Wat is het meest geschikte AI model voor het forecasten van de koers van cryptomunten aan de hand van open source data

INTERNE PROMOTOR: WOUTER GEVAERT
EXTERNE PROMOTOR: SEBASTIEN PEREZ

ONDERZOEKSVRAAG UITGEVOERD DOOR

STUDENT JOREN VANGOETHEM

VOOR HET BEHALEN VAN DE GRAAD VAN BACHELOR IN DE

MULTIMEDIA & CREATIVE TECHNOLOGIES

HOWEST | 2021-2022

Woord vooraf

Deze bachelorproef sluit aan op mijn research project van vorig semester waarbij ik onderzocht welke neurale netwerkmodellen het best geschikt waren om te gebruiken op de crypto markt voor crypto trading.
Gedurende een drietal weken heb ik geprobeerd het beste model te vinden, en welke data hier voor nodig was, om een optimaal 'trading model' te maken dat effectief kan ingezet worden op de crypto markt. Ik heb hierbij samengewerkt met Andreas Maerten. Wij vergeleken ons resultaat met andere manieren van crypto trading, zowel manueel als algoritmisch traden. Ons neuraal netwerk had voor- en nadelen vergeleken met beide, en die zal ik verder bespreken.

In deze bachelorproef zal ik de technische details en de resultaten van ons onderzoek bespreken, en advies formuleren voor wie gelijkaardig onderzoek zou willen uitvoeren of verder bouwen op ons werk.

Ik zal het onderzoek, de technische details en de resultaten zo goed mogelijk proberen bespreken in deze bachelorproef. Alsook advies geven voor iemand dat een gelijkaardig onderzoek zou willen uitvoeren of verder bouwen op mijn onderzoek.

Ik wil ook graag Wouter Gevaert en Marie Dewitte bedanken voor hun hulp tijdens mijn onderzoek.

Joren vanGoethem – 22/04/2022

Abstract

Mijn onderzoeksvraag “Wat is het meest geschikte model voor het voorspellen van de koers van cryptomunten aan de hand van open source data?” heb ik gekozen vanwege mijn eigen interesse voor crypto. Ik wou onderzoeken of het mogelijk was met een neuraal netwerk om patronen of correlaties te zien in deze data, die toelaten een correcte voorspelling te maken.

Mijn onderzoek ging vooral in op welke types van neurale netwerken hiervoor best geschikt zijn en welke data men hiervoor nodig heeft. Het werd al snel duidelijk dat LSTM’s de enige goede optie waren voor het voorspellen van time series data. LSTM’s kunnen dankzij hun long term memory (in tegenstelling tot een GRU netwerk dat geen long term memory heeft) betere voorspellingen maken omdat ze rekening kunnen houden met wat er net gebeurd is. Men kan niet echt voorspellen welke richting een crypto munt zal uitgaan (stijgen of dalen in waarde) door enkel de laatste prijswaarden te beschouwen.
Dus het model, en ook de structuur van het model zoals de aantal layers en aantal neurons per layer, bleken heel belangrijk om een snel maar accuraat model te verkrijgen. Het grote nadeel van LSTM modellen is dat training heel lang duurt vanwege de grote hoeveelheid berekeningen vergeleken met andere typen modellen. Maar gelukkig was ons model niet extreem groot en viel dit vrij goed mee.

Een ander belangrijke element was natuurlijk de training en de daarbij horende test data. De data werd opgehaald met de publieke API van Binance, een van de grootste crypto exchanges ter wereld. De candle data hiervan was echter niet genoeg om een goed model te bekomen. Ik zal verder bespreken wat we gedaan hebben om onze data zo goed mogelijk te corrigeren, en welke de impact daarvan was op ons model. Dit leverde een bevredigend resultaat, al is er natuurlijk ruimte voor verbetering indien meer tijd aan de training en het design van het model kan besteed worden.

Inhoudsopgave

Figurenlijst

Table of Figures

Lijst met afkortingen

ADOSC	Accumulation/Distribution Oscillator
ATR	Average True Range
EMA	Exponential Moving Average
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory
MACD	Moving Average Convergence Divergence
MFI	Money Flow Index
NLP	Natural Language Processing
RNN	Recurrent Neural Network
RSI	Relative Strength Index

Verklarende woordenlijst

bearish	een dalende trend in de prijs van een asset
layer	een laag van neurons in een neuraal netwerk
LSTM	een neural netwerk type waarbij er een long-term memory door alle layers heen gaat
bullish	een stijgende trend in de prijs van een asset
trend reversal	een switch tussen down en up trend
candle	een weergave van de Low, High, Open en Close prijs van een bepaalde tijdsperiode

1 Inleiding

Traden van cryptomunten is nog relatief nieuw maar toch al vrij populair bij de jeugd, vergeleken met het verhandelen van aandelen wat vooral bij de iets oudere generaties bekend is. Het leek mij een leuk idee om in te spelen op iets wat bij de jeugd populair is, en waar ikzelf trouwens ook actief mee bezig ben.

1.1 Aanleiding en inspiratie

Waarom koos ik als onderzoeksvraag: “Wat is het meest geschikte AI model voor het voorspellen van de koers van cryptomunten aan de hand van open source data?”

Het idee van een neuraal netwerk dat kan voorspellen wanneer je best kan kopen en verkopen op de cryptomarkt of op de aandelenmarkt leek heel boeiend. Momenteel worden de meeste verhandelingen op de crypto- en aandelenmarkt algoritmisch uitgevoerd. Dit wil dus zeggen dat men niet meer manueel kijkt naar de evolutie van de prijzen en/of naar de prestaties van de bedrijven om voorspellingen te doen. Wat men wel doet is deze data in een algoritme stoppen, en dat algoritme zal dan automatisch een buy, sell of hold target teruggeven. Het is niet precies gekend hoeveel trades algoritmisch gestuurd zijn, maar afhankelijk van de bronnen die men online kan raadplegen ligt dit tussen 60% en 80%.

Graag wou ik onderzoeken of we als alternatief voor algoritmes geen neurale netwerken kunnen gebruiken. Uiteindelijk is dit ook maar een reeks berekeningen op basis van input data, die dan een buy, sell of hold target kunnen voorspellen.

Dit onderzoek lijkt ook technisch interessant om te zien hoe goed neurale netwerken time series kunnen voorspellen in een zeer onvoorspelbare omgeving, waar zelfs geroutineerde mensen vaak fouten maken omwille van emoties en irrationele redeneringen.

1.2 Deelvragen

Dit zijn de deelvragen die in dit onderzoek ook beantwoord zullen worden.

– is het mogelijk met enkel prijs en volume data een voorspelling te doen van de crypto markt?

– Welk type model heeft het beste resultaat? is het mogelijk reinforcement learning te gebruiken?

– Is het resultaat beter wanneer we trainen per munt, i.p.v. een training op alle munten samen?

1.3 Keuzes

Er zijn enkele redenen waarom we dit onderzoek niet op de aandelenmarkt maar cryptomarkt doen. Bijvoorbeeld de cryptomarkt is 24/7 actief, in tegenstelling de aandelenmarkt die enkel op weekdagen en op wel bepaalde uren open is.

De commissies op cryptomunten zijn merkelijk lager dan op de aandelenmarkten, en de vergoedingen (fees) bij aan- en verkoop liggen slechts tussen 0% en 1% naargelang de gebruikte exchange. Bij aandelen loopt die vergoeding al snel hoog op omdat men vaak een maandelijkse kost betaalt voor de markt gegevens die men opvraagt via API, een ‘maintenance fee’ voor het behouden van een account, commissie op de trades, etc.…

Het wordt al snel duidelijk waarom crypto de betere keuze is voor ons onderzoek. Een ander belangrijk voordeel is dat - zeker bij het trainen van een neuraal netwerk – het opvragen van de data

bij crypto exchanges gratis en gemakkelijk is via de API. Voor ons onderzoek hebben we data gebruikt van de Binance Exchange.

1.4 Doelen

Het voornaamste doel van ons onderzoek is een werkend model te ontwikkelen dat kan beslissen wanneer te kopen en te verkopen. Dit moet aantonen of het wel degelijk mogelijk is om neurale netwerken te gebruiken op time series in een zeer wisselvallige en onvoorspelbare omgeving. Ook belangrijk is te onderzoeken welke data er precies relevant is om tot correcte beslissingen te komen: is candle data alleen voldoende, of zullen we meer nodig hebben zoals indicators? Dit zal nog tot in detail onderzocht en getest worden.

Hopelijk kan het model uiteindelijk goed genoeg voorspellen wanneer te kopen en te verkopen, zodat het kan ingezet worden op de echte crypto markt.

Het uiteindelijke doel is natuurlijk dat het model betrouwbaar genoeg is om later effectief in te zetten op de crypto markt om alzo winstgevende trades te sturen.

2 Research

De eerste taak bestond er in info te verzamelen over de beschikbare exchanges, zijnde platformen waarop men cryptomunten kan verhandelen, alsook of deze wel een API beschikbaar hadden om de nodige data op te halen voor training. Hiervoor hebben we voor de Binance Exchange API gekozen omdat deze gratis toegankelijk is. Er bestaan reeds een aantal libraries voor allerlei programmeertalen om data op te vragen, wat dit zeker vergemakkelijkte. Wij kozen voor de Python-binance library die te vinden is op Github.

2.1 Data

Het eerste dat we nodig hebben om voorspellingen te doen is de LOHC data, ook wel candles genoemd. Een candle bevat de Low, Open, High en Close prijs van een bepaalde periode. Deze candles worden meestal groen en rood weergegeven. De kleur duidt aan welke kant de koers opgaat: een groene candle is een candle waarbij de Close hoger ligt dan de Open. Dit is omgekeerd bij de rode candle. Ook halen we per candle het volume op, dit is de hoeveelheid van deze cryptomunt die verhandeld is tijdens deze periode. Het volume varieert echter wel van exchange tot exchange, omdat het volume dat door een exchange opgegeven wordt enkel het verhandelde volume binnen de eigen exchange is.

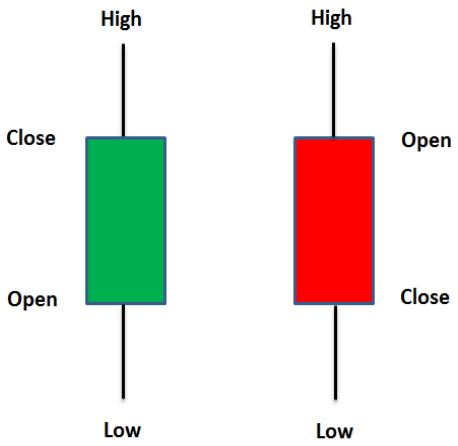


Figure 1: Candle

Voor ons onderzoek maken we gebruik van 1 minuut candles, dit wil zeggen dat er tussen de Open en de Close exact 1 minuut ligt. Dit is de hoogste resolutie die men bij exchanges kan ophalen, zodat we dus zoveel mogelijk data hebben en ook omdat ons model dan gebruik kan maken van de kleine schommelingen in prijs op minuten.

Dit laat ook toe de minuut-candles te comprimeren tot andere candles, zoals bijvoorbeeld uur-candles. Dit kan gemakkelijk gedaan worden door 60 minuut-candles te nemen. Van de eerste candle houdt men de Open bij, van de laatste candle de Close, en dan van de 60 candles behoudt men de hoogste High en de laagste Low. De volumes van de 60 minuut-candles kan men gewoon optellen, en zo bekomen we een uur-candle. Daarom zijn minuut-candles de beste optie omdat men hiervan elke andere lengte kan afleiden.

Voor het ophalen van onze data maken we gebruik van de Binance Exchange API. Deze is volledig gratis te gebruiken. Men moet wel een account aanmaken op de site en API keys onder ‘account management’. We hebben 15 GB aan minuut-candles van ongeveer 360 verschillende cryptomunten opgehaald en deze weggeschreven naar CSV bestanden. Deze worden daarna verwerkt voor normalisatie en data augmentation.

Naast de LOHC data hebben we ook targets nodig waarop we ons model kunnen trainen. Die worden bekomen door onze data door een c++ programma te runnen dat dan targets gaat toevoegen voor buy, sell en hold. Dit programma is in de bijlagen te vinden en wordt verder besproken.

Een andere optie, naast labelling, was reinforcement learning. Maar dit had een veel grotere trainingstijd gevraagd, en dit was nu al een limiterende factor tijdens het onderzoek.

De candle data van de verschillende munten moet natuurlijk genormaliseerd worden. Dit is vrij eenvoudig: we nemen voor elke candle gewoon het percentage verschil met de vorige candle. Stel dat de vorige minuut de Close prijs op 100 stond, en nu op 101, dan zal de genormaliseerde waarde 0.01 zijn.

Aan de aldus bekomen data worden ook nog indicator toegevoegd. Deze worden later besproken

Om het lezen, schrijven en volume van de data wat in te perken slaan we niet meer op naar CSV maar naar binary files. Dit bespaart ons een behoorlijk volume aan opslag en maakt het lezen en schrijven sneller. Het nadeel is echter dat deze files niet meer te lezen zijn voor mensen, maar dit is snel opgelost door de data in te laden met een scriptje en weer te geven in terminal voor controle van de data.

2.2 Model

Op basis van de nodige data, en de structuur hiervan, kunnen we een neuraal netwerk ontwerpen waarmee we onze voorspellingen willen berekenen. Hiervoor werd eerst wat onderzoek gedaan naar wat anderen al geprobeerd hebben om voorspellingen te doen voor stock- of cryptotrading op basis van time series data met verschillende soorten neurale netwerken. [1][2]

2.2.1 LSTM Netwerken

De naam LSTM staat voor Long Short Term Memory. LSTM netwerken hebben een Short Term memory maar ook een Long Term memory omdat er een flow van data door alle opeenvolgende neurons gaat, waardoor oudere data ook een invloed heeft op de volgende neurons, en de uiteindelijke output waarde, de impact van een neuron op de cell state is niet altijd even groot. Het model kan zelf bepalen of een datapunt al dan niet meer of minder relevant is dan andere. Het voordeel van LSTM's bij time series data is dat er soms nuttige informatie in zowel iets oudere data en de nieuwste data zit, en dat je beide nodig hebt voor een correcte voorspelling. Afbeeldingen in dit hoofdstuk komen uit het artikel van referentie [5].

2.2.2 LSTM Gates

LSTM's gebruiken een serie van 'gates' die bepalen hoe de data verwerkt wordt: de Forget Gate, de Input Gate en de Output Gate.

Hieronder de legende voor de volgende afbeeldingen over de verschillende gates.

