npr. povprečje 13, 8 in 5 stolpcev). Vrednosti so poimenovane kot aligatorjeva čeljust (ang. *jaw*), zobje (ang. *teeth*) in ustnice (ang. *lips*). Osnovni koncept pri indikatorju aligator je: ko so vrednosti skupaj, aligator spi in trg miruje, ko pa so vrednosti narazen, se aligator prehranjuje in trg je aktiven. V skladu s tem konceptom naj bi vlagatelj predvideval, kdaj je čas za vstop in izstop iz trga.

Average True Range (ATR) prikazuje vrednost spodnjega izraza v izbranem času t

$$\max \left(\frac{\text{najvišja}_t}{\text{najnižja}_t}, \frac{\text{zaključna}_{t-1}}{\text{najvišja}_t}, \frac{\text{zaključna}_{t-1}}{\text{najnižja}_t} \right)$$

Indikator se običajno uporablja v povezavi z drsečim povprečjem.

Awesome Oscillator (AO) je razlika dveh drsečih povprečij z različnim časovnim oknom. Povprečne vrednosti se računajo iz vrednosti $(\text{najvišja}_t + \text{najnižja}_t)/2$ v času t .

Bollinger Bands (BOLL) indikator prikazuje tri vrednosti poimenovane ML , TL in BL . ML je navadno drseče povprečje, TL je vrednost $ML + 2\sigma$, BL pa $ML - 2\sigma$, kjer $\sigma = \mu - \text{zaključna}_t$.

Bulls/Bears sta indikatorja, ki prikazujeta drseče povprečje razlike med začetno in končno vrednostjo svečke podanega valutnega para. Indikator *Bulls* upošteva samo svečke, katerih vrednost se je povečala, *Bears* pa samo svečke, katerih vrednost se je znižala.

Market Facilitation Index (BWMFI) je indikator, s katerim vlagatelji prepoznava, kdaj je trg najbolj dinamičen (tj. kadar je prisotnih veliko vlagateljev in kadar se vrednost valutnega para hitro spreminja). Izračunamo ga z izrazom

$$\frac{\text{najvišja}_t - \text{najnižja}_t}{\text{promet}_t}$$

Commodity Channel Index (CCI) meri standardni odklon tipične vrednosti. Tipična vrednost se izračuna kot $(\text{najvišja}_t + \text{najnižja}_t + \text{končna}_t)/3$, vrednost indikatorja pa je

$$\frac{\sum_{i=t}^{t+n} \sigma_{\text{tipična}_i}}{n\sigma_{\text{tipična}_t}}$$

Vrednosti nad 100 pomenijo precenjenost valutnega para, vrednosti pod 100 pa podcenjenost valutnega para.

DeMarker je indikator, ki ima vrednosti na intervalu $[0, 1]$. Izračunan je tako, da med seboj primerja najvišje in najnižje vrednosti trenutne in prejšnje svečke in jih povpreči v podanem časovnem oknu. Vlagatelji pričakujejo porast vrednosti valutnega para, če je vrednost indikatorja *DeMarker* višja od 0.7, in padec, če je vrednost indikatorja manjša od 0.3.

Envelopes (ENVEL) indikator vsebuje dve vrednosti drsečega povprečja. Obe se izračunata na enak način, le da prvi prištejemo faktor d , drugi pa odštejemo faktor d . Če prikažemo vrednosti na grafu, dobimo dve enaki krivulji razmaknjeni za razdaljo $2d$. Ko vrednost valutnega para doseže eno od krivulj, naj bi to bil znak za nakup/prodajo valutnega para.

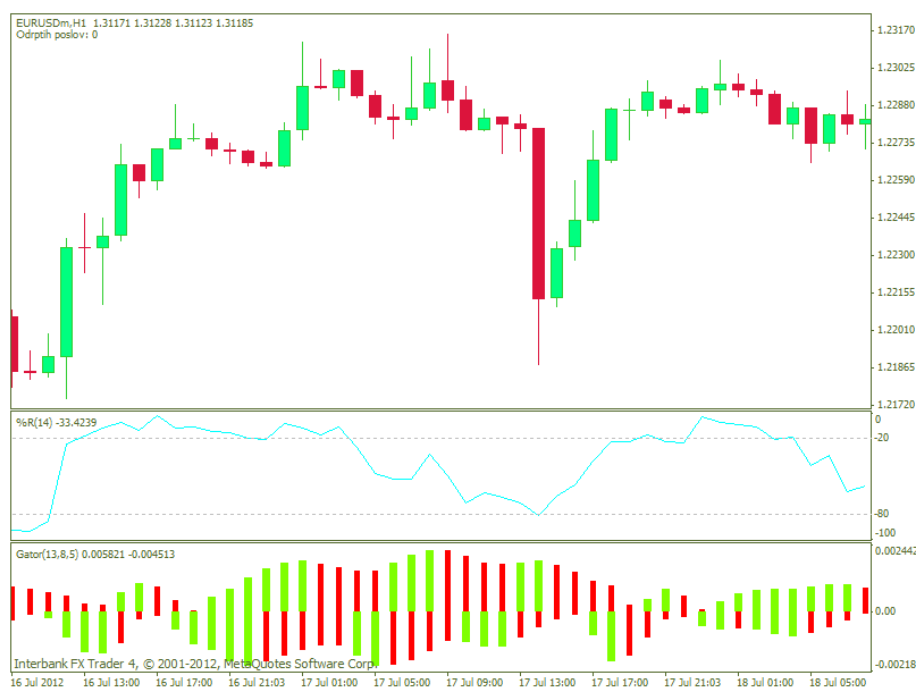
Force indikator prikazuje vrednosti podane s spodnjim izrazom

$$\frac{avg_{i=t}^{t+n}(\text{zaključna}_i)}{avg_{i=t}^{t+n}(\text{zaključna}_{i-1})} * \text{promet}_t$$

kjer je $avg_{i=t}^{t+n}(\text{zaključna}_i)$ povprečna vrednost zaključnih vrednosti od svečke v času t do $t + n$.

Fractals je indikator, ki ima ob izpolnjenem pogoju vrednost -1 ali 1. Pogoju za vrednost 1 je pet zaporednih pozitivnih svečk, med katerimi ima srednja najnižjo zaključno vrednost, pogoj za vrednost -1 pa je pet zaporednih negativnih svečk, med katerimi ima srednja najvišjo zaključno vrednost. Če pogoj ni izpolnjen, ima indikator vrednost 0.

Gator indikator vsebuje dve vrednosti, ki sta sestavljeni iz razlike povprečij vrednosti indikatorja *Aligator*. Prva se izračuna kot razlika med *Aligatorjevima* vrednostima 'čeljust' in 'zobje', druga vrednost pa je razlika med vrednostima 'zobje' in 'ustnice'. Običajno se prva vrednost indikatorja na grafu prikaže kot pozitivni stolpec, druga vrednost pa kot negativni stolpec. Izris vrednosti indikatorja na grafu je prikazan na sliki 2.4.



Slika 2.4: Primer grafa valutnega para EURUSD (zgoraj). Prikazana sta tudi tehnična indikatorja *Gator* (spodaj) in *Williams' Percent Range* (v sredini).

Ichimoku je indikator, ki ga je razvil japonec Ichimoku Sanjin. V času t vsebuje pet vrednosti. Tri izmed petih (*tenkansen*, *kijunsen* in *senkouspan B*) se izračunajo kot povprečje med najvišjo najvišjo vrednostjo svečke in najnižjo najnižjo vrednostjo svečke v danem časovnem oknu. Razlika med temi tremi vrednostmi je le v širini časovnega okna iskana najvišje in najnižje vrednosti. Četrta vrednost (*senkouspan A*) je povprečje vrednosti *tenkansen* in *kijunsen*, peta vrednost *chinkouspan* pa je vrednost zaključna _{t} premaknjena 26 stolpcev v preteklost. Vrednost *chinkouspan* je za uporabo pri napovedovanju neuporabna, ker je znana šele ob času $t + 26$, tako da nam pri generiranju napovednih podatkov (v času $t = 0$) ta vrednost še ni na voljo. Atributi, ki vsebujejo vrednost *chinkouspan* so vseeno zapisani v tabeli atributov, vendar so označeni kot meta-atributi, tako da se pri učenju in napovedovanju ne uporabljajo.

Moving Average Convergence/Divergence (MACD) vsebuje razmerje med dvema povprečji na različnih časovnih intervalih. Poleg razmerja indikator računa še vrednost *signal*, ki je povprečje indikatorja *MACD* na podanem intervalu.

Momentum indikator računa spremembo vrednosti valutnega para v podanem časovnem obdobju (npr. med $t = 0$ in $t = 15$).

Money Flow Index (MFI) prikazuje količino transakcij za izbran valutni par predstavljeno z deležem. Podoben je indikatorju *RSI*, le da *MFI* računa delež glede na promet podanega valutnega para.

OsMA vsebuje razliko vrednosti indikatorja *MACD* in njegove vrednosti *signal*.

On Balance Volume (OBV) sešteva promet podanega valutnega para. V primeru porasta vrednosti valutnega para se prejšnji vrednosti indikatorja *OBV* prišteje količina trgovanih lotov v času t , v primeru porenitve valutnega para pa se prejšnji vrednosti indikatorja *OBV* odšteje

količina trgovanih lotov v času t .

Relative Strength Index (RSI) računa delež med pozitivnimi in negativnimi spremembami vrednosti valutnega para

$$100 - \frac{100}{1 + \frac{P}{N}}$$

kjer je P povprečna pozitivna sprememba vrednosti valutnega para in N povprečna negativna sprememba vrednosti valutnega para na podanem časovnem intervalu.

Relative Vigor Index (RVI) je indikator, ki na podanem časovnem intervalu računa povprečje izraza

$$\frac{\text{zaključna}_t - \text{začetna}_t}{\text{najvišja}_t - \text{najnižja}_t}$$

Stop & Reverse (SAR) indikator se izračuna z izrazom

$$SAR_{t-1} + acc * (mhl - SAR_{t-1})$$

kjer je acc nek faktor in SAR_{t-1} vrednost indikatorja SAR v času $t - 1$. Vrednost mhl je v primeru pozitivnega povprečja cene valutnega para določena z najvišjo vrednostjo podanega časovnega intervala, v primeru negativnega povprečja pa je mhl najnižja vrednost valutnega para v podanem časovnem intervalu.

Standard deviation (StdDev) ali standardni odklon je razlika vrednosti v času t od povprečne vrednosti podanega časovnega intervala.

Stochastic je indikator, ki v podanem času t vsebuje vrednost, podano s spodnjim izrazom

$$\frac{\text{zaključna}_t - \text{najnižja}_t(n)}{\text{najvišja}_t(n) - \text{najnižja}_t(n)} * 100$$

kjer $\text{najnižja}_t(n)$ pomeni najnižjo najnižjo vrednost n vrednosti valutnega para in $\text{najvišja}_t(n)$ najvišjo najvišjo vrednost n vrednosti valutnega para v času od t do $t + n$.

Williams' Percent Range (WPR) je podoben indikatorju *Stochastic* in se izračuna z izrazom

$$\frac{\text{najvišja}_t(n) - \text{zaključna}_t}{\text{najvišja}_t(n) - \text{najnižja}_t(n)} * 100$$

Izris vrednosti indikatorja *WPR* na grafu je prikazan na sliki 2.4.

2.2.3 Funkcionalnosti platforme

V platformi *Metatrader4* so implementirane številne funkcionalnosti, ki jih na kratko opisujemo spodaj.

Pregledovanje valutnih tečajev. Platforma omogoča hrambo podatkov o gibanju tečajev na lokalnem disku. Vlagatelj te podatke pridobi od posrednika. Vsak valutni par ima več ločljivosti, kar pomeni, da ena svečka prikazuje nek časovni interval. Poleg gibanja tečajev je v platformi na voljo 30 tehničnih indikatorjev, ki so podlaga za tehnično analizo trga in jih večina vlagateljev uporablja pri odločanju o načinu trgovanja.

Ročno trgovanje. Vlagatelj lahko na podlagi grafičnih prikazov gibanja valutnih tečajev ročno odda naročilo za nakup ali prodajo poljubnega valutnega para. Platforma naročilo s podatki o valutnem paru, količini in ceni pošlje posredniku, ki ga obdela in sporoči uspešnost transakcije.

Skripte. Na valutnem trgu je pogosto hitra reakcija na razmere na trgu bistvenega pomena. Taka akcija je na primer zapiranje vseh odprtih naročil. Za tovrstne akcije se uporabljajo skripte, ki jih lahko vlagatelj napiše sam. Skripta je enkratni avtomatiziran postopek. Skripte lahko odpirajo ali zapirajo naročila, izvažajo podatke o valutnih tečajih v poljubni obliki ali celo pošiljajo elektronsko pošto. Primer skripte, napisane v jeziku *MQL4*, ki vključi zunanjo skripto in pokliče njeno metodo, izpiše nekaj informacij v konzolo platforme in na grafični prikaz, ter pošlje elektronsko sporočilo, je podan spodaj.

```
1 #include <OutputGenerator01.mq4>
2 int start() {
3     DumpTrain(4);
4     Print("Prostih sredstev na racunu: ",
5         AccountFreeMargin());
6     Comment("Odprih poslov: ", OrdersTotal());
7     SendMail("Sporocilo platforme MT4", "Trenutno stanje: "
8         + DoubleToStr(AccountBalance(), Digits));
9     return(0);
10 }
```

Avtomatski trgovalni sistemi. Platforma *Metatrader4* je zasnovana tako, da omogoča pisanje avtomatskih trgovalnih sistemov, ki so pravzaprav razširjene skripte. Tak sistem odloča, kdaj je pravi čas za vstop na trg, kdaj izstopiti in s kolikšno količino. Dostop ima do vseh podatkov in akcij, kot vlagatelj pri ročnem trgovanju. Avtomatski trgovalni sistemi običajno slonijo na neki strategiji, ki uporablja kombinacijo večih tehničnih indikatorjev. V našem trgovalnem sistemu to strategijo nadomešča strojno učenje, ki kot učne podatke prejme vrednosti vseh indikatorjev. Primer avtomatskega trgovalnega sistema je podan spodaj. Sistem odda naročilo za nakup valutnega para, če je razlika med trenutno in prejšnjo vrednostjo indikatorja *MACD* pozitivna, v nasprotnem primeru pa odda naročilo za prodajo valutnega para EURUSD. Primer avtomatskega trgovalnega sistema odda novo naročilo samo, če ni odprt še noben posel ($OrdersTotal() < 1$). Če bi hoteli, da bi spodnji sistem deloval, bi morali dopisati vsaj še logiko, ki zapira odprte posle.

```
1 int init() {
2     return(0);
3 }
4 int deinit() {
5     return(0);
6 }
```

```
7 int start() {
8   int MacdCur=iMACD(NULL,0,12,26,9,PRICE_CLOSE,MODEMAIN,0);
9   int MacdPrev=iMACD(NULL,0,12,26,9,PRICE_CLOSE,MODEMAIN,1);
10  int total=OrdersTotal();
11  int lots=1;
12  int ticket;
13  if(total<1){
14    if(AccountFreeMargin()<(1000*lots)){
15      Print("We have no money. Free Margin = ",
16           AccountFreeMargin());
17    }
18    //opening BUY order
19    if(MacdCur > MacdPrev){
20      ticket=OrderSend(Symbol(),OP_BUY,lots,Ask,3,0,
21                      Ask+100*Point,"macd sample",16384,0,Green);
22      if(ticket>0){
23        if(OrderSelect(ticket,SELECT_BY_TICKET,MODE_TRADES))
24          Print("BUY order opened: ",OrderOpenPrice());
25      }
26      else Print("Error opening BUY: ",GetLastError());
27      return(0);
28      //opening SELL order
29    }else{
30      ticket=OrderSend(Symbol(),OP_SELL,lots,Bid,3,0,
31                      Bid-100*Point,"macd sample",16384,0,Red);
32      if(ticket>0){
33        if(OrderSelect(ticket,SELECT_BY_TICKET,MODE_TRADES))
34          Print("SELL order opened: ",OrderOpenPrice());
35      }else Print("Error opening SELL: ",GetLastError());
36      return(0);
37    }
38    return(0);
39  }
40 }
```

Skripte in avtomatske trgovalne sisteme na platformi *Metatrader4* piše-
mo v skriptnem jeziku *MQL4* (*MetaQuotes Language 4*), ki je izdelan
posebej za platformo in je podoben programskemu jeziku *C*. Platforma
ponuja vrsto vgrajenih funkcij, ki se uporabljajo za oddajo naročil,
za dostop do informacij o trgovalnem računu, za dostop do vrednosti
tehničnih indikatorjev, ter za dostop do podatkov o preteklih giba-
njih tečajev. Dokumentacija jezika *MQL4* s podrobnim opisom vseh
vgrajenih funkcij je dostopna na <http://docs.mql4.com/>. Platforma
nudi tudi podporo komunikaciji z drugimi programi preko sistemskih
knjižnic *DLL*. V naši implementaciji avtomatskega trgovalnega sistema
smo raje uporabili enostavnejšo obliko komunikacije med programi,
preko datotek *LOG*.

Poglavje 3

Opis razvitega sistema in uporabljenih metod

Razvili smo sistem za avtomatsko trgovanje na valutnem trgu na trgovalni platformi *Metatrader4*. Razviti sistem iz podatkov o gibanju valutnega tečaja in vrednosti tehničnih indikatorjev pridobi učne podatke in izvaja trgovanje skladno z napovedmi strojnega učenja. Razvili smo tudi skripto v programskem jeziku *Python*, ki s pomočjo orodja *Orange* (<http://orange.biolab.si>) uporablja metode naključnih gozdov, logistične regresije in naučenega kombiniranja z meta-učenjem za napovedovanje prihodnjih gibanj vrednosti valutnih parov.

3.1 Učni podatki

Metode strojnega učenja potrebujejo učne podatke v fazi učenja oz. gradnje modela. Učni podatki so sestavljeni iz primerov. Vsak primer ima več atributov X in pripadajoč razred y . V našem primeru imamo 195 atributov in diskreten razred, ki ima vrednost -1 ali 1 . Negativen razred pomeni negativno razliko (oz. padec) valutnega tečaja med časom 0 in *odmik*, pozitiven razred pa pomeni pozitivno razliko (oz. porast) valutnega tečaja med časom 0 in *odmik*.

Učne podatke zapisuje skripta v trgovalni platformi *Metatrader4*. Skripta ima dostop do zgodovinskih vrednosti tečajev in do vrednosti indikatorjev, zato vse te podatke uporabi za opis stanja v nekem trenutku. Skripta podatke zapiše v obliki .tab datotek, ki so primerne za uporabo v orodju Orange. Datoteka vsebuje zaglavje in podatke ločene s tabulatorjem. Pri konfiguraciji sistema moramo določiti naslednje parametre sistema:

Valutni par je naziv valutnega para, sestavljen iz kratic obeh valut, ki ga sestavljata, npr. EURUSD.

Perioda ali ločljivost grafa je parameter podan v minutah, ki določa, koliko minut predstavlja ena svečka na grafu.

Odmik je številka vrednost, ki pove koliko časa vnaprej bomo napovedovali. Dejanski odmik v minutah je zmnožek parametrov *perioda* in *odmik*, npr. če je *perioda* 5 minut in *odmik* 6, napovedujemo za 30 minut vnaprej.

Med implementacijo sistema smo ugotovili, da je potrebno za natančnejše napovedi imeti na voljo čim večje število atributov. Želeli smo uporabiti okoli 200 atributov, zato smo poleg vrednosti v prejšnjem poglavju opisanih tehničnih indikatorjev dodatno uporabili tudi njihove izračunane vrednosti. Transformacije za te smo izvedli s spodaj opisanimi postopki:

Povprečna vrednost. Vsak indikator ima v nekem času t določeno vrednost. Povprečje predstavlja povprečno vrednost zadnjih n vrednosti indikatorja, kjer je n poljubno pozitivno število, navzgor omejeno z zalogo vrednosti (oz. številom stolpcev) izbranega grafa. Povprečje je najpreprostejša metoda zgoščevanja informacije. Poljubni indikator tako lahko opišemo z njegovo vrednostjo v času t in povprečji zadnjih 5, 10, 20, 50, 100 vrednosti.

Diskretizacija. Nekateri tehnični indikatorji (npr. *WPR* in *DeMarker*) prikazujejo vrednosti, ki so omejene na nek fiksni interval, npr. na intervalu $[0, 100]$. Ekonomisti jih uporabljajo predvsem v primerih ko

presežejo neko vrednost, npr. kupi valutni par, če je vrednost indikatorja nad 80, ali prodaj, če je pod 20. Takim indikatorjem je smiselno diskretizirati vrednosti tako, da jim določimo diskretne vrednosti, npr. ' <20 ', ' $20 > 80$ ', ' >80 '.

Razlika v času. Za nek tehnični indikator smo izbrali smiselno vrednost a . Nato smo od podanega časa t_1 nazaj iskali prvo vrednost indikatorja v času t_2 , ki je presegla vrednost a , kjer $t_1 < t_2$ (torej gledali smo nazaj v zgodovino). Razlika v času $t_2 - t_1$ predstavlja vrednost na novo pridobljenega atributa. Atribut predstavlja količino časa, ki je minil (glede na podan čas t_1), odkar je indikator nazadnje dosegel določeno vrednost a .

Več-vrednostni indikatorji. Tehnični indikator *Alligator* računa 3 vrednosti hkrati (glej sliko 3.1). Te vrednosti prikazujejo povprečje sprememb valutnih tečajev na različnih intervalih. Pri indikatorjih, ki računajo več vrednosti hkrati, lahko nov atribut pridobimo s tem, ko računamo razliko med posameznima dvema. Vsak indikator lahko izrišemo v času in v primeru več-vrednostnega indikatorja narišemo več krivulj. V primeru, da ima tehnični indikator take vrednosti, da se krivulje sekajo, lahko določimo nov atribut, ki vsebuje čas, ki je minil, odkar sta se krivulji nazadnje sekali. Primera takih tehničnih indikatorjev sta *Alligator* in *Ichimoku*.