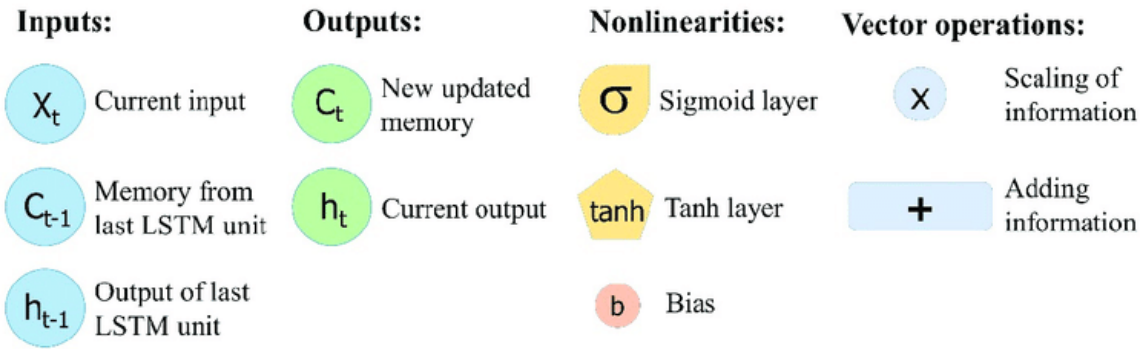


Figure 2: LSTM neuron legend

2.2.3 Forget Gate

Rechts ziet men een weergave van de Forget Gate.

Deze gate zal bepalen of de nieuwe input data relevant is, steunend op de Cell State en de Hidden State, door middel van een Sigmoid activation. De Cell State wordt doorgegeven van de vorige neuron en is dus het Long Term memory.

De Hidden State is de output van de vorige neuron. De input data is de nieuwe data die enkel toegevoegd wordt als deze relevant genoeg is. Kortom de Forget Gate bepaalt welke delen van het geheugen vergeten mogen worden.

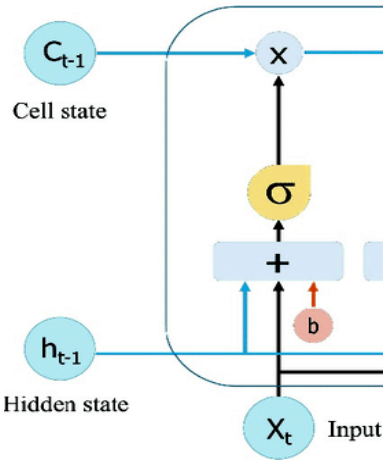


Figure 3: LSTM Forget Gate

2.2.4 Input Gate

De volgende gate word dan de Input Gate, deze zal bepalen welke gegevens van de input data toegevoegd zullen worden aan de cell state. De input in deze gate is dezelfde als de input in de Forget Gate, maar hier wordt er bepaald wat toegevoegd wordt en niet wat vergeten wordt.

Hier worden 2 verschillende activatiefuncties gebruikt.

De Tanh functie zal de vorige Hidden State combineren met de nieuwe input data om een memory update vector te maken. Deze vector bevat de informatie van de input data, en hoeveel de cell state moet geüpdatet worden met deze data. Tanh gebruikt omdat de waardes hier tussen -1 en 1 liggen omdat men mogelijks ook de impact van nieuwe data wil verminderen.

De Sigmoid activatiefunctie zal bepalen welke onderdelen van de nieuwe input data effectief relevant genoeg zijn om te onthouden. Het is dus mogelijk dat de Tanh functie bepaalde onderdelen een hoge waarde geeft, maar dat er hier toch bepaald wordt dat deze minder impact moeten hebben. Hier wordt Sigmoid gebruikt, met waarden tussen 0 en 1, waarbij 0 wil zeggen dat de data niet moet geüpdatet worden.

Deze vectoren worden pointwise (punt per punt) vermenigvuldigd met elkaar, en deze output vector wordt dan toegevoegd aan de cell state.

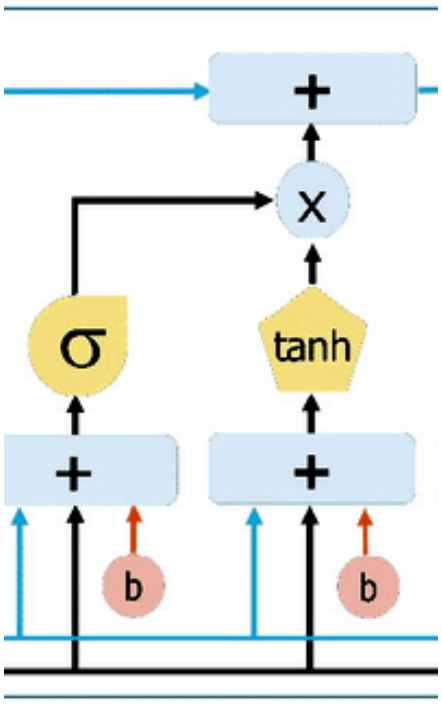


Figure 4: LSTM input gate

2.2.5 Output Gate

De laatste gate is de Output Gate. Deze zal de nieuwe Hidden State bepalen aan de hand van de nieuwe cell state, de vorige Hidden State en de nieuwe input data.

De nieuwe cell state wordt gecombineerd met de input data en de vorige Hidden State. De cell state wordt eerst nog door een Tanh functie gestuurd om de waarden binnen het bereik -1 en 1 te forceren.

De Hidden State en input data worden door en Sigmoid functie gestuurd.

Deze twee vectoren worden dan weer vermenigvuldigd met elkaar en dit vormt dan de nieuwe Hidden State.

In de afbeelding rechts ziet u ook de output, maar deze wordt enkel helemaal op het einde gegeven en niet tijdens het doorsturen van data.

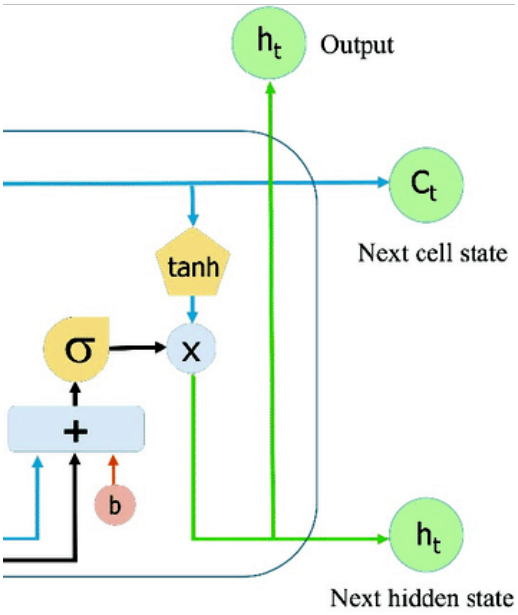


Figure 5: LSTM output gate

2.2.6 Eerste Tests

Een groot nadeel bij LSTM modellen is echter dat training heel lang duurt en krachtige hardware nodig heeft en dit werd ook voor ons al snel duidelijk.

We zijn begonnen met een simpel LSTM netwerk zoals hieronder te zien met 3 LSTM layers, een dense layer en een final output layer met 3 output neurons voor onze buy, sell en hold targets. De frame size, het aantal datapunten die je meegeeft om een prediction uit te voeren, was 240 candles bij al onze modellen, dit leek ons meer dan voldoende omdat we zelf ook niet verder dan dat zouden terug kijken om een trade uit te voeren.

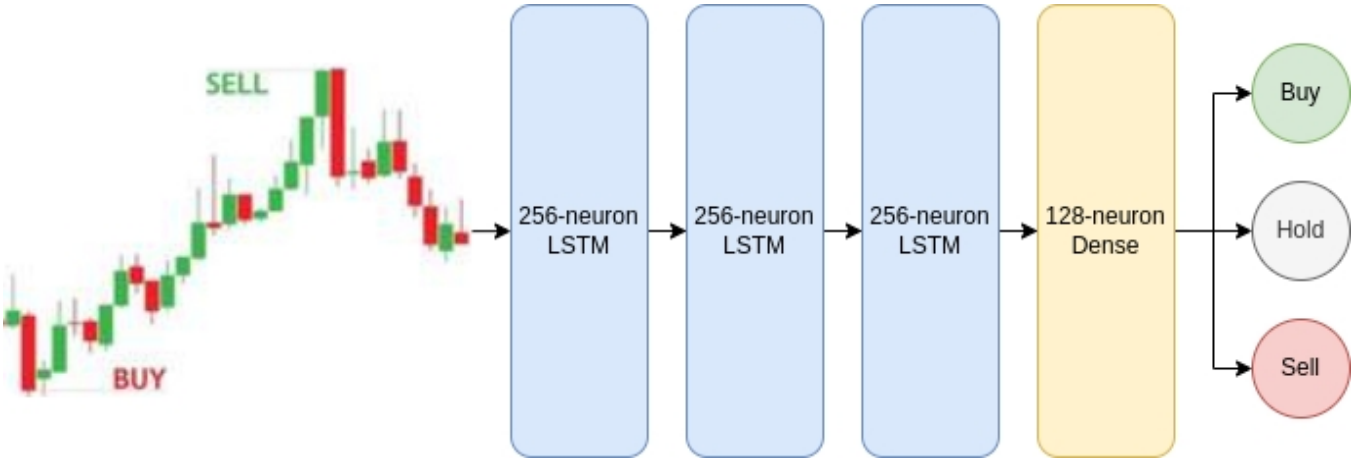


Figure 6: initial model layout

Het was echter snel duidelijk dat het model dat enkel op prijs en volume is gesteund geen goede voorspellingen kon maken. Maar dit hadden we eigenlijk wel verwacht. Daarom dat ons c++ programma ontworpen werd voor indicators berekeningen, maar deze werden aanvankelijk niet toegepast. Het is in theorie mogelijk dat het model zelf deze correlaties en berekeningen maakt tussen de prijs en volume data van de vorige X aantal candles maar het model zou mogelijks groter moeten zijn en training zou langer duren. De indicators worden berekend met enkel prijs en volume data. Er wordt dus geen nieuwe data toegevoegd, maar de berekeningen op die prijs en volume data laten toe om bepaalde trends te voorspellen. Dit is een belangrijke hulp voor het neuraal model omdat het die informatie direct kan gebruiken en dus zelf minder deze berekeningen moet maken.

Hieronder worden de berekeningen en het nut van de specifieke indicators die we gekozen hebben verduidelijkt.

2.3 Indicators

Indicators zijn een onderdeel van technische analyse, iets wat investeerders vaak nog manueel doen om de volatiliteit, richting en sterkte van een trend van een stock of coin te bepalen. Het doel van een goede technische analyse is om te voorspellen wat er in de nabije toekomst zal gebeuren, er zijn zeer veel verschillende indicators met verschillende resultaten en doelen. We zullen enkel dieper ingaan op de 6 volgende indicators omdat deze zeer gekend zijn en vaak gebruikt worden en deze ook in ons onderzoek gebruikt zullen worden.

- Accumulation / Distribution Oscillator
- Average True Range
- Bollinger Bands®
- Moving Average Convergence Divergence
- Money Flow Index
- Relative Strength Index

De reden voor het combineren van meerdere indicators is dat zij elk een ander aspect van de serie gebruiken of voorspellen [6]. Zo kan men indicators rangschikken in verschillende types waarvan de meest gebruikte de volgende zijn:

- momentum
- trend
- oscillator
- volatiliteit

de combinatie van deze verschillende soorten zorgt er voor dat we voldoende variatie hebben en alzo genoeg informatie voor het neuraal netwerk om te leren wanneer het beter een buy, sell of hold order moet voorspellen.

2.2.1 Accumulation / Distribution Oscillator

De A/D oscillator, ook wel gekend als de chaikin oscillator, is een momentum indicator van de Acummulation/Distribution lijn, en niet zozeer de prijs van de coin zelf. De A/D lijn is een cummulatieve indicator die door middel van volume en prijs het aanbod en de vraag probeert te bepalen en hiermee de sterkte van een trend, of deze nu up of down is. Het kan echter ook dat de indicator het omgekeerde voorspelt van de trend op dit moment. Bijvoorbeeld: Als de prijs aan het stijgen is maar de indicator daalt, dan is de kans groot dat er een trend reversal aankomt.

Hieronder bevindt zich de formule voor het berekenen van de A/D oscillator.[15]

$$\mathrm{N} = \frac{(\mathrm{Close} - \mathrm{Low}) - (\mathrm{High} - \mathrm{Close})}{\mathrm{High} - \mathrm{Low}}$$

$$\mathrm{M} = \mathrm{N} * \mathrm{Volume\,(Period)}$$

$$\mathrm{ADL} = \mathrm{M\,(Period - 1)} + \mathrm{M\,(Period)}$$

$$\mathrm{CO} = (\text{3-day EMA of ADL}) - (\text{10-day EMA of ADL})$$

where:

$\mathrm{N} = \text{Money flow multiplier}$

$\mathrm{M} = \text{Money flow volume}$

$\mathrm{ADL} = \text{Accumulation distribution line}$

$\mathrm{CO} = \text{Chaikin oscillator}$

Figure 7: A/D oscillator formula

hieronder ziet men een voorbeeld van de indicator op een grafiek. Bovenaan de candles, in het midden het volume en onderaan de indicator. Het is duidelijk dat het volume een grote rol speelt bij deze indicator.

Figure 8: A/D oscillator example

2.2.2 Average True Range

Deze indicator is een volatiliteit indicator die de volatiliteit van een coin probeert te bepalen. Deze zegt niet echt iets over de richting of sterkte van trend maar kan wel samen met andere indicators een duidelijker beeld geven over de sterkte van een trend. De ATR is een subjectieve indicator en is vrij te interpreteren, er is geen vaste regel voor welke waardes een trend reversal voorspellen.

Hieronder bevindt zich de formule voor het berekenen van de ATR. [17]

`Cp` staat voor Previous Close. L en H staan voor Low en High.

$$TR = \text{Max}[(H - L), \text{Abs}(H - C_P), \text{Abs}(L - C_P)]$$

$$ATR = \left(\frac{1}{n}\right) \sum_{(i=1)}^{(n)} TR_i$$

where:

$TR_i = \text{A particular true range}$

$n = \text{The time period employed}$

Figure 9: ATR formula

hier ziet u een voorbeeld van de indicator op een grafiek. Het volume is niet van belang bij deze indicator. Ook kan u zien dat de ATR niet de trend volgt maar stijgt bij grote veranderingen in prijs, dit is omdat het de volatiliteit aanduid en niet de trend.

Figure 10: ATR example

2.2.3 Bollinger Bands

Bollinger bands is ook een volatiliteit indicator maar wordt weergegeven over de candle grafiek en bevat 3 effectieve outputs, een lower, middle en upper band om een duidelijke volatiliteits 'range' aan te duiden. Deze indicator wordt vooral gebruikt om te zien of een coin oversold of overbought is. Als de waarde van de coin dicht bij of over de lower band gaat dan is deze oversold en vice versa voor de upper band. Deze formule maakt gebruik van een standaard afwijking en deze kan zelf gekozen worden maar meestal wordt 2 gebruikt. Een breakout buiten de bands is meestal een duidelijk teken van hoge volatiliteit en wordt meestal als een duidelijk signaal gezien om te kopen of verkopen.

Hieronden bevindt zich de formule om de bollinger bands te berekenen. De uiteindelijke middle band is het average van de upper en lower band en staat niet vermeld in deze formule.[16]

$$\mathrm{BOLU} = \mathrm{MA}(\mathrm{TP}, n) + m * \sigma[\mathrm{TP}, n]$$

$$\mathrm{BOLD} = \mathrm{MA}(\mathrm{TP}, n) - m * \sigma[\mathrm{TP}, n]$$

where:

$\mathrm{BOLU} = \text{Upper Bollinger Band}$

$\mathrm{BOLD} = \text{Lower Bollinger Band}$

$\mathrm{MA} = \text{Moving average}$

$\mathrm{TP\ (typical\ price)} = (\mathrm{High} + \mathrm{Low} + \mathrm{Close}) \div 3$

$n = \text{Number of days in smoothing period (typically 20)}$

$m = \text{Number of standard deviations (typically 2)}$

$\sigma[\mathrm{TP}, n] = \text{Standard Deviation over last } n \text{ periods of TP}$

Figure 11: Bollinger Bands Formula

Hieronder ziet u een voorbeeld van de indicator op een grafiek. Het is duidelijk te zien dat meestal als de candles de upper of lower band aanraken er een trend reversal is. De grootte of duur van de trend reversal is echter niet te bepalen met enkel bollinger bands dus deze zijn soms maar heel klein en van korte duur.

Figure 12: Bollinger Bands Example

2.2.4 Moving Average Convergence Divergence

De MACD is een trend-following momentum indicator die de relatie tussen 2 moving averages van een verschillende lengte weergeeft. De MACD wordt meestal gebruikt met exponential moving averages (EMA) maar andere types kunnen zeker ook gebruikt worden. Een EMA houdt meer rekening met de meer recente data punten minder met oude data punten.

De MACD is een lagging indicator, dit wilt zeggen dat deze eigelijk een beetje achter loopt op wat er eigelijk aan het gebeuren is maar desondanks is dit een vaak gebruikte en nuttige indicator en wordt deze toch gebruikt om trend reversals te voorspellen.

Hieronder bevindt zich de formule voor de MACD.[12]

$$\text{MACD} = \text{12-Period EMA} \; - \; \text{26-Period EMA}$$

Figure 13: MACD formula

Naast deze lijn kan je ook de MACD signal line gebruiken door een EMA te nemen van de MACD. Als je deze dan aftrekt van de effectieve MACD krijg je het MACD historgram te zien op onderstaande grafiek.

De blauwe lijn is de MACD, de oranje lijn is de Signal line en dan zie je ook het histogram in groen en rood.

Het kruisen van de blauwe en oranje lijn wordt gezien als een bullish of bearish crossover afhankelijk van of de blauwe naar boven of naar beneden door de oranje lijn gaat.

Figure 14: MACD example

2.2.5 Money Flow Index

De MFI is een oscillator die gebruik maakt van prijs en volume data om een overbought of oversold signaal weer te geven. De naam Money Flow Index is omdat deze de prijs en volume gebruikt en dit dus eigelijk een berekening is op de hoeveelheid geld die verhandeld wordt. Deze indicator wordt vooral gebruikt voor het voorspellen van een trend reversal. De MFI bevind zich altijd tussen een waarde van 0 en 100, een waarde boven 80 wordt meestal als een overbought signaal gezien en onder 20 als oversold. Als de indicator begint te stijgen tijdens het dalen van de prijs kan dit ook wijzen op een trend reversal.

Hieronder bevindt zich de formule vinden van de MFI.[14]

$$\text{Money Flow Index} = 100 - \frac{100}{1 + \text{Money Flow Ratio}}$$

where:

$$\text{Money Flow Ratio} = \frac{\text{14 Period Positive Money Flow}}{\text{14 Period Negative Money Flow}}$$

$$\text{Raw Money Flow} = \text{Typical Price} * \text{Volume}$$

$$\text{Typical Price} = \frac{\text{High} + \text{Low} + \text{Close}}{3}$$

Figure 15: MFI formula

Op onderstaande grafiek ziet u de MFI en er zijn ook horizontale lijnen getrokken op 80 en 20 om deze overbought en oversold signalen duidelijk te maken. Het is duidelijk te zien dat een groot volume een grote impact kan hebben op de MFI en deze indicator op zich niet heel duidelijk is en meestal samen met andere indicators gebruikt wordt.

Figure 16: MFI example

2.2.6 Relative Strength Index

De RSI is een momentum indicator dat ook wordt gebruikt voor overbought en oversold signalen maar deze gebruikt enkel prijs data en geen volume data. Er zijn meerdere varianten van de RSI maar er zit niet zo een groot verschil tussen de andere varianten dus wij hebben de originele RSI gekozen. De RSI is vooral nuttig in een situatie met hoge volatiliteit omdat deze anders een lange tijd hetzelfde signaal weergeeft als een coin blijft stijgen of dalen.

Ook deze indicator bevindt zich steeds tussen 0 en 100 en de signalen worden vooral gebruikt als deze boven 80 of onder 20 gaat.

Hieronder bevindt zich de formule voor de RSI.[13]

$$RSI_{\text{step one}} = 100 - \left[\frac{100}{1 + \frac{\text{Average gain}}{\text{Average loss}}}\right]$$

$$RSI_{\text{step two}} = 100 - \left[\frac{100}{1 + \frac{(\text{Previous Average Gain} \times 13) + \text{Current Gain}}{((\text{Previous Average Loss} \times 13) + \text{Current Loss})}}\right]$$

Figure 17: RSI formula

Zoals te zien op onderstaande grafiek volgt de RSI duidelijk de prijs maar de hoge en lage pieken buiten de 80 en 20 wijzen toch meestal wel op een trend reversal.

Figure 18: RSI example

2.4 Training

2.4.1 Supervised Learning

Voor het trainen van het model hebben we dus gekozen voor een supervised vorm van training. We hebben onze data op voorhand gelabeled op een manier waarvan we denken het een goed target is voor het model om naartoe te werken, echter heeft dit ook wat nadelen die hier toegelicht zullen worden.

2.4.2 Nadelen

Een van de nadelen van deze manier van werken is dat de accuracy tijdens training geen duidelijke maatstaf is voor de effectieve winst die het model zou kunnen halen, omdat het model een voorspellingen kan maken die in theorie fout zijn, maar in praktijk toch goed genoeg blijken om winst te maken.

Bijvoorbeeld, in onderstaande afbeelding ziet men groene en rode aanduidingen op de grafiek, die staan voor resp. aankoop en verkoop. Deze staan niet op de meest optimale punten, waar dus de labels zouden staan, maar het is wel duidelijk dat deze trades winstgevend zouden zijn. Toch zou het model in dit geval een lage accuracy gehaald hebben tijdens training, desondanks de goede winst. Dit is een nadeel bij onze manier van training die mogelijks verholpen kan worden door reinforcement learning te gebruiken waarbij men eerder rekening zou houden met de totale winst over een periode als reward, en niet met vaste labels.

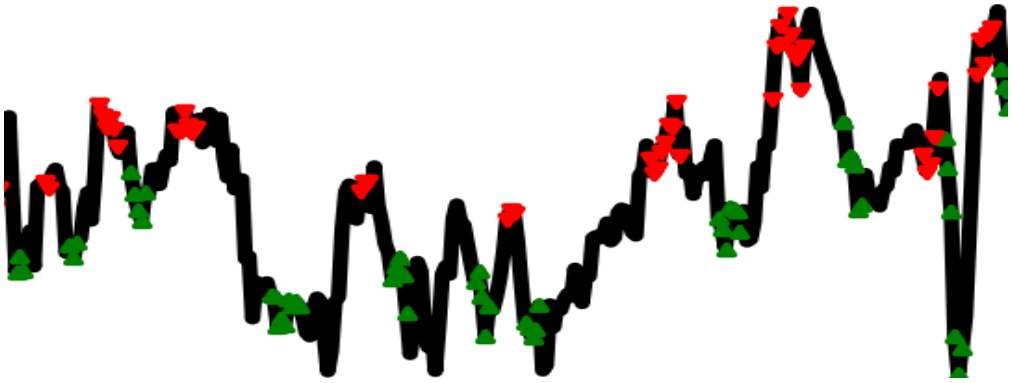


Figure 19: Buy & sell example

Reinforcement learning is zeker een goed alternatief en wij vonden ook informatie over de toepassing ervan op trading met LSTM netwerken dat interessant kan zijn voor verder onderzoek. [3][4]

Het trainen van LSTM modellen is in het algemeen vrij traag vergeleken met veel andere soorten layers vanwege de grote hoeveelheid berekeningen binnen 1 LSTM neuron. Hieronder ziet men de interne structuur van een LSTM neuron zoals eerder uitgelegd, en daaronder een eenvoudiger neuron dat wordt gebruikt in Dense layers.

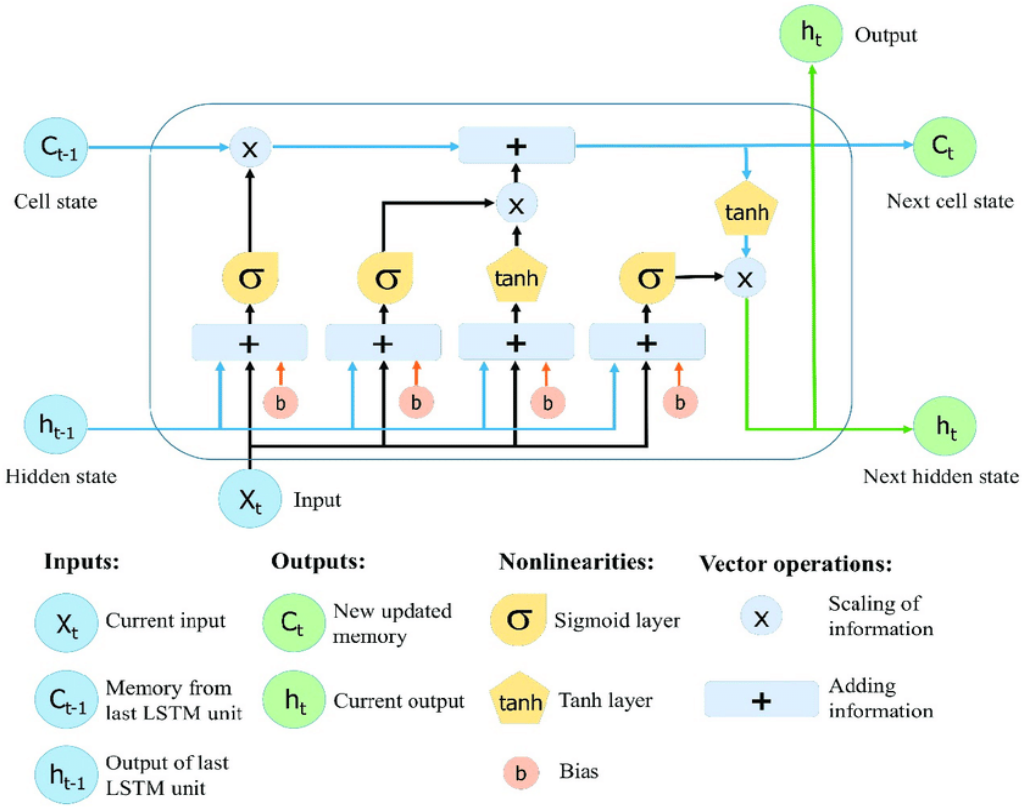


Figure 20: LSTM neuron structure

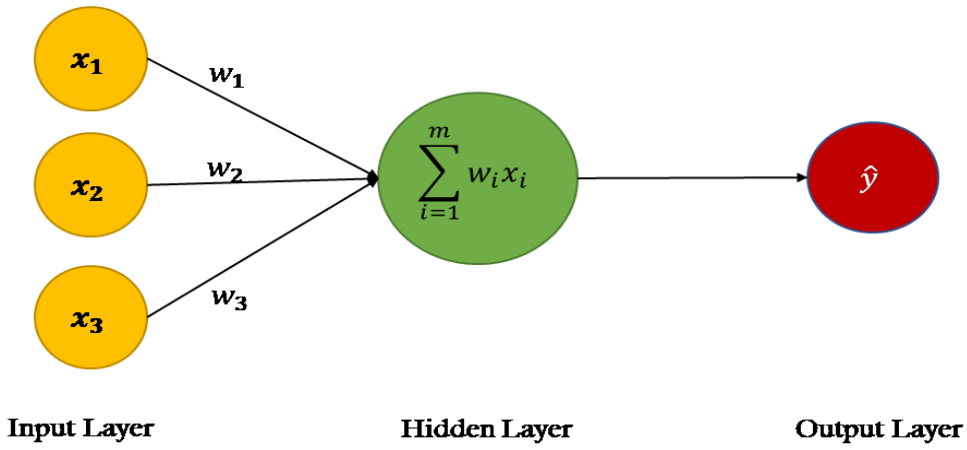


Figure 21: Simple Neuron Structure [21]

2.5 Testing

Een van de belangrijkste zaken voor het bevestigen van een succesvolle training is natuurlijk het testen. Vooral bij ons waar niet direct aan de accuracy tijdens training te zien is of het model effectief winst haalt of niet.

We proberen telkens meerdere munten te testen en de resultaten te vergelijken met vorige modellen dat we gemaakt hadden. Zo was uiteindelijk een 10 layer LSTM model gemiddeld het beste over alle data. We waren hierbij vooral geïnteresseerd in de winst per trade omdat deze hoog genoeg moet liggen om nog winstgevend te zijn na aftrek van trading fees e.d. We zouden deze wel kunnen verwerken in de berekening, maar deden dit niet omdat die kosten afhankelijk zijn van de exchange die zou gebruikt worden.

Wanneer de winst per trade hoog genoeg lag, dan werd gekeken naar de totaal behaalde winst over een bepaalde periode. We lieten het model op een deel van de dataset voorspellingen maken, om dan te kijken hoe goed dit model het gemiddeld doet, om alzo het betere model uit te kiezen en te kijken wat hier anders aan is om zo verder te optimaliseren. Ook werden er van elk model meerdere plots gemaakt met de voorspellingen om te kijken hoe het model deze resultaten behaalde.