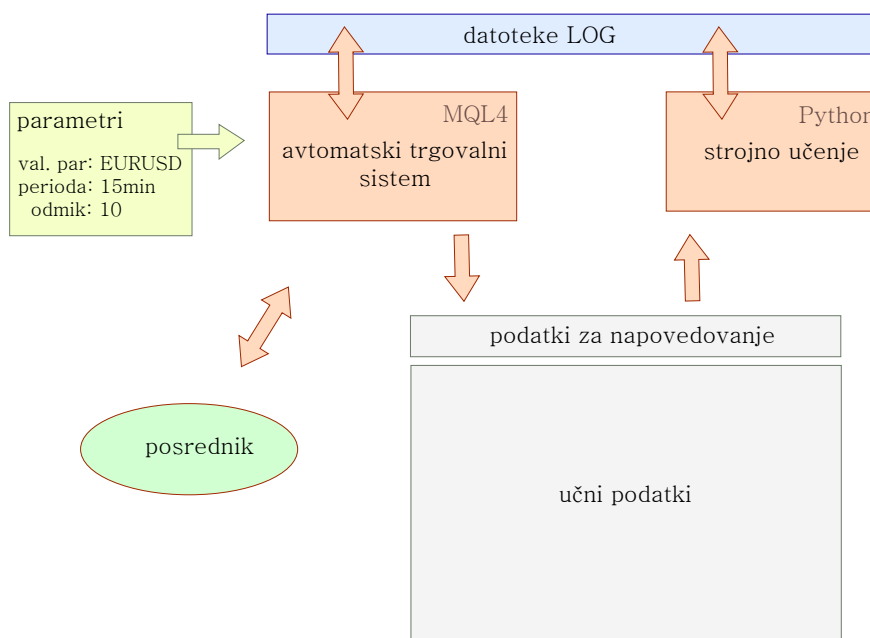
Seznam vseh atributov, ki smo jih uporabili, je podan v prilogi B.1.

3.2 Opis sistema

Sistem avtomatskega napovedovanja je razdeljen na dva programsko ločena dela. Prvi del je avtomatski trgovalni sistem, ki skrbi za izdelavo učnih in napovednih podatkov, ter za trgovanje. Drugi del je skripta v programskem jeziku *Python*, ki izvaja učenje in izdeluje napovedi. Komunikacija med avtomatskim trgovalnim sistemom in *Python* skripto lahko poteka na dva načina,



Slika 3.1: Poleg vrednosti valutnega para EURUSD na enourni ločljivosti je prikazan tehnični indikator *Alligator*, ki prikazuje 3 krivulje hkrati. Vse tri krivulje prikazujejo drseče povprečje gibanja vrednosti valutnega para. Zelena krivulja prikazuje povprečje 5 stolpcev, rdeča prikazuje povprečje 8 stolpcev, modra pa povprečje 13 stolpcev.



Slika 3.2: Arhitektura razvitega sistema.

preko sistemskih knjižnic *DLL* ali preko tekstovnih datotek *LOG*. Implementirali smo enostavnejšo različico komunikacije med programi, preko datotek *LOG*. Arhitektura razvitega sistema je prikazana na sliki 3.2.

3.2.1 Avtomatski trgovalni sistem

Avtomatski trgovalni sistem je napisan v jeziku *MQL4*, ki se uporablja za pisanje skript in avtomatskih trgovalnih sistemov, ki tečejo na platformi *Metatrader4*. Trgovalni sistem se od skripte razlikuje v tem, da mora vsebovati tri funkcije: `init()`, `deinit()` in `start()`.

`init()` se izvede samo enkrat, ob zagonu avtomatskega trgovalnega sistema.

V našem primeru v tej funkciji pokličemo skripto, ki zapiše učne podatke in ustvari potrebne datoteke *LOG*.

`deinit()` se izvede samo ob prekinitvi delovanja sistema.

`start()` se izvede ob prihodu nove informacije o vrednosti valutnega tečaja. Ta informacija lahko pride tudi večkrat na sekundo. Funkcija `start()` mora vsebovati logiko, ki se na podlagi nove vrednosti valutnega tečaja odloča, kaj storiti. V našem primeru ob zaključku svečke na grafu (t.j. dogodek, ko se zaključi oblikovanje svečke, npr. na 5-minutnem grafu se svečka zaključi ob 12:05, 12:10, 12:15, itd.) v eno od datotek *LOG* zapiše napovedne podatke, ki se nanašajo na ravnokar zaključeno svečko in jih skripta programskem jeziku *Python* vzame za primer, ki ga je potrebno klasificirati.

Pseudokoda, ki opisuje delovanje avtomatskega trgovalnega sistema:

```
1 int init () {
2     tmpCas = trenutenCas (); casZapisa = 0; casNapovedi = 0;
3     odmikVSekundah = 60 * locljivost () * odmik;
4     zapisiUcnePodatke (locljivost , odmik);
5     zapisiNapovednePodatke (locljivost , odmik);
6     return (0);
7 }
8
9 int deinit () {
10    zapriVseOdprtePosle ();
11    return (0);
12 }
13
14 int start () {
15    /* Zapisovanje napovednih podatkov. */
16    if (novaSveckaOblikovana) {
17        zapisiNapovednePodatke ();
18        if (steviloOdprtihPoslov () < 1) {
19            zapisiUkazZaIzdelavoNapovedi ();
20            casZapisa = trenutenCas ();
21        } else {
22            /* ce je napoved odprta dlje , kot je odmik , jo zapre */
23            if ((casZapisa + odmikVSekundah) < trenutenCas ()) {
24                zapriVseOdprtePosle ();
25            }
26        }
27    }
28 }
```

```
26     }
27 }
28
29 /* Branje napovedi in izvedba narocila. */
30 napoved, casNapovedi = preberiDatotekoLogNapoved();
31 if(casNapovedi > tmpCas){
32     tmpCas = casNapovedi;
33     if (napoved>0){
34         kupi(odmik, kolicina);
35     }else{
36         prodaj(odmik, kolicina);
37     }
38 }
39 return(0);
40 }
```

avtomatski-trgovalni-sistem.txt

Ob zagonu avtomatskega trgovalnega sistema se torej oblikujejo učni podatki, nato pa se ob vsaki spremembi vrednosti valutnega tečaja pokliče funkcija `start()`, ki preveri, če je oblikovanje nove svečke zaključeno. V primeru nove zaključene svečke se prične oblikovati nova svečka, obenem pa se takrat pokliče skripta, ki zapiše napovedne podatke, ki pripadajo ravnokar zaključeni svečki. Nato funkcija `start()` v datotekah *LOG* preverja, če je skripta v *Python*-u zapisala novo napoved. V tem času skripta v *Python*-u vzame pravkar zapisane napovedne podatke in napove vrednost razreda, ki jo zapiše v datoteko *LOG* in zraven doda čas napovedi. Ko avtomatski trgovalni sistem ugotovi, da je čas napovedi novejši od zadnje napovedi, pokliče ustrezne funkcije za nakup ali prodajo valutnega para.

Platforma omogoča več odprtih poslov naenkrat, vendar morajo biti vsi iste smeri. Nakup in prodaja ob istem času je onemogočena, zato moramo pred nakupom valutnega para zapreti vse odprte posle, ki smo jih prodali in obratno. Trgovalni sistem zaradi te omejitve pred oddajo naročila posle zapira, ko preteče s parametrom *odmik* določena količina časa. Naenkrat je tako odprt samo en posel istega valutnega para, vsak posel pa je odprt

najdlje $odmik \times perioda$ minut, potem pa ga trgovalni sistem, ne glede na dobiček ali izgubo, zapre.

3.2.2 Klasifikacija

Klasifikacija je napovedovanje diskretnih vrednosti oz. uvrščanje v diskretne razrede (npr. Da/Ne ali Sončno/Deževno/Oblačno/Megleno). V našem primeru smo se zaradi narave problema odločili za razred z dvema diskretnimi vrednostmi, in sicer:

- -1 , ki pomeni, da bo cena tečaja od nakupa do časa $odmik$ padla
- $+1$, ki pomeni, da bo cena tečaja od nakupa do časa $odmik$ narasla

Za izdelavo napovednih klasifikacijskih modelov smo uporabili metode strojnega učenja, ki za učenje prejmejo učne podatke \mathbf{X}, \mathbf{y} . Ta način strojnega učenja se imenuje nadzorovano učenje, kar pomeni, da algoritmom v fazi učenja podamo primere X s pripadajočimi razredi y . Metode se v fazi učenja učijo modela, s katerim kasneje neznan primer na naučen način uvrstijo v razred.

Vse uporabljene metode za strojno učenje so implementirane v knjižnici *Orange*, do katere dostopamo iz programskega jezika *Python*. Te metode so logika za odločanje o nakupu ali prodaji tečaja, ki je iz funkcije `start()`, ki je del avtomatskega trgovalnega sistema, predstavljena v *Python* skripti (podana v prilogi A). Skripta v *Python*-u ob zagonu naloži učne podatke in jih poda algoritmu za strojno učenje, nato pa se neprestano vrta v zanki in v datoteki *LOG* preverja, če je avtomatski trgovalni sistem že izdelal nov primer brez razreda, ki ga je potrebno klasificirati. Ko izdelava novo napoved, jo zapiše v datoteko *LOG*. Ob kakršnikoli spremembi učnih podatkov, npr. ko avtomatski trgovalni sistem izdelava novejša učna podatka, skripta samodejno naloži nove učne podatke in nadaljuje izvajanje. V nadaljevanju so na kratko opisani uporabljeni algoritmi za nadzorovano učenje, ki vključujejo naključna drevesa, logistično regresijo in združevanje napovedi različnih modelov z metodo naučenega kombiniranja z meta-učenjem.

Naivni Bayesov klasifikator

Naivni Bayesov klasifikator predpostavlja pogojno neodvisnost vrednosti atributov pri danem razredu:

$$P(v_1, v_2, \dots, v_n | c_k) = \prod_i P(v_i | c_k)$$

kjer je v_i vrednost i -tega atributa podanega primera in c_k vrednost k -tega razreda pri k različnih vrednostih razreda. Osnovno formulo naivnega Bayesovega klasifikatorja izpeljemo s pomočjo Bayesovega pravila:

$$P(c_k | v_1, v_2, \dots, v_n) = P(c_k) \prod_i \frac{P(c_k | v_i)}{P(c_k)}$$

Učni algoritem mora s pomočjo učnih podatkov aproksimirati verjetnosti na desni strani enačbe. Znanje, ki ga naivni Bayesov klasifikator pridobi z učenjem, so aproksimacije apriornih verjetnosti razredov $P(c_k)$, $k = 1 \dots n_0$ in pogojne verjetnosti razredov pri dani vrednosti v_i atributa A_i : $P(c_k | v_i)$, $k = 1 \dots n_0$, $i = 1 \dots a$. Pri učenju lahko primere, ki so brez vrednosti za dani atribut, izpustimo iz formule. Naivni Bayesov klasifikator uporabi vse attribute, katerih vrednosti so podane za nek primer, ki ga klasificiramo, tiste, ki jih ne poznamo, pa ignorira. Naivni Bayesov klasifikator brez težav uporabljamo z diskretnimi atributi, v primeru zveznih atributov, pa moramo prej poskrbeti še za diskretizacijo [5]. V našem sistemu smo naivni Bayesov klasifikator uporabili kot meta-učni algoritem pri metodi naučenega kombiniranja z meta-učenjem.