2.6 Extra Verbeteringen

2.6.1 Piramidding

Piramidding is een trading strategie waarmee men winst kan optimaliseren en verlies minimaliseren. Door middel van meerdere buy orders gaat men het gemiddelde aankooppunt verlagen om de winst te verhogen wanneer de prijs gaat stijgen, en het verlies bij een minder grote stijging te verlagen. Er hangt echter ook een groter risico aan vast in geval dat het toch blijft dalen, want dan werd al voor vrij veel kapitaal gekocht.

Bijvoorbeeld:

Je gaat bij je eerste aankoop slechts een deel van je kapitaal in een buy order zetten. Als de waarde direct begint te stijgen en je verkoopt, dan heb je direct winst maar dit is natuurlijk niet altijd het geval.

Als de waarde echter zou dalen, kan je nog een tweede deel van je kapitaal investeren. Hierdoor is je gemiddeld aankooppunt ergens halverwege de twee prijzen, naargelang de verdeling van kapitaal over de 2 orders. Dit kan je blijven herhalen zolang de waarde daalt en zolang je kapitaal dat kan geïnvesteerd worden.

Stel dat de waarde uiteindelijk toch stijgt, dan heb je meer winst dan wanneer je had gewacht en enkel je eerste koop order had gehad. Het is niet natuurlijk minder goed dan wanneer je alles in het laagste punt had geïnvesteerd, maar dit is natuurlijk onmogelijk te voorspellen. Zo kan je ook je verlies minimaliseren wanneer je een koop order boven en een koop order onder de waarde van je verkooppunt hebt gezet. Je zou enkel verkopen in zo een situatie als je verwacht dat de waarde weer sterk zou gaan dalen maar dit is natuurlijk ook altijd een gok.

Figure 22: Piramidding example

Na wat testen met piramidding stelden we vast dat dit in bijna alle gevallen voor een verhoogde winst zorgde. Men kan zelf bepalen hoe vaak men het model wil laten piramidden, want bij elke zet investeert men natuurlijk meer geld in waardoor het risico ook stijgt.

2.6.2 Reinforcement Learning

Onze trainingsmethode maakt gebruik van labelled data, dit heeft voordelen maar ook nadelen. De training zal sneller verlopen dan bij reinforcement learning, maar is ook heel gelimiteerd tot hoe goed de labels zijn. Als deze niet optimaal zijn, dan zal het neuraal netwerk ook nooit beter worden dan die labels. En zoals eerder bij training al besproken, kan men bij goede labels toch een lage accuracy hebben, terwijl het model wel winst boekt.

Maar eerst even, wat is reinforcement learning? Bij reinforcement learning gaat men modellen trainen op basis van hun beslissingen en een reward die men aan deze beslissingen geeft. De agent, het neuraal netwerk, zal proberen het environment te leren kennen door trial & error. Elke beslissing krijgt dan een reward afhankelijk van het resultaat van deze handeling. De agent zal proberen deze rewards te maximaliseren. Hoe de rewards berekend worden is volledig zelf te bepalen, maar het is zeer belangrijk, want een slechte rewards functie heeft slechte resultaten, net zoals slechte labelled data bij supervised learning.

Reinforcement learning zou een grote verbetering kunnen zijn in de uiteindelijke performance van het model, maar training zou veel langer duren dan training met een model van LSTM layers, wat nu al vrij aanzienlijk is. Desondanks is dit zeker iets om verder onderzoek bij de toepassing van neurale netwerken op crypto trading.

2.6.3 Automation van training & testing

Een limiterende factor van ons onderzoek was tijd. Hieraan zou kunnen verholpen worden als training en testing geautomatiseerd kon worden. Momenteel werd training telkens manueel gestart met parameters waarvan we dachten dat ze beter waren dan de vorige. Dit duurde dan telkens wel enkele uren om te trainen. Het testen van modellen achteraf duurde ook altijd wel even omdat men een groot deel van data moet testen voor een representatieve weergave van hoe goed een model is. Je kan bv. niet enkel testen op cryptomunten die bijna alleen maar stijgen, en daarop de beste nemen, om achteraf vast te stellen dat die op een downtrend enorm veel verlies maakt.

Als deze workflow automatisch zou gebeuren, van training tot testing, met dan een duidelijk test resultaat achteraf, dan zou dit heel wat tijd besparen en toelaten om meer onderzoek te doen of andere alternatieven te testen.

2.6.4 Explainable AI

Een verdere mogelijke verbetering is Explainable AI. Hiermee proberen we te begrijpen wat het model doet, welke data relevant is voor de uitkomst en welke niet. Hiermee zouden we onze indicatoren kunnen verbeteren en bepalen welke relevant zijn en welke niet. Dit wordt vaak toegepast op CNN's omdat je gemakkelijk een heatmap over de originele image kan leggen om aan te duiden welke pixels belangrijk waren en welke niet. Hieronder een voorbeeld hiervan. [18][19]

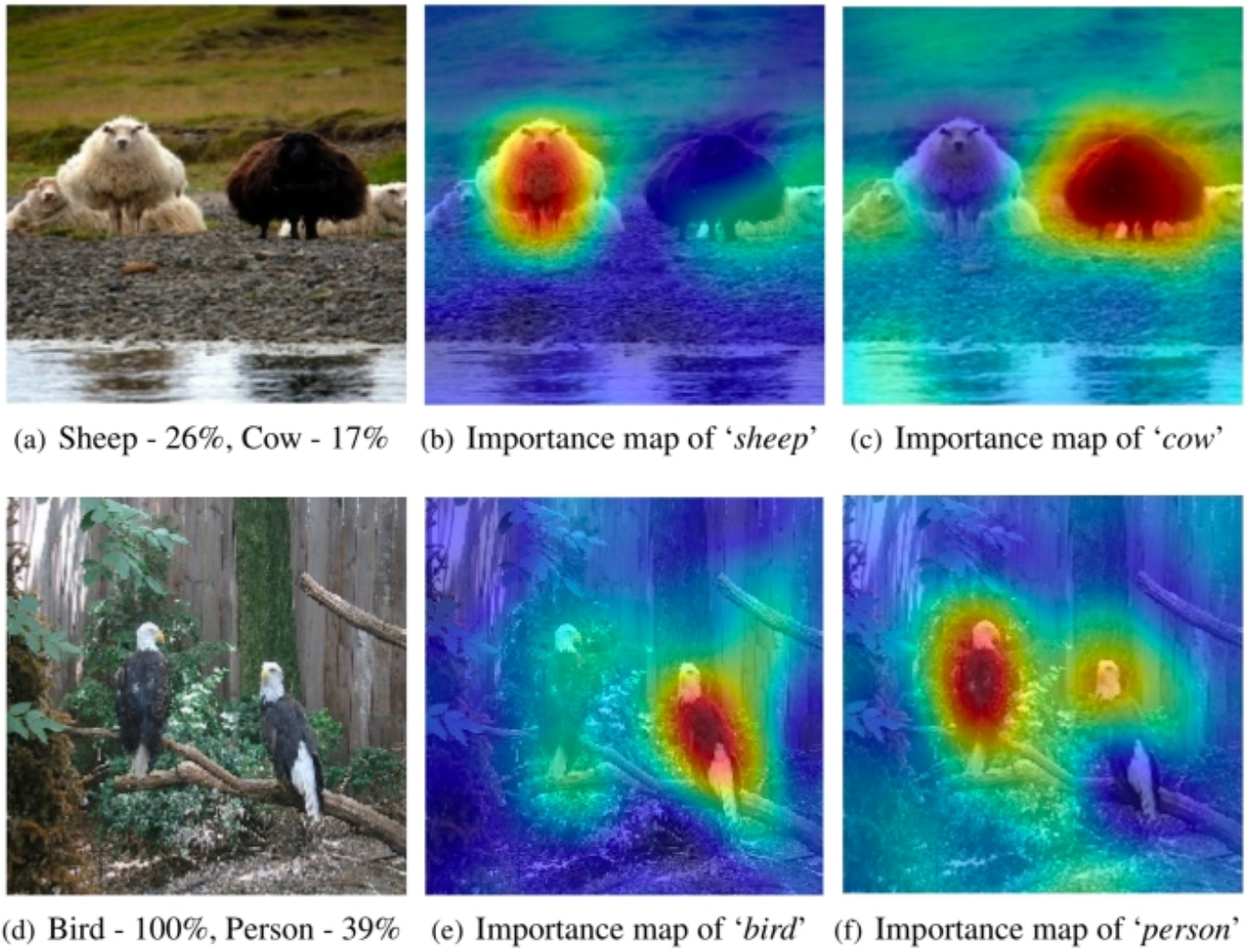


Figure 23: Explainable AI for CNN

3 Technisch onderzoek

3.1 software, tools en programmeertalen

Er word vooral gebruik gemaakt van Python en een deel C++ voor preprocessing van de data. We hebben C++ gekozen voor de preprocessing omdat dit een groot verschil maakte in de snelheid en dit was belangrijk omdat er toch een 15GB aan CSV files verwerkt moest worden.

De belangrijkste libraries die we gebruikt hebben zijn:

- Tensorflow & Keras
- Pandas
- Numpy
- TaLib (in c++ maar bestaat ook voor python) [9]
- Matplotlib
- python-binance (versie 1.0.12)

3.2 Structuur en Workflow

Hieronder een representatie over hoe we van raw data naar trained neural netwerk gaan. We beginnen eerst met de datacollectie met een eenvoudig python script, en deze wordt naar CSV files geschreven.

Dan zijn er 2 mogelijkheden. Als men data wil rescalen naar bijvoorbeeld uur-candles, dan kan men dit doen met ons ticker time rescale programma. Dit is geschreven in C++ en zal binary files wegschrijven in plaats van CSV omwille van 2 redenen. Het volume opslag voor de binary files vergeleken met de CSV files was ongeveer 30% minder, en lezen en schrijven in binary was veel sneller dan in CSV.

Als men de data niet wil rescalen kan men de CSV files rechtstreeks in onze data preprocessor verwerken. Deze is verantwoordelijk voor de indicatorenberekening, scaling en labelling van alle data. Dit wordt dan weer in binary geschreven om dan in te lezen in ons training script om neurale netwerken te trainen.

Nadat we een model getraind hebben zullen we het testen op de data om te zien of dit model het goed doet en of er bepaalde patronen zijn die goed of slecht zijn. We willen namelijk een model dat ook in een downtrend het goed doet, ook al wilt dit zeggen dat men mogelijks in een downtrend helemaal niets doet, wat natuurlijk beter is dan verlies maken.

We maken dan met matplotlib grafieken om te tonen waar de buy en sell orders zijn, en welke de winst is vergeleken met de andere modellen.

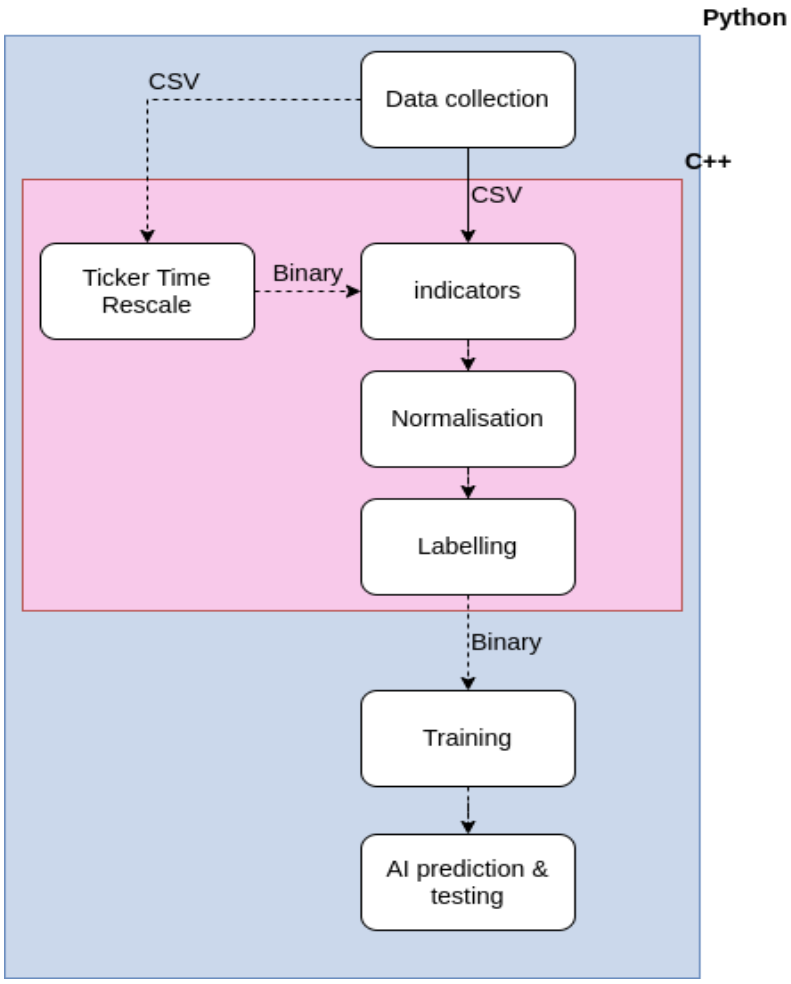


Figure 24: Data flow

3.3 Data Processing

Onze data processing workflow bevat 3 delen

- labelling
- indicator calculation
- normalization

Labelling

In het labelling deel van onze data preprocessing gaan we targets toevoegen aan onze data om achteraf ons model op te trainen, de manier waarop deze labels berekend worden is zeer belangrijk want dit zal een grote invloed hebben op het uiteindelijke model.

Voor onze berekening kijken we naar een stuk data en overlopen 1 voor 1 elke candle. We houden hiervan de laagste bij die we tegenkomen en gaan dan verder. Dan zolang het blijft stijgen houden we de hoogste candle ook bij tot het terug begint te dalen. Als de stijging van de laagste tot de hoogste meer is dan 1% (dit is voldoende om winst te maken, ook met realistische trading fees) dan zetten we een buy target op die laagste candle en een sell target op de hoogste en beginnen we terug opnieuw tot de hele dataset overlopen is. Hieronder bevinden zich een aantal stukken code waar dit gebeurd.

Dit zijn de variabelen die gebruikt worden in de verdere stukken code.

Hier bepalen we ook de min_change, deze variabele is om te bepalen of we willen dat een sell target ten minste 1%, 2%, … boven een buy target staat. We kunnen dit hoger zetten om het model te forceren op langere termijn te traden en minder de kleine schommelingen.

Hieronder bevind zich een kort stuk code dat zal bepalen of een candle stijgt of daalt door te checken of de close meer is dan 0 of niet (deze zijn reeds gescaled tussen -1 en 1).

deze worden dan aan lijsten toegevoegd om verder te gebruiken voor labelling.

```
std::vector<double> cum_down;
std::vector<double> cum_up;

std::vector<double> cum_down_buy;
std::vector<double> cum_up_buy;

std::vector<double> cum_down_sell;
std::vector<double> cum_up_sell;

double min_change = 0.01; // 1%

double last_max = 0;
bool allow_buy = false;
int last_max_index = 0;

double last_min = 0;
bool allow_sell = false;
int last_min_index = 0;
```

Figure 25: Labelling variables

```
if (candle->m_close > 0)
{
    double prev = cum_up.size() ? cum_up[cum_up.size() - 1] : 0;

    cum_up.push_back(prev + candle->m_close);
    cum_up_buy.push_back(prev + candle->m_close);
    cum_up_sell.push_back(prev + candle->m_close);
}
else
{
    double prev = cum_down.size() ? cum_down[cum_down.size() - 1] : 0;

    cum_down.push_back(prev + fabs(candle->m_close));
    cum_down_buy.push_back(prev + fabs(candle->m_close));
    cum_down_sell.push_back(prev + fabs(candle->m_close));
}
```

Figure 26: Labelling cumulative candle lists

Hier bevind zich het grootste deel van de labelling. We gaan hier telkens over elke candle loopen, er word dan gecheckt of deze stijgt of daalt net zoals hierboven.

Dan wordt er bepaald of de stijging vergeleken met de vorige buy voldoende is om een sell te plaatsen. Vanaf dan gaan we kijken tot waar het blijft stijgen.

Als het stopt met stijgen en de daling hierna is ook groter dan onze op voorhand bepaalde minimum change, dan word er een sell order gezet op deze piek. Voor de buy targets gebeurd hetzelfde maar omgekeerd.

```
for (size_t i = 0; i < candles->size(); i++)
{
    std::unique_ptr<candle>& candle = candles->at(i);

    if (cum_up.size() && cum_down.size())
    {
        double prev_up = cum_up[cum_up.size() - 1];
        double prev_down = cum_down[cum_down.size() - 1];

        double prev_up_buy = cum_up_buy[cum_up_buy.size() - 1];
        double prev_down_buy = cum_down_buy[cum_down_buy.size() - 1];

        double prev_up_sell = cum_up_sell[cum_up_sell.size() - 1];
        double prev_down_sell = cum_down_sell[cum_down_sell.size() - 1];

        if (candle->m_close > 0)
        {
            if (prev_up - prev_down > min_change)
            {
                if (prev_up_sell - prev_down_sell > last_max)
                {
                    last_max_index = i;
                    last_max = prev_up_sell - prev_down_sell;
                    allow_buy = true;
                    cum_up_buy.clear();
                    cum_down_buy.clear();
                    cum_up.clear();
                    cum_down.clear();
                }
                if (allow_sell)
                {
                    candles->at(last_min_index)->m_target = eTarget::SELL;
                    last_min_index = 0;
                    last_min = min_change;
                    allow_sell = false;
                    cum_up_buy.clear();
                    cum_down_buy.clear();
                }
            }
        }
        else if (candle->m_close < 0)
        {
            if (prev_down - prev_up > min_change)
            {
                if (allow_buy)
                {
                    candles->at(last_max_index)->m_target = eTarget::BUY;
                    last_max_index = 0;
                    last_max = min_change;
                    allow_buy = false;
                    cum_down_sell.clear();
                    cum_up_sell.clear();
                }
                if (prev_down_buy - prev_up_buy > last_min)
                {
                    last_min_index = i;
                    last_min = prev_down_buy - prev_up_buy;
                    allow_sell = true;
                    cum_down_sell.clear();
                    cum_up_sell.clear();
                    cum_up.clear();
                    cum_down.clear();
                }
            }

        }
    }
```

Figure 27: Labelling calculation

Het resultaat ziet er dan zo uit, afhankelijk van de min_change zijn de targets enkel op grote prijsverschillen of ook op kleine prijsverschillen.

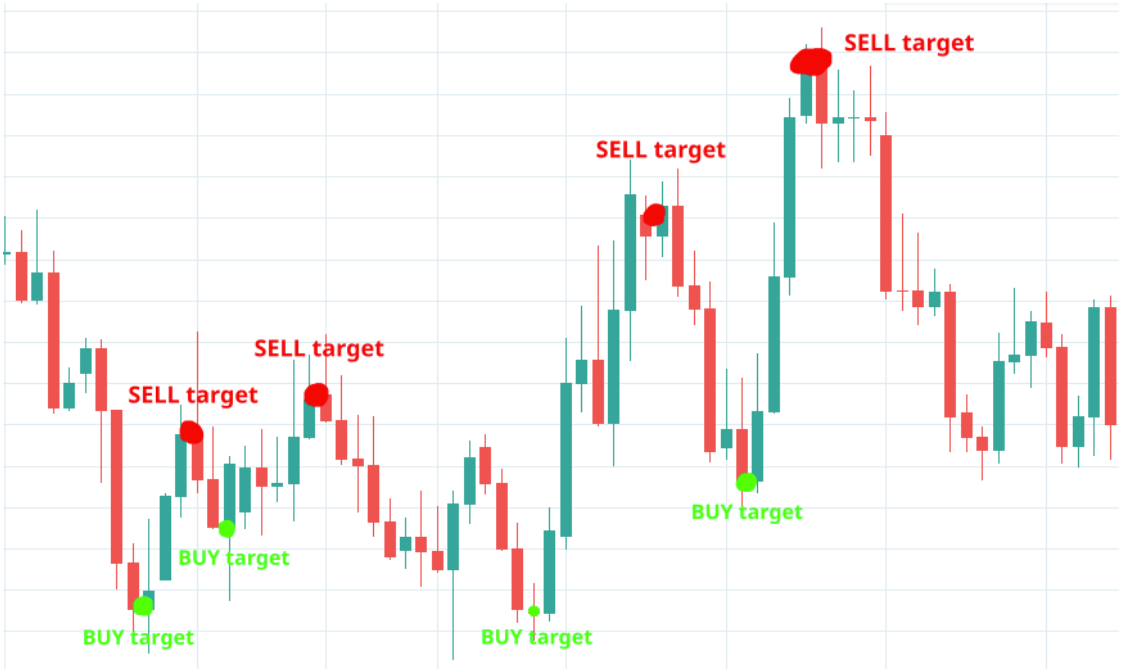


Figure 28: labelling example

Indicators

Voor onze indicators maken we gebruik van een heel bekende library TA-lib, opgestart als hobby project in 1999 door Mario Fortier, deze is gelicenseerd onder de BSD license. Dit laat het gebruik toe in open-source en commerciele producten.

Door TA-lib is de indicator berekening een heel eevnoudig process. Je moet enkel de data meegeven en de parameters van die specifieke indicator. Afhankelijk van de indicator kan dit tussen 1 en 4 parameters liggen.

Om even te schetsen hoe dit eruit ziet in code. Hieronder bevind zich een van onze indicator berekeningen, namelijk de MACD. Deze indicator verwacht 3 parameters. De fast, slow en signal period. Deze bepalen met hoeveel historische gegevens de Moving averages en het signal berekend worden. Ta-lib geeft je de data terug in vooraf gedefinieerde variabelen.

```
void calculate_macd(const size_t fast_period = 12, const size_t slow_period = 26, const
size_t signal_period = 9)
{
    double* tmp_macd = new double[m_alloc_size];
    double* tmp_macd_signal = new double[m_alloc_size];
    double* tmp_macd_hist = new double[m_alloc_size];

    int beginIdx, endIdx;
    TA_MACD(0, m_alloc_size, m_close, fast_period, slow_period, signal_period, &beginIdx,
    &endIdx, tmp_macd, tmp_macd_signal, tmp_macd_hist);

    for (size_t i = beginIdx; i < endIdx; i++)
    {
        const std::unique_ptr<candle>& candle = m_candles->at(i);

        candle->m_macd = tmp_macd[i];
        candle->m_macd_signal = tmp_macd_signal[i];
        candle->m_macd_hist = tmp_macd_hist[i];
    }

    delete[] tmp_macd;
    delete[] tmp_macd_signal;
    delete[] tmp_macd_hist;
}
```

Figure 29: MACD calculation TA-lib

Scalen

Een heel belangrijk deel van data processing bij neurale netwerken is het correct scalen van de data. Als de data niet goed gescaled is kan het model moeilijker de correcte weights en biases vinden. Het model moet zich trainen met de gegevens waarvan de waarden op dezelfde schaal bevinden. bijvoorbeeld Bitcoin heeft een prijs van ongeveer 30000 euro, terwijl dat bij Dogecoin 0.30 euro is. Als je model getraind is met Bitcoin data dan zal het de prijswaardes van Dogecoin minder relevant beschouwen, ook al is dit niet het geval.

De manier van scaling is afhankelijk van de data die men gebruikt. Wij gebruiken het percentage verandering van het ene data punt naar het volgende. Dit is de meest logische vorm van scaling voor onze data. Zo blijft dit gelijk over alle munten en is het niet afhankelijk van de maximum en minimum waarden.

De meeste waarden scalen we dus door het procentuele verschil met de vorige candle te berekenen. Bij de MFI en RSI indicatoren delen we deze gewoon door 100 omdat deze al op een schaal van 0 tot 100 staan.

Hieronder ziet men het stuk code waarmee dit gebeurt. Deze functie bevindt zich in onze candle struct waarin alle data per candle zit.

```
void normalize(candle* other)
{
    // don't update timestamp since we're on the newest candle of the two
    auto calc = [](double a1, double a2) { return a1 == 0 ? a1 : (a2 - a1) / a1; };

    this->m_open = calc(this->m_open, other->m_open);
    this->m_close = calc(this->m_close, other->m_close);
    this->m_high = calc(this->m_high, other->m_high);
    this->m_low = calc(this->m_low, other->m_low);
    this->m_volume = calc(this->m_volume, other->m_volume);

    this->m_adosc = calc(this->m_adosc, other->m_adosc);
    this->m_atr = calc(this->m_atr, other->m_atr);

    this->m_macd = calc(this->m_macd, other->m_macd);
    this->m_macd_hist = calc(this->m_macd_hist, other->m_macd_hist);
    this->m_macd_signal = calc(this->m_macd_signal, other->m_macd_signal);

    this->m_upper_band = calc(this->m_upper_band, other->m_upper_band);
    this->m_middle_band = calc(this->m_middle_band, other->m_middle_band);
    this->m_lower_band = calc(this->m_lower_band, other->m_lower_band);

    this->m_mfi /= 100;
    this->m_rsi /= 100;

    this->m_difference_lowhigh = calc(other->m_low, other->m_high);
    this->m_difference_openclose = calc(other->m_open, other->m_close);
}
```

Figure 30: Normalize candle code

3.4 Model opbouw

Het model ging een type RNN worden, maar hoe of wat we exact zouden gebruiken was bij de aanvang nog niet zo duidelijk. Na wat research zijn we dan voor LSTM netwerken gegaan. Hierin hebben we wat geëxperimenteerd met de grootte van layers, aantal layers, training time en learning rate. Uiteindelijk kregen we een vrij goed 3 Layer LSTM model. We wilden natuurlijk ook weten of groter beter zou zijn, en dit was in ons onderzoek slechts deels het geval. Het beste model was uiteindelijk een 10 layer LSTM model, maar die was algemeen hooguit een beetje beter dan het 3 layer model, en heel soms was het zelfs minder goed.

Bij LSTM layers is het uitzonderlijk dat meer dan 3 layers gebruikt worden. Het is mogelijk dat - mits meer training of wat veranderingen aan de hyperparameters - we een even goed of beter 3 layer model kunnen bekomen. Uit onderzoek blijkt ook dat grotere modellen zeker niet altijd beter zijn, vooral bij NLP is dit opvallend. [7][8]

4 Reflectie

4.1 Resultaat

Na het onderzoek was het resultaat enerzijds beter dan verwacht, maar op bepaalde gebieden ook teleurstellend. De resultaten zijn niet echt representatief voor wat men zou kunnen halen bij echte crypto trading, want hierbij komen nog allerlei onkosten kijken (fees, …) alsook kleine variaties in prijs tijdens het voorspellen en versturen van de orders. Het model kan in de meeste situaties wel correct inschatten wanneer men best zou aankopen of verkopen, wat wel indrukwekkend is gelet op de onvoorspelbaarheid van de crypto markt en rekening houdend met onze korte onderzoeksperiode en beperkte training.

Zolang er voldoende volatiliteit is in de prijs van een munt kan het model zeer goed de aankopen en verkopen aanduiden. In een uptrend is dit zeker geen probleem, maar in een downtrend werkt dit niet altijd even goed. Het model gaat hier vaak blijven kopen en verkopen op momenten waar het eigenlijk verlies maakt. Dit is te wijten aan het feit dat ons model tijdens training geen besef heeft van winst. Bij reinforcement learning zou men hogere rewards kunnen toekennen aan acties met hogere winst, en zo mogelijks het model aanleren dat dit beter is.

Het model haalde in ongeveer twee maanden tijd bij verschillende munten meer dan 150% winst. Zo'n grote winst op een korte periode ligt aan de volatiliteit van crypto, waardoor het model zelfs in een downtrend soms een reeks kleine trades kan maken met winst. Iemand die manueel koopt en verkoopt kan natuurlijk ook deze winst halen, of meer, en dus outperformed het model de goede traders zeker nog niet, maar voor een automatisch systeem is dit toch al vrij goed. Het grootste voordeel is dat men zelf niet constant de prijs in de gaten moet houden zoals traders wel dagelijks moeten doen.

Het is moeilijk onze resultaten te vergelijken met gegevens online te vinden van andere algoritmische trading strategieen omdat de exacte modellen, indicators en data gebruikt niet altijd beschikbaar zijn. Het is ook niet altijd duidelijk welke trading fees gebruikt zijn, als er al fees gebruikt zijn in de berekening van de winsten. De resultaten van machine learning en algoritmische trading bots lopen ver uit elkaar van slechts een 10% per jaar tot meer dan 100% per jaar. [22][23][24][25][26]

Het model kan zeker nog verbeterd worden, want momenteel koopt het model nog te snel aan. Het zou beter meestal nog even wachten. Mogelijks met wat extra training of met het toepassen van een andere trainingsmethode zoals reinforcement learning kan dit waarschijnlijk wel beter worden. Ook hebben we nog niet kunnen experimenteren met Explainable AI: een manier om te onderzoeken welke inputs het meest invloed hebben op de voorspellingen van het model. Dit zou een beter inzicht kunnen geven in het effectieve nut van de indicatoren of de prijsdata.

Als we even kijken naar hoe goed onze modellen het doen op een deel van onze dataset.

Hieronder eerst de resultaten van het 10 layer LSTM model en dan het 3 layer LSTM model.