K najbližjih sosedov

Metoda k najbližjih sosedov spada med lene metode, saj si v fazi učenja samo zapomni celotno množico učnih primerov, medtem ko glavno delo opravi uvrščanje. Uvrščanje poteka tako, da metoda novemu primeru X med učnimi podatki poišče k najbližjih primerov X_1, \dots, X_k , podobnih po vrednostih atributov in klasificira tako, da napove večinski razred (razred, ki mu pripada največ izmed k najbližjih sosedov). Število k je parameter, ki je ponavadi

liho število. Vrednost parametra k je potrebno določiti eksperimentalno, saj lahko prevelik k povzroči, da kot najbližje sosede izberemo primere, ki sploh niso podobni danemu primeru, premajhen k pa v primeru šumnih učnih podatkov povzroči napačno klasifikacijo [5].

Izboljšane različice metode k najbližjih sosedov upoštevajo razdaljo sosedov od danega primera. Taka je tudi implementacija algoritma v knjižnici *Orange*. Razdalja je utež, ki se upošteva pri določanju razreda danemu primeru. Bližji sosedi imajo tako večjo težo. V knjižnici *Orange* je utež vsakega izmed učnih primerov naslednja:

$$\exp\left(\frac{-r^2}{s^2}\right)$$

kjer je r razdalja od učnega primera do danega primera, s pa je glede na učno množico izbran tako, da ima najbolj oddaljen učni primer utež enako 0.001. Z dodanimi utežmi lahko izberemo tudi večji k , saj v primeru izbire napačnih primerov za sosede zaradi manjše uteži le-te upoštevamo manj kot bližnje sosede. Zaradi implementacije metode k najbližjih sosedov z utežmi smo se odločili za parameter $k = 500$. Metodo smo uporabili kot eno izmed osnovnih učnih algoritmov pri naučenem kombiniranju z meta-učenjem.

Naključna drevesa

Naključna drevesa spadajo med ansambelske (ang. *ensemble*) metode, ki združujejo večje število enostavnejših klasifikatorjev. Te metode izdelajo večje število enostavnih klasifikatorjev, katerih napovedi nato združimo. Točnost združenih napovedi je večinoma precej boljša od najboljše posamezne metode izmed uporabljenih. Pogoj za združevanje enostavnih klasifikatorjev je, da so med seboj različni. Enostavne klasifikatorje nato povprečimo in zaradi zakona velikih števil dobimo optimalno klasifikacijo.

Naključna drevesa so neke vrste izboljšava metode bagging (ang. *bootstrap aggregating*), ki s pomočjo metode stremena (ang. *bootstrap*) vsakemu odločitvenemu drevesu izbere rahlo drugačno učno množico [1]. Zgrajena odločitvena drevesa so zato med seboj različna. Pri baggingu vsako drevo

glasuje tako, da napove svoj razred, razred z največ glasovi je izbran kot končni razred.

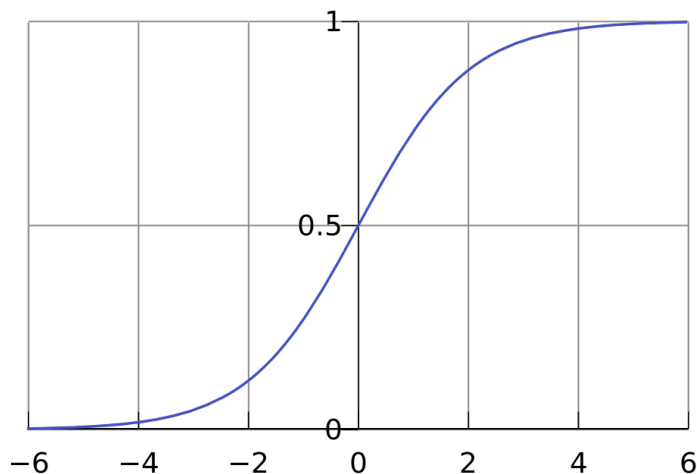
Naključna drevesa se od bagginga razlikujejo le v načinu gradnje dreves, in sicer v vsakem vozlišču vsakega izmed dreves izberemo naključnih m atributov ($m < a$, a - št. atributov), izmed katerih najbolj informativni atribut izberemo za vozlišče drevesa. Naključna drevesa se lahko uporabljajo tudi za ocenjevanje atributov, računanje podobnosti med primeri in iskanje skupin, vendar v našem primeru uporabljamo klasifikacijo - uvrščanje v razred.

Pri uporabi naključnih dreves je potrebno določiti dva parametra: število dreves n in št. atributov m , ki jih algoritem v postopku gradnje dreves naključno izbere za ožji izbor v vsakem vozlišču. Število dreves lahko izberemo zgolj premajhno, zato smo po nekaj začetnih poskusih izbrali dovolj velik $n = 100$. Pri izboru števila atributov m Breiman [1] priporoča $m = \log(a)$ ali $m = \sqrt{a}$ (a je št. atributov), kar je tudi privzeta vrednost za m v orodju *Orange*. Ker smo želeli izbrati najbolj optimalno vrednost m , smo s preverjanjem z vzorčenjem preverili točnost napovedi z različnim številom atributov m .

Logistična regresija

Logistična regresija spada v družino splošnih linearnih modelov (regresijskih modelov). Metoda je podobna linearni regresiji, saj obe iščeta model, ki se bo najboljše prilegal učnim podatkom. Razlika med njima je ta, da linearna regresija napoveduje zvezne vrednosti, logistična regresija pa diskretne razrede, v našem primeru razred z binarno vrednostjo 0 ali 1 (oz. -1 ali 1).

Iščemo torej model, oz. hipotezo $h(\theta)$, katere parametri $\theta_1, \theta_2, \dots, \theta_n$ določajo obliko funkcije, ki se čim boljše prilega učnim podatkom. Naloga logistične regresije je iz naučenega modela napovedati verjetnost, da vhodnim atributom x pripada vrednost razreda 1. Za hipotezo logistične regresije uporabimo *sigmoido* (ali *logistično funkcijo*, na sliki 3.3), ki se pri $-\infty$ asimptotično približuje vrednosti 0 in pri $+\infty$ vrednosti 1, ter s tem omeji izhod hipoteze na $0 \leq h_\theta(x) < 1$.



Slika 3.3: *Logistična funkcija* ali *sigmoida*, ki je osnova za hipotezo $h_{\theta}(x)$ pri logistični regresiji. Zaradi svoje oblike omeji izhod logistične regresije na $0 \leq h_{\theta}(x) < 1$.

Funkcija napake meri točnost hipoteze $h_{\theta}(x)$ in cilj učenja modela je minimizirati funkcijo napake (ang. *cost function*). Funkcija napake je izbrana tako, da bolj kot hipoteza $h_{\theta}(x)$ napačno napove verjetnost, večjo napako vrne in bolj podobno vrednost pravi vrednosti y , kot jo hipoteza napove, manjšo napako vrne. Za učenje modela se tipično uporablja gradientna metoda (ang. *gradient descent*), ki minimizira funkcijo napake. Na začetku izbere naključne parametre $\theta_1, \theta_2, \dots, \theta_n$ in jih nato z majhnimi koraki popravlja v smeri zmanjševanja napake. Način, kako poiskati to smer, je računanje odvoda (tangente) na funkcijo napake. Naklon tangente je odvod v tej točki in pomeni smer v katero se moramo premakniti. Gradientna metoda tako išče najoptimalnejšo funkcijo $h_{\theta}(x)$, ki med seboj ločuje vrednosti razreda. Pravimo ji tudi ločitvena meja (decision boundary) [4].

Kot že rečeno, je logistična regresija klasifikator in napoveduje razred, zato moramo iz zvezne verjetnosti, da ima razred vrednost 1, določiti diskreten razred. To dosežemo tako, da postavimo neko mejo verjetnosti, kjer primer še spada v pozitiven razred, ponavadi z mejo 0.5. Interval < 0.5 tako

določa diskretno vrednosti razreda 0, ≥ 0.5 pa diskretno vrednost 1.

Naučeno kombiniranje z meta-učenjem

Naučeno kombiniranje z meta-učenjem (ang. *stacking*) je tako kot metoda naključnih dreves ansambelska metoda, ki združuje več klasifikatorjev. Za razliko od naključnih dreves, kjer so vsi klasifikatorji odločitvena drevesa, pri naučenem kombiniranju združujemo različne učne algoritme. Osnovni učni algoritmi se ločeno naučijo klasifikacijskih modelov in vrnejo verjetnosti klasifikacije v vsakega od razredov. Nato te verjetnosti razredov posameznega klasifikatorja združimo s pomočjo meta-učnega algoritma. Meta-učni algoritem je običajen učni algoritem, le da kot primere učne množice uporabi verjetnosti razredov vsakega od klasifikatorjev, za razred pa pravilni razred. Nato se nauči modela in s tem pravilnega kombiniranja osnovnih učnih algoritmov.

Ob napovedovanju razreda y za neznan primer X vsi osnovni učni algoritmi izdelajo svoje napovedi v obliki verjetnosti razredov (algoritem j izdelava napovedi p_j^i , i razredov), nato pa meta-učni algoritem te verjetnosti vzame kot primer p in model napove kombinacijo napovedi osnovnih učnih algoritmov in izbere končni razred y .

Veliko raziskovalcev s tega področja dosega bistveno boljše rezultate z metodo naučenega kombiniranja z meta-učenjem v primerjavi z metodami, ki izglasujejo končni razred (bagging, naključna drevesa) [3]. V našem primeru smo kot meta-učni algoritem uporabili naivni Bayesov klasifikator, ki je privzet meta-učni algoritem naučenega kombiniranja z meta-učenjem v orodju *Orange*. Izvedli smo tudi eksperiment, da bi ugotovili katera kombinacija osnovnih algoritmov in kateri meta-učni algoritem napoveduje bolje (rezultati so podani v poglavju 4.1). Na podlagi eksperimentalnih rezultatov smo za nadaljne eksperimente izbrali naslednje osnovne učne algoritme:

- logistična regresija,
- naključna drevesa,

- k najbližjih sosedov.

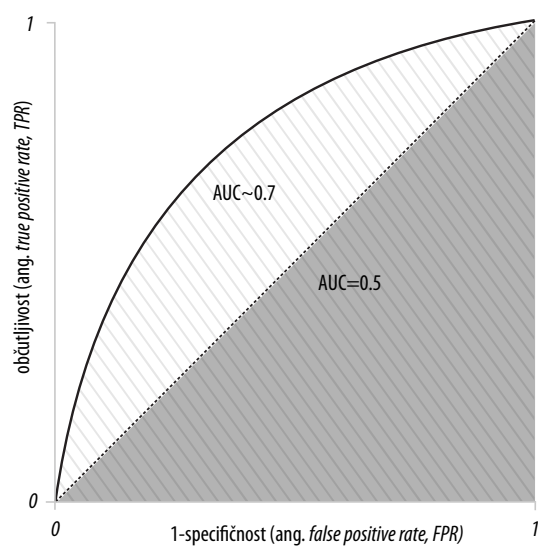
3.2.3 Površina pod krivuljo ROC

Za mero točnosti klasifikatorjev smo izbrali površino pod krivuljo ROC (ang. *area under ROC curve* ali *AUC*). Površina pod krivuljo ROC je verjetnost, da za naključno izbrani primer s pozitivno vrednostjo razreda X^+ in naključno izbran primer z negativno vrednostjo razreda X^- velja:

$$P(+, X^+) > P(+, X^-)$$

oz. verjetnost, da bo primer X^+ klasificiran kot pozitiven je večja od verjetnosti, da bo primer X^- klasificiran kot pozitiven.