Average % per hour: 0.6742 %

Average % per trade: 0.0594 %

Average # trades per hour: 11.34

Average % per hour: 1.1511 %

Average % per trade: 0.0553 %

Average # trades per hour: 20.81

Ogenschijnlijk haalt het 3 layer model betere resultaten omdat het per uur een hoger percentage winst haalt. Maar dit is een vertekend beeld, want hier zijn namelijk nog geen onkosten in verwerkt, en als men dan kijkt naar het average percentage per trade ga men meer overhouden per trade bij het 10 layer model. We hebben ook langere testen uitgevoerd waarbij er wel transactiekosten werden verrekend, en daarbij kan men dan duidelijk zien dat het % per trade hoog genoeg moet zijn om deze kosten te overbruggen, en dat dit uiteindelijk een belangrijke rol speelt. Als het % per trade niet hoog genoeg is zullen de positieve trades niet genoeg opbrengen om de occasionele negatieve trades op te vangen.

Als we hieronder ook even kijken naar een plot waarop men de sell en buy orders ziet is het al snel duidelijk dat het model nog veel te leren heeft. In het algemeen geeft het model wel degelijk aan dat men moet kopen na een downtrend en verkopen na een uptrend, maar dit gebeurt vaak veel te snel. Al te vaak wordt er een buy of sell uitgevoerd voordat dat de prijs voldoende gedaald of gestegen is

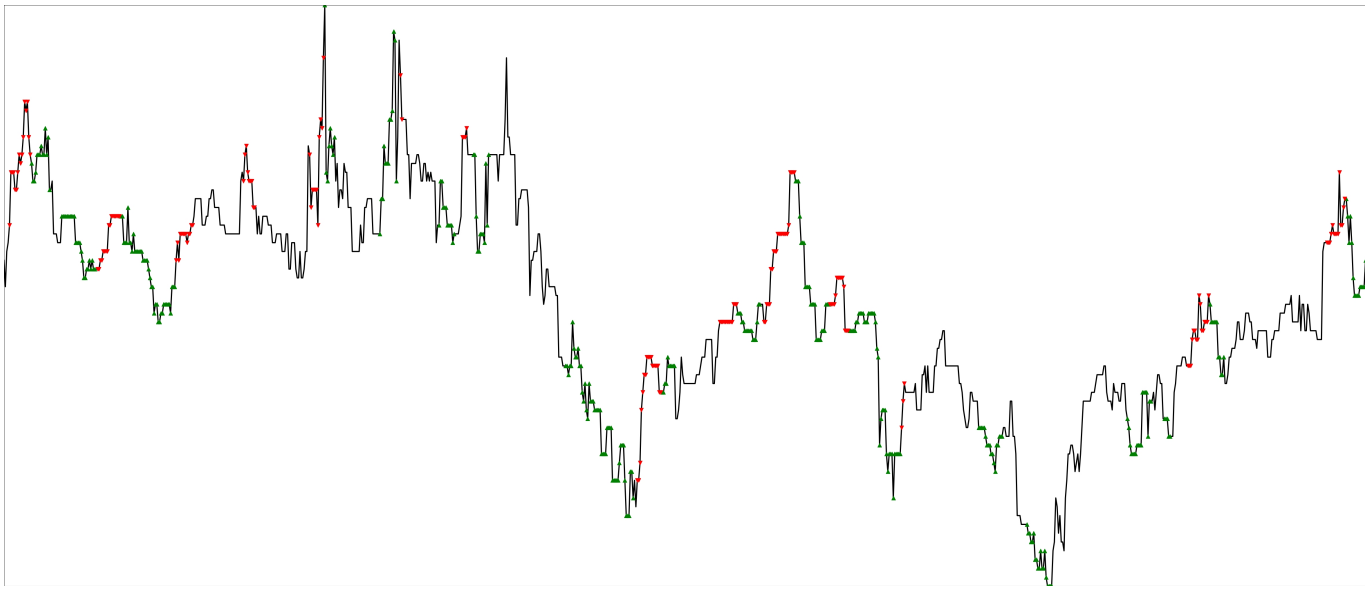


Figure 31: Model Predictions

4.2 Sterke en zwakkere punten

Sterke punten

Dankzij de kleine candle size en de snelheid van het model kan deze op een zeer korte periode al winst maken en heeft deze geen al te sterke computer nodig om het model te runnen. Het getrainde model gebruikt ongeveer 1 GB aan VRAM op een GPU maar er zal rond 1.5 GB nodig zijn inclusief de data.

De data die gebruikt wordt heeft dezelfde structuur als de data van andere exchanges, zodat de BOT niet alleen op de Binance exchange kan gebruikt worden, waarop deze getraind is, maar evenzeer op andere exchanges zoals bijvoorbeeld Crypto.com. Dit zou in theorie even goed moeten werken, maar men moet wel rekening houden met het volume. Op Binance wordt vooral USDT en BUSD gebruikt en hier heb je een groter volume en dus een betere weergave van de aankopen en verkopen. Op Crypto.com daarentegen wordt er veel USDC gebruikt en is hier het volume hoger. Een te laag volume aan USDT en BUSD kan een negatieve impact hebben op de berekening van de indicatoren en op het neurale netwerk.

Zwakke punten

Het model maakt heel veel kleine trades, wat nadelig is in geval van hoge trading fees, omdat men die onkosten moeilijk kan compenseren met de kleine winsten die met de trades worden gemaakt. Het model zou eigenlijk iets minder vaak moeten aankopen, en deze aankopen op lagere punten doen. Dit is mogelijks te verhelpen door de minimum change bij het labellen van de data te verhogen, waardoor het neuraal netwerk leert om grotere trades te doen in plaats van bij elke kleine schommeling.

Vergeleken met algoritmisch traden hebben we wel een NVIDIA GPU nodig, wat een initieel hogere kost vereist om het model op te zetten voor constant gebruik, alsook een hoger stroomverbruik.

4.3 Bruikbaarheid en implementatie

Het model is in zijn huidige toestand wel bruikbaar, maar na berekeningen van de uiteindelijke winst rekening houdend met de verhandelingskosten ligt de gerealiseerde winst wat lager en is het resultaat soms zelfs negatief. Dit is afhankelijk van de munt waarop getest wordt en hoe hoog de onkosten liggen op het gekozen exchange. Dit is deels ook omdat het model vaak nog te vroeg aankoopt waardoor het heel wat potentiele winst laat liggen.

Ik denk wel dat dit kan verholpen worden door verbeteringen aan de layout van het model of door meer training, waardoor het dan een vrij goed model kan worden.

De implementatie van het model is vrij gemakkelijk. Er zijn veel crypto exchanges die publieke API's aanbieden alsook python libraries, waardoor men makkelijk dit model met Tensorflow zou kunnen gebruiken op real-time data die men opvraagt via de API. Binance en Crypto.com zijn twee van de bekendste crypto exchanges met een goede API.

4.4 Alternatieven

Een alternatief dat vaak wordt toegepast op zowel aandelen en crypto is natuurlijk algoritmisch traden. [22][23][24][25][26] Dit maakt meestal gebruik van indicatoren, heel vaak dezelfde die als inputs gebruikt worden in ons model. Van deze indicatoren wordt dan afgeleid of de markt in een down of een up trend zit, en probeert men te voorspellen wanneer dit gaat omdraaien. Dit is wel een goede optie voor een meer voorspelbare trading bot waarbij men beter begrijpt wat de bot juist doet en waarbij men dit gemakkelijk zelf kan aanpassen. Ik heb dit zelf in het verleden ook al gebruikt en kan bevestigen dat dit relatief eenvoudig is en dat het wel degelijk winstgevend kan zijn.

4.5 Meerwaarde

Dit onderzoek zal een beperkte meerwaarde bieden aan de maatschappij maar kan om economische redenen wel interessant zijn voor zowel individuen als bedrijven die kapitaal willen investeren maar zelf niet de tijd, kennis of zin hebben om manueel de markt te volgen en trades te maken wanneer nodig. Het model kan evengoed gebruikt worden voor crypro's als voor aandelen.

Het is echter niet mogelijk om dit commercieel aan te bieden als een service, zeker niet als er slechts één neuraal netwerk wordt gebruikt. Men zou dan met een zeer groot kapitaal aankopen en verkopen doen, afhankelijk van hoeveel mensen het gebruiken en wat hun ingezette kapitaal is natuurlijk. Dit zou dan voor prijs manipulatie zorgen, vooral bij munten met een kleinere marktkapitalisatie.

Fictief voorbeeld: Stel de totale waarde van alle Dogecoins is 100 000 euro en men koopt er voor 10 000 euro, zijnde 10% van het totale aanbod. Dit zou voor een enorme prijsstijging zorgen, met negatieve effecten op je eigen winsten. Dit kan deels verholpen worden door het kapitaal te verdelen over een grote hoeveelheid munten maar als men geen limiet zet op de inzet kan dit negatieve gevolgen hebben.

4.6 Vervolgonderzoek

Het is vooral belangrijk om meer onderzoek te doen naar welke layers en welk aantal neuron het beste werken en hoeveel training er juist nodig is. Zoals reeds bewezen in het onderzoek naar meerdere soorten modellen, is het niet altijd beter om een groter model te gebruiken maar kan een kleiner model vaak betere performance halen wanneer het beter opgebouwd of getraind is, en misschien omdat de data beter is.

Een onderwerp dat ik zeker wil verkennen is Reinforcement Learning [10]. Ik ben reeds gestart met het maken van een eigen OpenAI gym environment voor crypto trading die te vinden is op Github. Ik ben er van overtuigd dat dit goede resultaten kan opleveren als ik mijn reward functies op punt krijg. Een Mede student Tuur Vanhoutte heeft al wat meer ervaring met Reinforcement learning vanuit zijn research project en heeft dit toegepast met LibTorch, de c++ versie van PyTorch. Hij ondervond enorme verbeteringen in training snelheid in c++. Dit zou ik zeker ook verder willen verkennen.

Verder denk ik dat Explainable AI zeer belangrijk is voor elk onderzoek met neurale netwerken. Jammer genoeg was er niet voldoende tijd om dit uit te werken, maar ik wil dit zeker nog verder onderzoeken.

4.7 Feedback van externen

Ik heb aan een aantal mensen enkele vragen gesteld en feedback gevraagd over mijn onderzoek. Hieronder de feedback van Nick Langens, Computer Science student aan de KUL, die ook onderzoek deed naar trading, maar dan met reinforcement learning en machine learning in plaats van deep learning.

Ook Yente De Wael, master student Computer Science aan de VUB, crypto trader in zijn vrije tijd, heeft ook ervaring met algoritmisch traden.

1. Wat is uw ervaring met trading, zowel met en zonder AI?

Nick heeft zowel ervaring met manueel en algoritmisch traden en uit zijn ervaring heeft het economisch of politiek nieuws soms een grotere effect op de prijzen dan de effectief observeerbare patronen die je kan afleiden uit prijsdata. Daarom is het moeilijk om een onderzoek uit te voeren op trading want het implementeren van real-time media nieuws data is niet betrouwbaar en moeilijk te implementeren.

Yente heeft ook ervaring met zowel manueel en algoritmisch traden en heeft zelf al een algoritmische trading BOT gemaakt voor de Binance Exchange. Uit zijn ondervindingen is het mogelijk om enkel op basis van prijsdata goede voorspellingen te maken, maar kunnen nieuwsartikels of tweets soms een hulp zijn. In het bijzonder tweets of artikels van Elon Musk hebben vaak een grote impact op de prijs van DogeCoin, een cryptomunt die ontstaan is als grap maar snel populair werd.

2. Vind u dat ik mijn onderzoek correct heb aangepakt of had u dingen anders gedaan en indien ja, wat?

De aanpak met neurale netwerken vond Nick vooral interessant, maar hij mist een beetje de quantitatieve vergelijkingen met andere methoden zoals machine learning, algoritmisch traden en andere model soorten.

Yente vond de aanpak goed maar na het bekijken van de source code en de manier waarop we alles uitvoeren heeft hij wel wat feedback die we meestal manueel moeten doen. We moeten namelijk zelf de preprocessing uitvoeren en zelf de juiste folders meegeven. Dan moeten we zelf de training runnen en weeral manueel de folders voor de juiste data telkens manueel aanpassen. Idem voor testing. Dit kan veel beter natuurlijk door het gehele proces meer te automatiseren.

Ik vind dit zeker een terechte feedback. Het zou ook beter zijn als we wat meer vergelijkingen konden toevoegen, maar voor algoritmisch en machine learning zou minstens nog maand bijkomend onderzoek nodig zijn. Zeker bij algoritmisch omdat men dan zelf manueel de parameters moet aanpassen van de indicatoren en de beslissingen voor kopen en verkopen.

3. Denkt u dat reinforcement learning betere resultaten zou opleveren dan supervised learning?

Volgens Nick is dit moeilijk correct te doen omdat het model een zo realistisch mogelijke environment nodig heeft om effectief correct te leren wat te doen. Indien er fouten zitten in de environment, hoe klein dan ook, zal het model deze aanleren en misbruiken wanneer mogelijk. Reinforcement learning is mogelijk maar je moet goed opletten dat alle data zo juist mogelijk zijn.

Yente heeft zelf niet zo veel ervaring met reinforcement learning maar denkt dat het zeker de problemen kan oplossen die we ondervonden met labelled training.

4. Hebt u ervaring met indicatorsen en indien ja, wat denkt u van de combinatie van indicatoren, en hebt u hier feedback op?

Nick heeft een beperkte ervaring met het gebruiken en combineren van indicatoren, maar had hier geen feedback over.

Yente heeft redelijk veel ervaring met Indicatoren vanwege zijn eigen trading Bot op de Binance exchange. Hij heeft zelf een set van heel gelijkaardige indicatoren gekozen en vindt onze keuze dus zeer goed. De combinatie momentum, trend en volatiliteit vindt hij ook goed.

5. Hebt u ervaring met LSTM modellen en wat denkt u van de manier waarop ik LSTM heb gebruikt?

Nick heeft niet genoeg ervaring hiermee om er een oordeel over te vellen.

Yente zijn ervaring met LSTM’s is beperkt tot geziene leerstof in de les en heeft zelf geen ervaring op trading met LSTM’s, maar vanuit zijn theoretische kennis ziet hij dit als een goede toepassing van LSTM’s.

5 Advies

5.1 Introductie

Hier geef ik graag enkele aanbevelingen aan wie wil verder bouwen op dit onderzoek of een gelijkaardig onderzoek zou willen starten. Zaken die ik anders zou doen mocht ik opnieuw beginnen, en de dingen die ik zelf ook nog graag zou uitwerken om het resultaat te verbeteren zullen hier behandeld worden.