Pri klasifikaciji z dvovrednostnim razredom sta se uveljavili dve meri, občutljivost (ang. *sensitivity*) in specifičnost (ang. *specificity*). Občutljivost predstavlja delež pravilno klasificiranih pozitivnih primerov, specifičnost pa predstavlja delež pravilno klasificiranih negativnih primerov. Krivulja ROC prikazuje odvisnosti med $1 - \text{specifičnostjo}$ (ang. *false positive rate, FPR*) in občutljivostjo (ang. *true positive rate, TPR*), primer je prikazan na sliki 3.4. Ker klasifikatorji napovedujejo verjetnost, da primer X spada v pozitiven razred, lahko krivuljo ROC enostavno narišemo tako, da spreminjamo mejo verjetnosti, da primer spada v pozitiven razred. Ob vsaki spremembi meje na grafu označimo točko, ki jo določata TPR in FPR , vse točke med seboj povežemo in dobimo krivuljo ROC. Večja kot je površina pod krivuljo ROC, bolj kvaliteten klasifikator imamo, oz. bolje ločuje med pozitivnimi in negativnimi razredi. Vrednost površine pod krivuljo ROC, ki je enaka 0.5, pomeni, da klasifikator ne ločuje med pozitivnimi in negativnimi vrednostmi razreda, bolj kvalitetni klasifikatorji pa dosegajo vrednosti večje od 0.7.



Slika 3.4: Krivulja ROC, ki prikazuje odvisnosti med specifičnostjo in občutljivostjo. Ploščino pod krivuljo uporabljamo kot mero točnosti napovedovanja.

Poglavje 4

Eksperimentalni rezultati

Uspešnost trgovanja z razvitim sistemom smo ocenjevali na dva načina, na učnih podatkih z metodo preverjanja z vzorčenjem in s trgovanjem v živo. Pri obeh načinih ocenjevanja smo izdelali več različic avtomatskih trgovalnih sistemov, ki so imeli različne nastavitve (izbran valutni par, odmik, perioda, uporabljen napovedni model) in jih med seboj primerjali.

4.1 Preverjanje z vzorčenjem

Množica podatkov o vrednostih valutnih tečajev spada med časovne vrste. Lastnost časovnih vrst je vrstni red podatkov, saj imamo v nekem trenutku informacije iz preteklosti, napovedujemo pa, kaj se bo zgodilo v prihodnosti. To moramo upoštevati tudi pri ocenjevanju uspešnosti napovednih algoritmov, in sicer morajo biti učni primeri v času pred testnimi primeri. Zato smo v prvi iteraciji preverjanja z vzorčenjem testne primere (vzorec) izbrali tako, da so ti primeri predstavljali 10% najnovejših učnih primerov. V vseh naslednjih iteracijah preverjanja z vzorčenjem smo testno množico iz prejšnje iteracije zavrgli, učno množico iz prejšnje iteracije pa zopet vzorčili tako, da je 10% najnovejših primerov testnih, ostali pa so nova učna množica. Razmerje med velikostjo učne in testne množice je bilo v vseh eksperimentih 90%/10%.

Metodi naključnih dreves moramo pri učenju podati dva parametra: število dreves in število naključnih atributov m , iz katerih v fazi učenja v vozlišče drevesa vključimo najbolj informativnega. Število dreves lahko izberemo zgolj premajhno, pri večjih številih pa se točnost algoritma ustali pri neki vrednosti, zato ni smiselno posebej izbirati tega parametra, le da je dovolj velik. Drugače pa je z izbiro m . Če je m premajhen, so drevesa premalo točna, prevelik m pa lahko povzroči preveliko koreliranost (podobnost) med drevesi, tako da z glasovanjem ne pridobimo kaj dosti. Različni raziskovalci priporočajo različne vrednosti: $m = 1, 2, \log(a)$ ali \sqrt{a} (a - število atributov ki opisujejo primer). Ker je to odvisno od problemske domene, smo izvedli primerjavo med istim algoritmom na istih podatkih z različnimi vrednostmi m . V tabeli 4.1 so prikazane točnosti in pripadajoči parameter m nekaj najboljših in nekaj najslabših algoritmov. Za nadaljnje poskuse smo tako uporabili $m = 8$.

Tabela 4.1: Primerjava točnosti napovedovanja (AUC) z metodo naključnih dreves z različnimi parametri m . Prikazani so najboljši in najslabši štirje algoritmi.

m	AUC
8	0.527
11	0.525
19	0.524
17	0.522
5	0.510
4	0.508
3	0.507
1	0.500

Zanimalo nas je tudi, katere osnovne in meta-učne algoritme uporabiti pri naučenem kombiniranju z meta-učenjem (ang. *stacking*), da bo njegova klasifikacijska točnost največja. Uporabili smo pet različnih osnovnih algo-

ritmov in z njimi izvedli preverjanje z vzorčenjem z velikostjo testne množice 10%. Pri preverjanju smo upoštevali, da morajo biti učni podatki starejši od testnih podatkov. Iz rezultatov, prikazanih v tabeli 4.2 lahko razberemo, da prisotnost osnovnega algoritma Majority ni izboljšala točnosti napovedovanja. Poskusili smo tudi različne meta-učne algoritme. Poleg privzetega v orodju *Orange* (naivni Bayesov klasifikator), smo poskusili še z logistično regresijo. Rezultati preverjanja z vzorčenjem so pri uporabi logistične regresije za meta-učni algoritem, le malenkost slabši od naivnega Bayesovega klasifikatorja. Za nadaljnje poskuse smo izbrali privzeti meta-učni algoritem in najboljšo kombinacijo osnovnih učnih algoritmov. To so logistična regresija, naključna drevesa in k najbližjih sosedov.

Naslednji korak je bilo primerjanje uspešnosti večih avtomatskih trgovalnih sistemov z različnimi parametri. Ti parametri so **valutni par**, časovni **odmik** napovedovanja, **perioda** grafa in **algoritem** strojnega učenja. Vseh kombinacij je bilo preveč, zato smo se omejili na štiri valutne pare, tri periode grafa (1min, 15min in 1h), tri različne odmike (4, 10 in 20 svečk vnaprej) in tri različne napovedne modele. Izmed izbranih valutnih parov so trije najbolj prometni in eden manj prometen (AUDNZD). Nabor parametrov je podan v tabeli 4.3. S preverjanjem z vzorčenjem smo ocenili uspešnost vsakega od 108-ih sistemov. Za pošteno primerjavo med sistemi smo velikost učnih podatkov pri vsakem sistemu omejili na 5000 primerov.

V tabeli 4.4 so prikazane vrednosti površine pod krivuljo ROC in parametri najboljših 30 napovednih sistemov. Presenetljivo je, da najpomembnejšega valutnega para EURUSD ni med boljšimi. Prav tako med boljšimi ni valutnega para AUDNZD, ki je najmanj prometen izmed štirih prisotnih v eksperimentu. To morda pomeni, da se naš sistem napovedovanja ne obnese dobro pri prenasičenem trgu, kjer se vse odvija hitro in prav tako slabo na nizko-prometnem trgu. Opazimo tudi, da so v vrhu sistemi s 15-minutnimi periodami grafa. Če med seboj primerjamo napovedne modele, so najboljšo točnost dosegli sistemi, ki so uporabljali naučeno kombiniranje z meta-učenjem. Odmik napovedovanja ne predstavlja posebno pomembnega

Tabela 4.2: Eksperimentalni rezultati preverjanja z vzorčenjem nabora osnovnih algoritmov pri naučenem kombiniranju z meta-učenjem. Prikazane so AUC vrednosti pri uporabi logistične regresije in naivnega Bayesovega klasifikatorja kot meta-učni algoritem. Znak **X** označuje prisotnost osnovnega algoritma v naboru osnovnih algoritmov, krepka vrednost AUC pa je najvišja vrednost med vsemi kombinacijami učnih algoritmov pri podanem meta-učnem algoritmu.

osnovni algoritmi					meta-učni algoritem	
logreg	rf	bayes	knn	majority	logreg	nbayes
X	X		X		0.593	0.597
X	X		X	X	0.593	0.597
X	X	X	X		0.592	0.595
X	X	X	X	X	0.592	0.595
X			X		0.595	0.594
X	X				0.591	0.594
X	X	X			0.592	0.593
		X	X		0.574	0.581
		X	X		0.575	0.573
	X	X			0.552	0.553

Tabela 4.3: Nabor parametrov, s katerimi smo izdelali 108 avtomatskih trgovalnih sistemov in jih ocenili z metodo preverjanja z vzorčenjem.

parameter	izbrane vrednosti
valutni par	EURUSD, USDJPY, GBPUSD, AUDNZD
perioda grafa	1min, 15min, 1h
odmik napovedi	4, 10, 20 svečk
algoritem	logistična regresija naključna drevesa naučeno kombiniranje z meta-učenjem

parametra, saj na primer drugi in tretji najboljši napovedni sistem uporabljata iste parametre, le odmik imata drugačen (20 in 4). Njuna razlika v točnosti je le 1.1, kar je presenetljivo, glede na to, da prvi napoveduje 4 ure v prihodnost, drugi pa le 1 uro. Izbrali smo nekaj najboljših sistemov in jih vključili v eksperiment v realnem času, ki je opisan v nadaljevanju.

4.2 Ocenjevanje uspešnosti sistema “v živo”

Izbrali smo nekaj avtomatskih trgovalnih sistemov, ki so bili po ocenah preverjanja z vzorčenjem najbolj kvalitetni, ter jih preizkusili v trgovanju na realnem trgu v realnem času. Trgovanje je potekalo na trgovalnih računih, ki so namenjeni preizkušanju trgovalne platforme in omogočajo trgovanje z namišljenimi sredstvi. Želeli smo zagotoviti pošteno primerjavo med sistemi, to pa smo lahko dosegli samo tako, da vsi delujejo na istih podatkih, torej s hkratnim trgovanjem. Na enem računalniku lahko teče le ena instanca trgovalne platforme. Poleg tega so lahko na trgovalnem računu naenkrat odprti samo posli iste smeri, kar je vzrok, da lahko na enem računalniku za podan valutni par teče samo en avtomatski trgovalni sistem hkrati. Na voljo smo imeli štiri virtualne računalnike, zato smo izbrali štiri sisteme, ki so trgovali z valutnim parom GBPUSD, štiri sisteme, ki so trgovali z valutnim parom

Tabela 4.4: Eksperimentalni rezultati preverjanja različnih sistemov. Prikazanih je samo najboljših 30 sistemov, ostali so navedeni v prilogi C.

valutni par	perioda	algoritem	odmik	AUC
GBPUSD	15min	naučeno komb.	20	0.644
USDJPY	15min	naučeno komb.	20	0.636
USDJPY	15min	naučeno komb.	4	0.625
USDJPY	15min	naučeno komb.	10	0.624
USDJPY	1min	naučeno komb.	4	0.619
USDJPY	15min	naklj. drevesa	10	0.616
GBPUSD	15min	naučeno komb.	10	0.615
EURUSD	15min	logist. regresija	4	0.606
USDJPY	15min	naklj. drevesa	20	0.603
GBPUSD	15min	naklj. drevesa	20	0.602
GBPUSD	15min	logist. regresija	4	0.601
USDJPY	15min	logist. regresija	4	0.598
GBPUSD	15min	naučeno komb.	4	0.593
GBPUSD	60min	naučeno komb.	4	0.592
GBPUSD	1min	naučeno komb.	10	0.591
GBPUSD	15min	logist. regresija	10	0.590
USDJPY	60min	logist. regresija	4	0.590
GBPUSD	60min	logist. regresija	4	0.587
EURUSD	15min	naučeno komb.	4	0.585
EURUSD	1min	naučeno komb.	10	0.581
EURUSD	1min	naučeno komb.	4	0.581
USDJPY	1min	naučeno komb.	10	0.580
GBPUSD	1min	naklj. drevesa	10	0.580
EURUSD	1min	naučeno komb.	20	0.580
USDJPY	1min	logist. regresija	4	0.579
AUDNZD	1min	naučeno komb.	4	0.579
AUDNZD	15min	logist. regresija	4	0.577
AUDNZD	60min	logist. regresija	4	0.576
USDJPY	60min	logist. regresija	20	0.574
EURUSD	1min	logist. regresija	20	0.572

USDJPY, ter en sistem, ki je trgoval z valutnim parom EURUSD. Izbrani trgovalni sistemi s pripadajočimi parametri so podani v tabeli 4.5. Samo en trgovalni sistem je trgoval z 1-minutno ločljivostjo grafa (tj. parameter perioda), ostali pa na 15-minutni ločljivosti grafa. Trgovanje je trajalo 14 delovnih dni. Potrebno je upoštevati, da so imeli trgovalni sistemi različne odmike, zato so lahko izvedli različno število poslov. Trgovalni sistem z odkikom 20 in periodo 15min je tako v 14 dneh lahko opravil le okoli 60 poslov, medtem ko je podoben sistem z odkikom 4 lahko opravil do 300 poslov. Število opravljenih poslov se med podobnimi sistemi rahlo razlikuje, saj se med trgovanjem lahko zgodi, da posrednik zaradi velikega prometa ne obdela našega naročila ali ga zavrne. Zaradi različnih odkikov smo zaradi medsebojne primerjave v tabelo rezultatov vključili tudi povprečen dobiček posla.

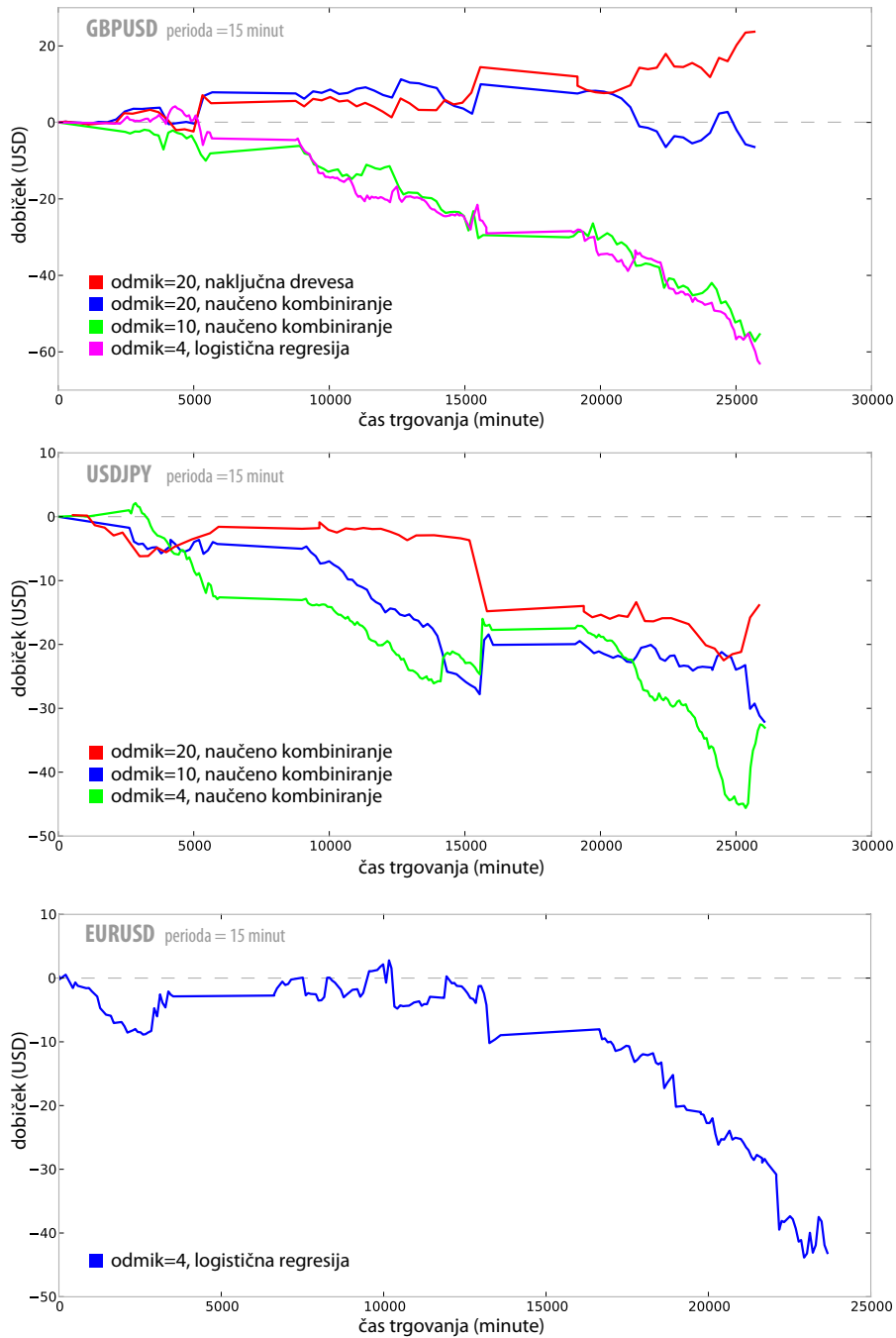
Tabela 4.5: Izbranih devet trgovalnih sistemov, ki smo jih preizkusili v avtomatskem trgovanju v živo. Poleg parametrov sistema je prikazana točnost na testnih podatkih (AUC), dobiček in povprečni dobiček (označen kot $\overline{\text{dobiček}}$).

valutni par	perioda	algoritem	odmik	AUC	dobiček	$\overline{\text{dobiček}}$	poslov
GBPUSD	15min	naklj. drevesa	20	0.602	23.72	0.39	60
GBPUSD	15min	naučeno komb.	20	0.644	-6.42	-0.11	59
USDJPY	15min	naučeno komb.	4	0.625	-33.03	-0.15	213
USDJPY	15min	naučeno komb.	20	0.636	-13.86	-0.25	55
EURUSD	15min	logist. regresija	4	0.606	-43.13	-0.25	171
USDJPY	1min	naučeno komb	4	0.619	-200.40	-0.28	705
USDJPY	15min	naučeno komb.	10	0.624	-32.10	-0.31	103
GBPUSD	15min	logist. regresija	4	0.601	-63.03	-0.34	185
GBPUSD	15min	naučeno komb.	10	0.615	-55.40	-0.56	99

Uspešnost izbranih avtomatskih trgovalnih sistemov je prikazana v tabeli 4.5 in je razvrščena glede na povprečni dobiček posla. Za lažjo primer-

javo dobička smo dobiček trgovalnih sistemov izrisali v času. Grafični prikaz uspešnosti testiranih trgovalnih sistemov je podan na sliki 4.1, z izjemo trgovalnega sistema na valutnem paru USDJPY s parametrom $perioda = 1$, ki je zaradi svoje kratke periode opravil 705 poslov, med njimi le 24 dobičkonosnih in zaradi velike izgube ni prikazan na grafu.

Opazimo, da je izmed devetih testiranih samo en avtomatski trgovalni sistem dobičkonosen. Med valutnimi pari je glede na eksperimentalne rezultate smiselno trgovati na valutnem paru GBPUSD, kjer je poleg dobičkonosnega prisoten še en sistem, katerega stanje računa se skozi trgovanje giblje blizu začetnega stanja računa 0 USD. Glede na uspešnost (oz. manjšo izgubo) tistih trgovalnih sistemov, ki napovedujejo dlje v prihodnost, lahko rečemo, da avtomatsko trgovanje deluje najbolje pri napovedovanju za 5 ur v prihodnost ($20 \times 15min$). Pri odločanju o odmiku in periodi moramo upoštevati tudi razliko med nakupno in prodajno ceno. Na primer, ko kupimo nek valutni par (odpremo posel), imamo takoj po nakupu izgubo v višini razlike med nakupno in prodajno ceno. Če želimo isti valutni par prodati po isti ceni, kot smo ga kupili, mora vrednost tečaja najprej pridobiti na ceni. V primeru prekratkih odmikov in premajhne periode je nek posel odprt premalo časa, da bi presegel to razliko, kar se je zgodilo pri sistemu s parametrom $perioda = 1$.



Slika 4.1: Grafični prikaz uspešnosti izbranih avtomatskih trgovalnih sistemov. Vsi prikazani sistemi so trgovali s parametrom $perioda = 15min$.

Poglavje 5

Sklepne ugotovitve

Razvili smo avtomatski trgovalni sistem, ki je “pametnejši” od običajnih trgovalnih sistemov. Običajni trgovalni sistemi se zanašajo zgolj na vrednosti tehničnih indikatorjev, naš trgovalni sistem pa vrednosti tehničnih indikatorjev uporabi kot učno množico pri metodah strojnega učenja. Z metodo preverjanja z vzorčenjem smo ocenili kvaliteto naključnih gozdov z različnim parametrom k in nabor učnih algoritmov pri naučenem kombiniranju z meta-učenjem, ter tako določili optimalne nastavitve učnih algoritmov. Na učnih podatkih smo z metodo preverjanja z vzorčenjem ocenili točnost napovedovanja 108-ih avtomatskih trgovalnih sistemov z različnimi nastavitvami, nato pa devet izmed boljših preizkusili še v sočasnem trgovanju v realnem času.

Z rezultati metode preverjanja z vzorčenjem smo bili precej zadovoljni. Vrednotenje “v živo” je pokazalo, da je za zanesljivo trgovanje potrebno izboljšati točnost napovedovanja, saj nam je uspelo izdelati le en dobičkonosen avtomatski trgovalni sistem.

Prvi problem pri uporabi sistema je kombinatorična eksplozija možnih parametrov sistema. Pri izboru parametrov trgovalnih sistemov v poglavju 4.1 smo se omejili zgolj na tri različne vrednosti posameznega parametra (oz. na štiri valutne pare), nepreverjenih kombinacij parametrov trgovalnega sistema in valutnih parov pa je še mnogo in ostajajo neraziskane.

Ugotovili smo, da pri napovedovanju pomembno vlogo igra časovni od-

mik, za katerega napovedujemo v prihodnost. V primeru, ko napovedujemo premalo vnaprej, oz. ko je posel odprt premalo časa, ne uspemo nadoknaditi razlike med nakupno in prodajno ceno, ki nam jo določi posrednik. Temu bi se lahko izognili tako, da bi razredu določili še tretjo vrednost, ki bi predstavljala akcijo “ni vredno odpirati novega posla, ker ne bomo uspeli nadoknaditi razlike med nakupno in prodajno vrednostjo valutnega para”. Z uvedbo tretje vrednosti bi se avtomatski trgovalni sistem lahko izognil poslom, kjer vrednost valutnega para sicer naraste, a ne dovolj, da preseže razliko med nakupno in prodajno vrednostjo valutnega para.