5.2 Risico

Automated trading heeft natuurlijk voor- en nadelen [11]. Het is belangrijk deze goed af te wegen en de mogelijke problemen en nadelen te minimaliseren.

Traden met echt geld zou ik enkel en alleen doen als je terdege bewust bent van alle risico's. Er bestaat altijd een kans dat je al je inzet verliest. Men kan natuurlijk ook zonder geld onderzoek doen op historische data of op real time data, maar dan zonder de effectieve trades uit te voeren. Hiervoor kan men gebruik maken van Binance met een Demo account of Alpaca API, hun Paper Trading API.

5.3 Voor Wie Is Dit?

Dit onderzoek is bedoelt voor AI onderzoekers, hobby traders of eender wie zich wil verdiepen in zowel trading en/of neurale netwerken. Een voorkennis van trading is aangeraden, zeker op vlak van indicatoren, maar dit is niet moeilijk aan te leren en er is veel info online te vinden over hoe je indicatoren juist moet gebruiken.

5.4 Model

LSTM modellen zijn het meest populair bij trading modellen, en ik zou het ook aanraden, maar het is zeker ook mogelijk het probleem op andere manieren aan te pakken. Wij hebben zeker nog geen optimaal model in dit onderzoek, en ik raad aan zelf te testen wat goed werkt en wat nog kan verbeterd worden.

5.5 Data

Voor het verzamelen van data raad ik de Binance Exchange api aan. Deze is stabiel, snel, gratis te gebruiken, en omdat Binance al een aantal jaar bestaat vind men hier voldoende historische data voor verschillende munten

De indicatoren de men al dan niet wil toevoegen zijn vrij te kiezen. Ik koos voor indicatorens die ik al ken en gebruikt heb omdat ik weet dat deze een goede aanwijzing geven voor het gepaste moment van kopen en verkopen. Zonder indicatoren zou het model veel moeilijker correcte buy en sell targets kunnen voorspellen van.

5.6 Aanbevelingen

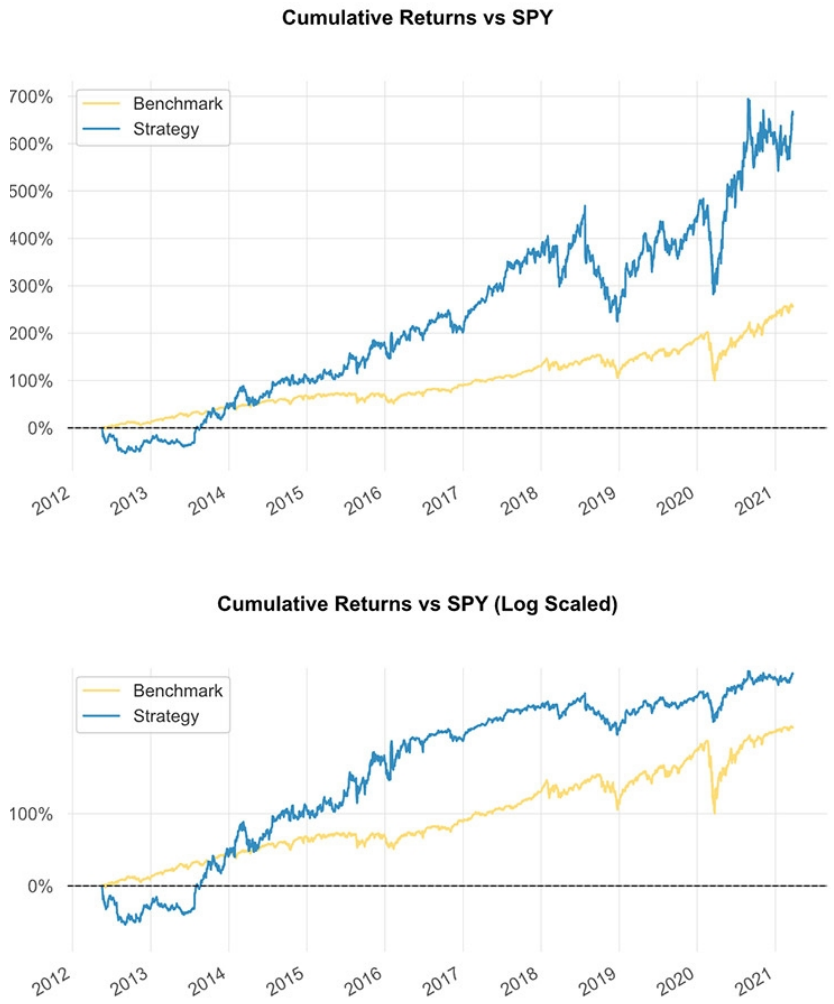
Wanneer mogelijk zou ik reinforcement learning willen aanraden. De problemen die zich voordoen bij labeled training kunnen hiermee verholpen worden. Er is dan natuurlijk wel meer tijd voor nodig voor de training.

Een van de belangrijkste uitbreidingen zou Explainable AI zijn. Er werd eerder al vermeld waarom dit zo belangrijk is. Dit was nog niet toegepast in dit onderzoek waardoor we eigenlijk meer 'gokten' naar en correcte model layout en welke data te gebruiken, wat natuurlijk kan verbeterd worden. Door een

vorm van Explainable AI toe te passen zou men indicatoren die niet relevant zijn kunnen ignoreren of andere indicators toevoegen die wel een meerwaarde bieden.

Indien je effectief zou willen traden met ons neuraal netwerk raad ik het Binance exchange platform aan: dit heeft de laagste kosten voor kleine trades. Mocht je een groot kapitaal hebben en dus een vrij hoog maandelijks trade volume investeren, dan is het Crypto.com platform mogelijks beter omdat men hierop minder onkosten betaalt voor hoge trade volumes.

Om het testen te verbeteren en duidelijkere vergelijkingen te kunnen maken tussen modellen raad ik de library Quantstats aan. Hiermee kan men de performance van het model veel beter onderzoeken vanwege de grote lijst aan grafieken en parameters die ter beschikking zijn. Zo worden onder andere de drawdown, expected returns, profit factor en nog veel andere reultaten berekend en weergegeven op grafieken. Hieronder een voorbeeld van een deel van een Quantstats report. [20]



Key Performance Metrics

Metric	Strategy	Benchmark
Risk-Free Rate	0.0%	0.0%
Time in Market	100.0%	100.0%
Cumulative Return	660.21%	255.34%
CAGR%	25.75%	15.4%
Sharpe	0.8	0.95
Sortino	1.21	1.32
Sortino/√2	0.86	0.93
Max Drawdown	-53.62%	-33.72%
Longest DD Days	532	272
Volatility (ann.)	37.32%	16.6%
R^2	0.23	0.23
Calmar	0.48	0.46
Skew	0.95	-0.7
Kurtosis	18.55	17.87
Expected Daily %	0.09%	0.06%
Expected Monthly %	1.91%	1.19%
Expected Yearly %	22.49%	13.52%
Kelly Criterion	4.86%	6.4%
Risk of Ruin	0.0%	0.0%
Daily Value-at-Risk	-3.75%	-1.66%
Expected Shortfall (cVaR)	-3.75%	-1.66%
Gain/Pain Ratio	0.17	0.21
Gain/Pain (1M)	0.97	1.55

Figure 32: Quantstats report

5.7 Tips

Wanneer mogelijk raad ik aan de data in binair formaat op te slaan omdat dit de lees- en schrijfsnelheid aanzienlijk verbetert en het opslagvolume vermindert.

Automatiseer zo mogelijk het training en testing proces zodat dit minder manueel werk vergt. Dit hadden wij bij het begin niet gedaan, en hierdoor moesten we vaak manueel stukjes code aanpassen om het juiste model te testen. Test ook zeker op voldoende data. Verschillende munten reageren soms volledig anders omdat ze minder of meer volatiel zijn of eerder up of down trends hebben.

Piramidding is een eenvoudige toevoeging. Het mag zeker niet vergeten worden want het kan de eventuele winst in belangrijke mate vergroten.

Tot slot, focus je niet op de totale winst wanneer de onkosten niet in de berekeningen zijn opgenomen. Focus je dan eerder op winst per trade. Enkel wanneer de onkosten in de berekeningen incluis zijn is de berekende totale winst realistisch.

6 Conclusie

Dit onderzoek was zeer interessant, om te de mogelijkheden van AI te ontdekken op vlak van time-series voorspellingen in een onvoorspelbare omgeving, maar vooral omwille van mijn persoonlijke interesse voor crypto trading. Door de beperkte tijd van dit onderzoek is het werk zeker nog niet af, maar ik zal hierop zeker verder bouwen na deze bachelorproef.

Tijdens het onderzoek zijn er nogal wat problemen en mogelijke verbeteringen duidelijk geworden. Dit zal me toelaten wat bij te sturen om een nog beter model te bekomen dat hopelijk effectief ingezet kan worden op een echt trading platform. Alhoewel het nu al winstgevend zou zijn is het nog net niet betrouwbaar genoeg om in te zetten met echt geld. Bij een langdurige downtrend zou het wel eens grote fouten kunnen maken waardoor mogelijks veel geld wordt verloren. Het is dus zeer belangrijk dat het model beter leert hoe het moet reageren in downtrends, vooraleer het echt ingezet kan worden.

Laten we even terugkomen op de onderzoeksvraag: “Wat is het meest geschikte AI model voor het voorspellen van de koers van cryptomunten aan de hand van open source data?”

Een LSTM netwerk lijkt hier duidelijk de beste optie. De open source data moet wel uitgebreid worden met indicator berekeningen, maar dit is vrij eenvoudig door Ta-Lib. Anderzijds, hoewel een LSTM netwerk hiervoor wel goed geschikt is, is labelled training dat niet. Unsupervised learning lijkt mij de volgende stap naar een beter resultaat.

Graag raad ik iedereen met wat interesse in trading en AI zeker aan om deze manier van trading te verkennen, of om algoritmisch te traden. Niet alleen voor de mogelijkse winst maar ook voor de kennis die je onderweg bijleert. Aangezien geld op een spaarrekening niets opbrengt, en erger nog, waarde verliest door inflatie, is het zeker interessant om nieuwe manieren van investeren te onderzoeken.

7 Literatuurlijst

[1] Pedro Lara-Ben´ıtez, Manuel Carranza-Garc´ıa, Jos´e C. Riquelme, 2021, An Experimental Review on Deep Learning Architectures for Time Series Forecasting. Available: https://arxiv.org/pdf/2103.12057.pdf

[2] Jingyi Shen & M. Omair Shafiq, 28 Aug, 2020, Short-term stock market price trend prediction using a comprehensive deep learning system. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6

[3] Armando Vieira, 29 Sep, 2019, Trading Through Reinforcement Learning using LSTM Neural Networks, Available: https://medium.com/@Lidinwise/trading-through-reinforcement-learning-using-lstm-neural-networks-6ffbb1f5e4a5

[4] Bruce Yang, 25 Aug, 2020, Deep Reinforcement Learning for Automated Stock Trading, Available: https://towardsdatascience.com/deep-reinforcement-learning-for-automated-stock-trading-f1dad0126a02

[5] Rian Dolphin, 21 Oct, 2020, LSTM Networks | A Detailed Explanation, Available: https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9

[6] Team Choice, 21 Dec, 2021, Best Combination of Technical Indicators for Intraday Trading, Available: https://choiceindia.com/blog/best-combination-of-technical-indicators-for-intraday-trading/

[7] Alberto Romero, A New AI Trend: Chinchilla (70B) Greatly Outperforms GPT-3 (175B) and Gopher (280B), Available: https://towardsdatascience.com/a-new-ai-trend-chinchilla-70b-greatly-outperforms-gpt-3-175b-and-gopher-280b-408b9b4510

[8] Edd Gent, DeepMind's New AI With a Memory Outperforms Algorithms 25 Times Its Size, Available: https://singularityhub.com/2021/12/20/biggers-not-always-better-deepminds-new-language-ai-is-small-but-mighty/

[9] TicTacTec LLC, Ta-lib, Available: https://www.ta-lib.org

[10] Daniel Johnson, Reinforcement Learning: What is, Algorithms, Types & Examples, Available: https://www.guru99.com/reinforcement-learning-tutorial.html

[11] Jean Folger, 4 March, 2021, Automated Trading Systems: The Pros and Cons, Available: https://www.investopedia.com/articles/trading/11/automated-trading-systems.asp

[12] Jason Fernando, Moving Average Convergence Divengence (MACD), Available: https://www.investopedia.com/terms/m/macd.asp

[13] Jason Fernando, Relative Strength Index (RSI), Available: https://www.investopedia.com/terms/r/rsi.asp

[14] Cory Mitchell, Money Flow Index – MFI Definition and Uses, Available: https://www.investopedia.com/terms/m/mfi.asp

[15] James Chen, Chaikin Oscillator, Available: https://www.investopedia.com/terms/c/chaikinoscillator.asp

[16] Adam Hayes, Bollinger Band®, Available: https://www.investopedia.com/terms/b/bollingerbands.asp

[17] Adam Hayes, Average True Range (ATR), Available: https://www.investopedia.com/terms/a/atr.asp

[18] Ben Dickson, 15 June, 2020, The cas for self-explainable AI, https://bdtechtalks.com/2020/06/15/self-explainable-artificial-intelligence/

[19] Ben Dickson, 15 Oct, 2018, Explainable AI: interpreting the neuron soup of deep learning, available: https://bdtechtalks.com/2018/10/15/kate-saenko-explainable-ai-deeplearning-rise/

[20] Ran Aroussi, Available: https://github.com/ranaroussi/quantstats

[21] Vaibhav Sahu, 29 June, 2018, Power of a Single Neuron, Available: https://towardsdatascience.com/power-of-a-single-neuron-perceptron-c418ba445095

[22] Algo.lt, 2022, Algorithmic Trading Portfiolio, Available: https://www.algo.lt/en/

[23] Tomiwa, 3 Nov, 2018, How My Machine Learning Trading Algorithm Outperformed the SP500 for 10 years, Available: https://towardsdatascience.com/the-austrian-quant-my-machine-learning-trading-algorithm-outperformed-the-sp500-for-10-years-bf7ee1d6a235

[24] Thomas Rochefort-Beaudoin, 15 Oct, 2019, Beating the S&P500 using machine Learning, Available: https://towardsdatascience.com/beating-the-s-p500-using-machine-learning-c5d2f5a19211

[25] Andrea Nalon, The rise of automated trading: Machines trading the S&P 500, Available: https://www.toptal.com/machine-learning/s-p-500-automated-trading

[26] Abhay Pawar, 10 March, 2021, Machine Learning models for 100% better returns in algo-trading, Available: https://medium.datadriveninvestor.com/machine-learning-models-for-market-beating-trading-strategies-c773ba46db66

8 Bijlages

8.1 Verslag Computer Crime unit

Dinsdag 11 January 2022.