Naslednja možna izboljšava bi bila razširitev učne množice z dodatnimi atributi. Vrednosti tehničnih indikatorjev so verjetno preveč povezane in podobne, saj vsi temeljijo na vrednostih istega valutnega para. Prva možnost razširjanja nabora atributov bi bila vključevanje atributov drugih valutnih parov, saj so valutni pari med seboj zagotovo do neke mere odvisni. Verjetno lahko spremembo vrednosti valutnega para povzroči sprememba vrednosti nekega tretjega valutnega para. K atributom valutnega para EURUSD bi lahko vključili še attribute valutnih parov EURAUD, EURCHF, USDCHF ali drugih in morda tako dobili bolj široko informacijo o dogajanju na največjem finančnem trgu. Še ena možnost razširjanja nabora atributov bi lahko bila analiza finančnih novic. Virov novic o finančnem dogajanju po svetu je veliko, obstajajo pa celo spletni koledarji finančnih in političnih dogodkov z ocenami vpliva na valutni trg. Ustrezna obdelava in analiza naštetih virov informacij bi prav gotovo izboljšala točnost napovedovanja gibanja valutnih tečajev.

Literatura

- [1] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.
- [2] Tomaž Curk, Janez Demšar, Qikai Xu, Gregor Leban, Uroš Petrovič, Ivan Bratko, Gad Shaulsky, and Blaž Zupan. Microarray data mining with visual programming. *Bioinformatics*, 21:396–398, 2005.
- [3] Sašo Džeroski and Bernard Ženko. Is combining classifiers better than selecting the best one. In *Proceedings of the Nineteenth International Conference on Machine Learning, ICML '02*, pages 123–130, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [4] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- [5] Igor Kononenko. *Strojno učenje*. Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2005.
- [6] Monetary and Economic Department of Bank for International Settlements. Triennial central bank survey of foreign exchange and derivatives market activity in 2010.

Dodatek A

Python skripta za napovedovanje gibanja vrednosti valutnih parov

Skripta, ki smo jo razvili za strojno učenje oz. izdelavo napovednih modelov in napovedovanje gibanja vrednosti valutnih parov. Napisana je v programskem jeziku *Python*.

```
1 import Orange, time, cPickle
2
3 class pred():
4     def __init__(self, odmik=10, valuta="EURUSDm1", useDumpLearner
5         =True):
6         filePath="C:/InterbankFX/experts/files/"
7         prvic=True
8         running=True
9         self.odmik=odmik
10        self.valuta=valuta
11        print "\nParametri\nODMIK:", str(odmik), "\nVALUTA:", valuta
12        eaLog = open(filePath+"EA ["+str(self.odmik)+"] "+self.
13            valuta+"D.log", "w")
14        lastPredictionTime=time.time();
15        eaLog.write(str(time.time()).split(".")[0])
16        eaLog.close()
```

```
15     tableTime=0
16     data=[]
17     while (running):
18         time.sleep(1)
19         """
20         Check in '[+odmik] valuta.log' if new table exists.
21         """
22         tableLog = open(filePath+"["+str(self.odmik)+"] "+self.
23             valuta+"D.log", "r")
24         tmpTime=tableLog.readline()
25         tableLog.close()
26         if (useDumpLearner):
27             if (prvic):
28                 prvic=False
29                 lr=cPickle.load(open(valuta+'['+str(odmik)+'] '+'.pkl',
30                     'rb'))
31                 print "\tlearner nalozen"
32                 tableTime=tmpTime
33             else:
34                 if (tmpTime>tableTime):
35                     tableTime=tmpTime
36                     print "\tnnalagam novo tabelo"
37                     data = Orange.data.Table(filePath+"["+str(self.odmik)
38                         +"] "+self.valuta+"D.tab")
39                     print "\tnalozil, dolzina:", len(data)
40                     """
41                     Initialize learner.
42                     """
43                     logreg = Orange.classification.logreg.LogRegLearner(
44                         removeSingular=1)
45                     forest = Orange.ensemble.forest.RandomForestLearner(
46                         trees=50, attributes=8)
47                     knnLearner = Orange.classification.knn.kNNLearner()
48                     knnLearner.k = 500
49                     learners=[ logreg , forest , knnLearner ]
50                     stack = Orange.ensemble.stacking.
51                         StackedClassificationLearner(learners)
52                     lr = stack(data)
```

```

47     print "\tkoncal ucenje"
48     # Pickle the learner.
49     output = open(valuta+'['+str(odmik)+']'+'.pkl', 'wb')
50     cPickle.dump(lr, output, -1)
51     output.close()
52     print "pickle dumped"
53
54     """
55     Check in 'EA.log' for making new prediction.
56     """
57     eaLog = open(filePath+"EA ["+str(self.odmik)+"] "+self.
58         valuta+"D.log", "r")
59     tmpTime=eaLog.readline()
60     eaLog.close()
61     if (tmpTime>lastPredictionTime):
62         lastPredictionTime=tmpTime
63         try:
64             dataForPred = Orange.data.Table(filePath+"
65                 DataForPrediction["+str(self.odmik)+"] "+self.
66                 valuta+"D.tab")
67             prediction=lr(dataForPred[0])
68             print prediction, "\tveljavno do:", int(tmpTime)+10
69             predFile = open(filePath+" PredictionFile["+str(self.
70                 odmik)+"] "+self.valuta+"D.txt", "w")
71             predFile.write( str(prediction) + "\n" + str(int(
72                 tmpTime)+10) )
73             predFile.close()
74         except TypeError:
75             prediction = lr.meta_classifier(dataForPred[0])
76             print prediction, "\t STACK veljavno do:", int(tmpTime
77                 )+10
78             predFile = open(filePath+" PredictionFile["+str(self.
79                 odmik)+"] "+self.valuta+"D.txt", "w")
80             predFile.write( str(prediction) + "\n" + str(int(
81                 tmpTime)+10) )
82             predFile.close()
83         except:

```

```
76         print "EXCEPTION in DataForPrediction["+str(self.odmik  
            +")+]" "+self.valuta+"D.tab"  
            predictorExample.py
```

Dodatek B

Seznam uporabljenih atributov

Tabela B.1 vsebuje seznam uporabljenih atributov. Atribut *time* predstavlja čas, atributi *open*, *high*, *low*, *close* predstavljajo vrednosti zaključene svečke, atribut *price* predstavlja spremembo valutnega tečaja med časom t in $t - 1$ izraženo v deležu, atribut *volume* predstavlja količino trgovanih sredstev na valutnem trgu v času podane svečke. Ostali atributi so označeni s kraticami, njihov izračun in pomen je opisan v razdelku 2.2.2. Atributi so bili ustvarjeni z različnimi postopki (opisani v 3.1). Notacija, ki jo uporabljamo v tabeli je naslednja:

avg(n) in **avg(n, a)** pomeni povprečenje vrednosti indikatorja v časovnem intervalu $[t + n, t]$. V primeru več-vrednostnega indikatorja podamo še izbrano vrednost a .

avgDifference(n, a, b) pomeni povprečno razliko med vrednostmi indikatorja a in b v časovnem intervalu $[t + n, t]$.

timeFromCross(a, b) predstavlja čas v sekundah, ki je minil od pogoja $a == b$, kjer sta a in b različni vrednosti indikatorja.

timeSameSign() predstavlja čas v sekundah, ki je minil odkar ima vrednost izbranega indikatorja trenutni predznak.

timeOver(n) in timeUnder(n) predstavlja čas v sekundah, ki je minil, odkar je vrednost indikatorja višja (oz. nižja) od vrednosti n .

isOver(a, b) pomeni binarno vrednost 0, ko je $a > b$ in vrednost 1, ko je $a \leq b$.

change pomeni, da smo pri podanem indikatorju izračunali spremembo med vrednostjo v času t in $t - 1$ izraženo v deležu.

timeLast(f) predstavlja čas v sekundah, ki je minil od zadnje neničelne vrednosti indikatorja *Fractal*.

timeLastGT(a) in timeLastLT(a) predstavlja čas v sekundah, ki je minil odkar je bila vrednost indikatorja večja (oz. manjša) od vrednosti a .

Tabela B.1: Seznam uporabljenih atributov.

1	time1	99	FORCE avg(10)
2	open	100	FORCE timeSameSign()
3	high	101	FRACTALS (up)
4	low	102	FRACTALS (down)
5	close	103	FRACTAL (change)
6	price change	104	FRACTAL avg(3)
7	price avg(3)	105	FRACTAL avg(5)
8	price avg(10)	106	FRACTAL avg(10)
9	volume	107	FRACTAL timeLast(Fractal)
10	volume avg(3)	108	GATOR Upper
11	volume avg(10)	109	GATOR Lower
12	AC	110	GATOR change Upper
13	AC avg(3)	111	GATOR change Lower
14	AC avg(10)	112	GATOR avg(3, Upper)
15	AD	113	GATOR avg(5, Upper)
16	AD avg(3)	114	GATOR avg(10, Upper)
17	AD avg(10)	115	GATOR avg(3, Lower)
18	ADX high main	116	GATOR avg(5, Lower)
19	ADX median main	117	GATOR avg(10, Lower)
20	ADX (main-plusdi)	118	ICHIMOKU timeFromCross(Kij, Tenk)
21	ADX (minusdi-plusdi)	119	ICHIMOKU timeFromCross(Chin, Tenk)
22	ADX timeFromCross(minusdi, plusdi)	120	ICHIMOKU timeFromCross(Kij, Chin)
23	ADX timeFromCross(minusdi, main)	121	ICHIMOKU difference(A, B)
24	ADX timeFromCross(main, plusdi)	122	ICHIMOKU avgDifference(3, A, B)
25	ALLIGATOR jaw	123	ICHIMOKU avgDifference(5, A, B)

26	ALLIGATOR lips	124	ICHIMOKU avgDifference(10, A, B)
27	ALLIGATOR teeth	125	ICHIMOKU difference(Chin, Close)
28	ALLIGATOR (jaw-lips)	126	ICHIMOKU difference(Kij, Close)
29	ALLIGATOR (jaw-teeth)	127	ICHIMOKU difference(Tenk, Close)
30	ALLIGATOR (lips-teeth)	128	ICHIMOKU difference(SenkA, Close)
31	ALLIGATOR timeFromCross(jaw,teeth)	129	ICHIMOKU difference(SenkB, Close)
32	ALLIGATOR timeFromCross(jaw, lips)	130	ICHIMOKU avgDifference(3,Chin, Close)
33	ALLIGATOR timeFromCross(lips ,teeth)	131	ICHIMOKU avgDifference(5,Chin, Close)
34	ALLIGATOR avg(3, jaw)	132	ICHIMOKU avgDifference(10, Chin, Close)
35	ALLIGATOR avg(10, jaw)	133	ICHIMOKU avgDifference(3, Kij, Close)
36	ALLIGATOR avg(3, teeth)	134	ICHIMOKU avgDifference(5, Kij, Close)
37	ALLIGATOR avg(10, teeth)	135	ICHIMOKU avgDifference(10, Kij, Close)
38	ALLIGATOR avg(3, lips)	136	ICHIMOKU avgDifference(3, Tenk, Close)
39	ALLIGATOR avg(10, lips)	137	ICHIMOKU avgDifference(5, Tenk, Close)
40	ATR (3, 14)	138	ICHIMOKU avgDifference(10, Tenk, Close)
41	ATR (10, 14)	139	MACD avg(3, Signal)
42	ATR (3, 6)	140	MACD avg(3, Main)
43	ATR (10, 6)	141	MACD timeSameSign(Signal)
44	AO avg(3)	142	MACD timeSameSign(Main)
45	AO avg(5)	143	MOMENTUM avg(3)
46	AO avg(10)	144	MOMENTUM avg(5)
47	AO timeSameSign()	145	MOMENTUM avg(10)
48	BOLL	146	MFI avg(3)
49	BOLL avg(3) change	147	MFI avg(5)
50	BOLL avg(5) change	148	MFI avg(10)
51	BOLL avg(10) change	149	MFI timeLastGT(80)
52	BOLL isBarCrossing(Upper)	150	MFI timeLastGT(60)
53	BOLL isBarCrossing(Lower)	151	MFI timeLastLT(40)
54	BOLL difference(Upper, Close)	152	MFI timeLastLT(20)
55	BOLL difference(Lower, Close)	153	OsMA
56	BOLL avgDifference(3, Upper, Close)	154	OBV
57	BOLL avgDifference(10, Upper, Close)	155	OBV avg(3)
58	BOLL avgDifference(3, Lower, Close)	156	OBV avg(5)
59	BOLL avgDifference(10, Lower, Close)	157	OBV avg(10)
60	BULLS avg(3)	158	RSI
61	BULLS avg(5)	159	RSI avg(3)
62	BULLS avg(10)	160	RSI avg(5)
63	BEARS avg(3)	161	RSI avg(10)
64	BEARS avg(5)	162	RSI timeLastGT(70)
65	BEARS avg(10)	163	RSI timeLastGT(60)
66	BULLS timeSameSign()	164	RSI timeLastLT(40)
67	BEARS timeSameSign()	165	RSI timeLastLT(30)
68	BWMFI	166	RVI
69	BWMFI avg(3)	167	RVI timeLast()
70	BWMFI avg(5)	168	RVI timeSameSign(Main)
71	BWMFI avg(10)	169	RVI timeSameSign(Signal)
72	CCI (5)	170	RVI avgDifference(3,Main, Signal)

73	CCI avg(3)	171	RVI avgDifference(5,Main, Signal)
74	CCI avg(5)	172	RVI avgDifference(10,Main, Signal)
75	CCI avg(10)	173	SAR difference((High+Low/2), Sar)
76	CCI timeOver(100)	174	SAR avgDifference(3, (High+Low/2), Sar)
77	CCI timeUnder(-100)	175	SAR avgDifference(5, (High+Low/2), Sar)
78	CCI timeOver(0)	176	SAR avgDifference(10, (High+Low/2), Sar)
79	CCI timeUnder(0)	177	StdDev avg(3, MaPeriod=10)
80	DeMARKER	178	StdDev avg(5, MaPeriod=10)
81	DeMARKER avg(3)	179	StdDev avg(10, MaPeriod=10)
82	DeMARKER avg(5)	180	StdDev avg(3, MaPeriod=20)
83	DeMARKER avg(10)	181	StdDev avg(5, MaPeriod=20)
84	DeMARKER timeOver(70)	182	StdDev avg(10, MaPeriod=20)
85	DeMARKER timeOver(30)	183	STOCHASTIC timeLast()
86	ENVEL isOver(Close, Upper)	184	STOCHASTIC timeLastGT(80, Main)
87	ENVEL isOver(Close, Lower)	185	STOCHASTIC timeLastGT(80, Signal)
88	ENVEL difference(Close, Upper)	186	STOCHASTIC timeLastLT(20, Main)
89	ENVEL difference(Lower, Close)	187	STOCHASTIC timeLastLT(20, Signal)
90	ENVEL avgDifference(3,Close, Upper)	188	STOCHASTIC avgDifference(3,Main, Signal)
91	ENVEL avgDifference(5,Close, Upper)	189	STOCHASTIC avgDifference(5,Main, Signal)
92	ENVEL avgDifference(10,Close, Upper)	190	STOCHASTIC avgDifference(10,Main, Signal)
93	ENVEL avgDifference(3, Lower, Close)	191	WPR timeLastGT(-20)
94	ENVEL avgDifference(5, Lower, Close)	192	WPR timeLastLT(-80)
95	ENVEL avgDifference(10, Lower, Close)	193	WPR avg(3)
96	FORCE	194	WPR avg(5)
97	FORCE avg(3)	195	WPR avg(10)
98	FORCE avg(5)		

Dodatek C

Eksperimentalni rezultati preverjanja z vzorčenjem

valutni par	perioda	algoritem	odmik	AUC
GBPUSD	15min	naučeno komb. z meta-učenjem	20	0.644
USDJPY	15min	naučeno komb. z meta-učenjem	20	0.636
USDJPY	15min	naučeno komb. z meta-učenjem	4	0.625
USDJPY	15min	naučeno komb. z meta-učenjem	10	0.624
USDJPY	1min	naučeno komb. z meta-učenjem	4	0.619
USDJPY	15min	naključna drevesa	10	0.616
GBPUSD	15min	naučeno komb. z meta-učenjem	10	0.615
EURUSD	15min	logistična regresija	4	0.606
USDJPY	15min	naključna drevesa	20	0.603
GBPUSD	15min	naključna drevesa	20	0.602
GBPUSD	15min	logistična regresija	4	0.601
USDJPY	15min	logistična regresija	4	0.598
GBPUSD	15min	naučeno komb. z meta-učenjem	4	0.593
GBPUSD	60min	naučeno komb. z meta-učenjem	4	0.592
GBPUSD	1min	naučeno komb. z meta-učenjem	10	0.591
GBPUSD	15min	logistična regresija	10	0.590
USDJPY	60min	logistična regresija	4	0.590

GBPUSD	60min	logistična regresija	4	0.587
EURUSD	15min	naučeno komb. z meta-učenjem	4	0.585
EURUSD	1min	naučeno komb. z meta-učenjem	10	0.581
EURUSD	1min	naučeno komb. z meta-učenjem	4	0.581
USDJPY	1min	naučeno komb. z meta-učenjem	10	0.580
GBPUSD	1min	naključna drevesa	10	0.580
EURUSD	1min	naučeno komb. z meta-učenjem	20	0.580
USDJPY	1min	logistična regresija	4	0.579
AUDNZD	1min	naučeno komb. z meta-učenjem	4	0.579
AUDNZD	15min	logistična regresija	4	0.577
AUDNZD	60min	logistična regresija	4	0.576
USDJPY	60min	logistična regresija	20	0.574
EURUSD	1min	logistična regresija	20	0.572
EURUSD	15min	logistična regresija	10	0.572
EURUSD	60min	logistična regresija	20	0.571
AUDNZD	60min	naučeno komb. z meta-učenjem	4	0.571
USDJPY	60min	naučeno komb. z meta-učenjem	10	0.569
EURUSD	15min	naučeno komb. z meta-učenjem	10	0.568
AUDNZD	1min	naučeno komb. z meta-učenjem	10	0.568
GBPUSD	15min	logistična regresija	20	0.568
USDJPY	60min	naučeno komb. z meta-učenjem	20	0.566
USDJPY	60min	naučeno komb. z meta-učenjem	4	0.565
GBPUSD	1min	naučeno komb. z meta-učenjem	4	0.565
AUDNZD	1min	logistična regresija	4	0.565
EURUSD	1min	logistična regresija	4	0.563
USDJPY	1min	logistična regresija	10	0.563
AUDNZD	15min	naučeno komb. z meta-učenjem	4	0.563
GBPUSD	15min	naključna drevesa	10	0.562
USDJPY	15min	naključna drevesa	4	0.561
AUDNZD	1min	naključna drevesa	20	0.561
AUDNZD	60min	logistična regresija	10	0.561

GBPUSD	60min	naključna drevesa	20	0.561
GBPUSD	1min	logistična regresija	10	0.558
EURUSD	60min	logistična regresija	4	0.558
EURUSD	15min	naučeno komb. z meta-učenjem	20	0.557
EURUSD	15min	logistična regresija	20	0.556
EURUSD	60min	logistična regresija	10	0.556
AUDNZD	1min	naključna drevesa	10	0.556
USDJPY	60min	logistična regresija	10	0.555
GBPUSD	60min	naučeno komb. z meta-učenjem	10	0.554
GBPUSD	1min	logistična regresija	4	0.553
USDJPY	1min	logistična regresija	20	0.551
EURUSD	60min	naučeno komb. z meta-učenjem	20	0.548
USDJPY	1min	naključna drevesa	4	0.548
USDJPY	15min	logistična regresija	10	0.547
USDJPY	1min	naučeno komb. z meta-učenjem	20	0.545
GBPUSD	1min	logistična regresija	20	0.545
GBPUSD	60min	naučeno komb. z meta-učenjem	20	0.545
EURUSD	1min	naključna drevesa	10	0.541
EURUSD	1min	logistična regresija	10	0.540
AUDNZD	15min	logistična regresija	10	0.539
GBPUSD	60min	logistična regresija	10	0.538
GBPUSD	1min	naključna drevesa	20	0.537
AUDNZD	1min	naključna drevesa	4	0.537
EURUSD	1min	naključna drevesa	4	0.536
USDJPY	15min	logistična regresija	20	0.535
AUDNZD	1min	logistična regresija	10	0.535
GBPUSD	60min	naključna drevesa	10	0.535
AUDNZD	1min	naučeno komb. z meta-učenjem	20	0.535
AUDNZD	60min	naključna drevesa	20	0.534
AUDNZD	60min	naučeno komb. z meta-učenjem	20	0.528
GBPUSD	15min	naključna drevesa	4	0.528

EURUSD	1min	naključna drevesa	20	0.526
EURUSD	60min	naučeno komb. z meta-učenjem	4	0.525
AUDNZD	15min	naučeno komb. z meta-učenjem	20	0.524
USDJPY	1min	naključna drevesa	20	0.524
GBPUSD	1min	naključna drevesa	4	0.523
AUDNZD	15min	logistična regresija	20	0.522
GBPUSD	1min	naučeno komb. z meta-učenjem	20	0.521
USDJPY	60min	naključna drevesa	10	0.521
AUDNZD	60min	naučeno komb. z meta-učenjem	10	0.520
AUDNZD	15min	naučeno komb. z meta-učenjem	10	0.520
EURUSD	15min	naključna drevesa	10	0.519
AUDNZD	60min	logistična regresija	20	0.519
GBPUSD	60min	naključna drevesa	4	0.518
USDJPY	1min	naključna drevesa	10	0.516
AUDNZD	1min	logistična regresija	20	0.516
AUDNZD	15min	naključna drevesa	20	0.516
EURUSD	15min	naključna drevesa	4	0.516
EURUSD	60min	naučeno komb. z meta-učenjem	10	0.516
EURUSD	60min	naključna drevesa	20	0.512
GBPUSD	60min	logistična regresija	20	0.511
USDJPY	60min	naključna drevesa	4	0.508
USDJPY	60min	naključna drevesa	20	0.507
EURUSD	15min	naključna drevesa	20	0.506
EURUSD	60min	naključna drevesa	10	0.505
AUDNZD	60min	naključna drevesa	4	0.504
AUDNZD	15min	naključna drevesa	4	0.499
EURUSD	60min	naključna drevesa	4	0.495
AUDNZD	60min	naključna drevesa	10	0.474
AUDNZD	15min	naključna drevesa	10	0.470