Francis Nolf van de federale politie – computer crime unit kwam een presentatie geven over hoe ze te werk gaan en wat ze zoal doen om computer crime tegen te gaan. De CCU is ontstaan in 2001 na het samenvoegen van 2 aparte teams na de samenvoeging van alle politie departementen. De CCU is opgedeeld in 2 delen. De federale CCU en de regionale CCU. Deze hebben onder zich nog verschillende kleinere teams gespecialiseerd in bepaalde dingen.

Ik zal even verder uitleggen wat de verschillende teams doen.

Zo is er bijvoorbeeld een OSINT team (Open Source Intelligence) dat zich focust op het verzamelen en verwerken van publieke informatie dat beschikbaar is op het internet. Onder andere een bepaalde locatie onderzoeken via Google maps voordat er een interventie op deze locatie gebeurd zodat ze zich zo goed mogelijk kunnen voorbereiden hierop zonder ter plaatsen te moeten gaan rondkijken en zo mogelijks hun geplande interventie verklappen.

Het volgende team specialiseert zich in het afluisteren van telefoons en het decrypten van deze communicatie tussen 2 personen. Zo zal dit team ook proberen de locatie van onder andere drugs dealers te ontdekken gebaseerd op de communicatie patronen van de smartphones.

Er is ook een team dat onderzoek doet naar alles wat te maken heeft met data opslag op allerlei soorten apparaten. De meest voor de hand liggende zijn dan smartphones, computers, etc. Maar er zijn heel veel apparaten die iets minder voor de hand liggen dat ook vaak gebruikt worden zoals spelconsoles zoals een XBOX, routers, access points, etc. Elk apparaat dat data kan bevatten kan nuttig zijn in een onderzoek.

Het laatste en ook nieuwste team is verantwoordelijke voor Hacking. Dit team is nog maar recent in werking getreden na een verandering in de wet waardoor het nu toegestaan is dat de federale politie effectief hacking en exploiting kan gaan toepassen om informatie te verzamelen van verdachten. Dit wordt meestal gedaan met hacking specifieke besturingssystemen zoals Kali Linux. Dit besturingssysteem komt met een hele reeks aan programma's specifiek om te hacken.

Na een introductie van de CCU ging de presentatie verder in op de details van hoe onderzoeken gevoerd worden. Een van de voorbeelden bijvoorbeeld ging over de manier van omgaan met apparaten waarvan de waarde van groot belang is. Ze proberen geen schade aan te brengen aan het originele apparaat en gaan dus ook niet de originele harde schijf gebruiken maar nemen een schijf kopie die bit per bit een exacte kopie is van de originele schijf. Hier worden ook hashes van genomen om later te kunnen aantonen dat deze wel degelijk exact hetzelfde zijn en niet aangepast.

Verder ging het ook over de tools die ze gebruiken, zowel publieke en private tools, binnen de ccu voor de extractie van data van apparaten. Er zijn veel programma's reeds ingebouwd in Linux zoals DD, DCFLDD,.. om drives te kopiëren op byte-level. Er komen natuurlijk ook veel wetten kijken bij alles rond data verzameling en hoelang ze deze moeten bijhouden. Deze zijn zeer strikt en hier mag geen fout gemaakt worden. Er is ook een groot verschil tussen de wetten omtrent data verzameling en opslag over verschillende landen. In België zal de data na een onderzoek voor 6 maanden lang bijgehouden worden om later terug op te komen in verder onderzoek, maar in Duitsland is het verboden om data bij te houden als het onderzoek is afgerond.

Het is ook enkel toegestaan deze data op te vragen als er toestemming wordt gegeven door een procureur of magistraat. Onder geen enkele andere voorwaarde mag de data opgevraagd worden.

Een sysadmin van een bedrijf kan ook als medeplichtige beschouwd worden als deze weigert mee te werken met de politie in een onderzoek waarbij er mogelijks data nodig is van bedrijfsservers.

Naast de Belgische wetten zijn er ook internationale wetten waardoor het opvragen van data over buitenlandse verdachten verplicht via het ministerie van buitenlandse zaken moet opgevraagd worden en dit kan heel lang duren van weken tot jaren. Er zijn bepaalde uitzonderingen zoals facebook, outlook,…

deze bedrijven bieden rechtstreeks contact met interne teams voor het opvragen van informatie. Het is nog steeds nodig dat dit goedgekeurd wordt door een procureur of magistraat.

Na de uitleg over het gebruik en opvragen van data was er ook een belangrijk deel over het in beslag nemen van fysieke apparaten. Het is enkel voor de politie toegestaan om apparaten in beslag te nemen en zelfs binnen de politie enkel degene met een hogere rang. Na het in beslag nemen van een apparaat proberen ze alle signalen te blokkeren om het apparaat in een oorspronkelijke staat te bewaren. Dit is moeilijk omdat veel apparaten altijd een soort van verbinding hebben zoals 4G, bluetooth, cell service, etc. en om hier tegen in te gaan gebruiken ze speciale zakken die werken als een kooi van Faraday. Dit is omdat bewijs dat gevonden wordt anders ongeldig beschouwd kan worden als het apparaat verbonden geraakt met het internet omdat er dan mogelijks aanpassingen aan zijn gebeurd.

Voor bepaalde apps zoals messenger en whatsapp moet er weer toegang verleend worden door een magistraat en enkel als de verdachte toestemming geeft hiervoor.

Ook het afluisteren van apparaten moet toegestaan worden door een magistraat en elk afgeluisterd bericht moet een rapport van op papier gezet worden en naar de magistraat gestuurd worden elke 5 dagen. Daarbovenop kan het soms gebeuren dat een advocaat vraagt om elk gesprek (niet alleen die dat als relevant worden beschouwd voor het onderzoek) uit te typen.

Het laatste deel van de sessie ging over de bedreigingen die momenteel populair zijn en dan gaat het bijvoorbeeld over ransomware, hierbij wordt je data op je apparaat geëncrypteerd en moet je betalen om deze terug te laten decrypten. Recent worden er ook heel veel scams gedaan, in de meeste gevallen doen de scammers zich voor als iemand anders of een bedrijf.

Deze zaken zijn moeilijk op te lossen omdat deze scammers meestal uit andere landen komen wat het onderzoek moeilijk maakt door de vele regels opgelegd door de wet.

Bij deze scams horen vaak ‘money mules’, ze proberen je te doen geloven dat je geld kan verdienen zonder echt te moeten werken, dit is een tussenpersoon waarnaar het geld initieel verstuurd word, deze houden dan ook een deel van het geld en storten de rest weer verder door naar een andere rekening om het moeilijker te maken om te traceren naar wie het geld eigenlijk gaat.

Het gevaar hierbij is dus dat deze money mules, dat vaak eigenlijk ook mensen zijn die slachtoffer zijn van een scam want ze weten vaak niet welk geld ze eigenlijk doorstorten, ook medeplichtig beschouwd worden voor de wet en dus ook een straf kunnen krijgen.

Tot slot werden er nog een paar problemen besproken die het moeilijk maken om deze misdaden op te lossen. Onderzoekers mogen bijvoorbeeld geen acties van een verdachte uitlokken omdat dit ongeldig wordt beschouwd in een rechtszaak. Een 2de probleem is dat peer-to-peer netwerken niet onderzocht mogen worden en dus het delen van bestanden en dergelijke niet onderzocht kan worden. Ook het toestaan van legal hacking, dus waarbij de onderzoekers zelf de verdachte gaan proberen hacken om te onderzoeken, zou het onderzoek helpen.

De presentatie van Francis Nolf was zeer informatief en leuk gebracht en zo zien we eens hoe cyber crime wordt aangepakt in België.

8.2 Handleiding Researchproject

8.2.1 Software

Python

Alvorens iets te doen, is het belangrijk ervoor te zorgen dat python geïnstalleerd is, python 3.9.x wordt ten zeerste aanbevolen. U kunt python installeren vanaf de volgende url:

https://www.python.org/downloads/

Eens python geïnstalleerd is, kan je de installatie dubbel controleren door een terminal te openen en het volgende commando in te voeren.

python -V

dit zou dan de geinstaleerde versie moeten weergeven.

Cuda

Als u een Nvidia grafische kaart heeft, is het mogelijk om het model hiermee te runnen. Om dit te doen is een stukje software nodig genaamd CUDA. CUDA laat Nvidia gpu's toe om hun parallelle rekenkracht te gebruiken voor het trainen en gebruiken van AI modellen.

Om CUDA te installeren, kun je de officiële Nvidia gids

https://docs.nvidia.com/cuda/ cudainstallation-guide-microsoft-windows/index.html

volgen. Om het kort samen te vatten, heb je twee delen nodig om CUDA te installeren, het eerste is de installatie software die de drivers zal installeren die nodig zijn om CUDA te gebruiken, en de tweede tegenhanger is de cuDNN Library.

Om cuDNN te downloaden heb je een Nvidia account nodig ter verificatie, de gids vertelt je hoe en waar alles te installeren. Zodra CUDA is geïnstalleerd kun je de installatie testen met het volgende commando.

nvcc --version

Je uitvoer zou er ongeveer zo uit moeten zien als de installatie succesvol was:

nvcc : NVIDIA (R) Cuda compiler driver
Copyright (c) 2005–2021 NVIDIA Corporation
Built on Thu_Nov_18_09:45:30_PST_2021
Cuda compilation tools , release 11.5 , V11.5.119
Build cuda_11 . 5 . r11 .5/ compiler .30672275_0

8.2.2 Data Collector

Het opzetten van deze repository is vrij eenvoudig, je kunt ofwel de repo clonen of downloaden als een zip.

git clone https://github.com/Research−Project−Crypto/ DataCollector.git

u heeft een binance api key en secret nodig om te beginnen met het verzamelen van crypto prijs en volume data, maar dit kan gratis verkregen worden door een Binance account aan te maken en naar de Binance API management pagina te gaan.

https://www.binance.com/en/my/settings/api-management

Hier dient u een API-sleutel aan te maken en deze in het crypto.py bestand te plaatsen.

Het verzamelen van prijsgegevens kan lang duren, afhankelijk van hoeveel gegevens u wilt, om te voorkomen dat u IP address een api ban krijt is er een timer tussen elke api call. Je kunt het programma altijd stoppen als je denkt dat je genoeg data hebt.

8.2.3 Ticker Timescale swap

De ticker timescale-swap toepassing wordt gebruikt om de tijdschaal van candles te veranderen.

U zult bijvoorbeeld waarschijnlijk de meest nauwkeurige gegevens die u hebt (1min candles) willen converteren naar iets minder specifieks dan dat (5min / 10min / etc…).

Repository

Om de repository te clonen dien je volgend commando uit te voeren:

git clone https://github.com/Research−Project−Crypto/TickerTimescaleSwap.git

−−recursive

Als je vergeten bent de repository te klonen met het recursieve argument gebruik je het volgende commando na klonen:

git submodule update −−i n i t −−recursive

Build Dependencies

De volgende commandos zijn voor arch-based linux systemen, alle nodige libraries zijn ook beschikbaar voor debian en andere linux systemen maar dit zal me een ander commando moeten geinstalleerd worden.

pacman −S −−noconfirm −−needed gcc make premake

Compileren

Genereer build instructies met volgend commando:

premake5 gmake2

Compileer het progamma met volgend commando

make config=release

Timescale swap

Standaard is de tijdschaal 1:60, momenteel kun je dit niet dynamisch definiëren via argumenten bij het aanroepen van het programma, dus je zult de code direct moeten bewerken en hercompileren. Om de verhouding te veranderen ga naar het bestand

src/main.cpp

en zoek waar de

processor.start()

wordt aangeroepen. Als je een getal opgeeft in de start methode kun je de tijdschaal instellen.

Argumenten

Positie	Argument
1	Data Input Folder
2	Data Output Folder

Voorbeeld:

TickerTimescaleSwap data/ input_folder data/output_folder

Data Integriteit Verifieren

Bij dit project is een python script inbegrepen waarmee je de binaire uitvoer kunt verifiëren.

python3 scripts/binary_reader.py

Het zal langzaam over alle cellen lopen, dit vooral om kort na te gaan of er rekenfouten in het programma zitten, je kan zelf manueel kijken of de output correcte waardes zijn.

8.2.4 Data Preprocessor

Klonen

Om de repository te clonen dien je volgend commando uit te voeren:

git clone https :// github . com/Research-Project-Crypto/DataPreprocessor . git

--recursive

Als je vergeten bent de repository te klonen met het recursieve argument gebruik je het volgende commando na klonen:

git submodule update --i n i t --recursive

Build Dependencies

De volgende commandos zijn voor arch-based linux systemen, alle nodige libraries zijn ook beschikbaar voor debian en andere linux systemen maar dit zal me een ander commando moeten geinstalleerd worden.

pacman -S --noconfirm --needed gcc make premake

yay -S talib

paru kan ook gebruikt worden in plaats van yay, afhankelijk van welke geinstalleerd is op je systeem.

Compileren

Genereer build instructies met volgend commando:

premake5 gmake2

Compileer het progamma met volgend commando

make config=release

Argumenten

Positie	Argument
1	Data Input Folder
2	Data Output Folder

Voorbeeld:

DataProcessor data/ input_folder data/output_folder

Nadeel van alleen argumenten. u kunt alleen CSV-tekstgegevens parsen omdat deze setting in de settings.json staat. Indien u helemaal geen argumenten meegeeft zullen alle argumenten uit settings.json gebruikt worden.

Gebruik alleen de optie is_binary als u onze timescale swap gebruikt hebt, dit zal de bestanden proberen inlezen als binary files in plaats van csv.

De output is altijd een binary file om schijfruimte te besparen en snelheid te verbeteren.

Data Integriteit Verifieren

Bij dit project is een python script inbegrepen waarmee je de binaire uitvoer kunt verifiëren.

python3 scripts/binary_reader.py

Het zal langzaam over alle cellen lopen, dit vooral om kort na te gaan of er rekenfouten in het programma zitten, je kan zelf manueel kijken of de output correcte waardes zijn.

8.2.5 Model Testing

Local Environment Opzetten

Omdat bepaalde pip packages niet werken met de nieuwste versie van Python op het moment van schrijven waren we verplicht om een oudere versie van Python te gebruiken. Om deze reden wordt gebruik van een venv aanbevolen omdat het gemakkelijker is om de gebruikte Python versie af te dwingen indien er meerdere versies geinstalleerd zouden zijn op u systeem.

Gebruikt volgende commandos om en venv op te zetten in de huidige folder:

python3 .9 –m venv . venv

source . venv/bin/ activate

installeer dan alle nodige packages met volgend commando:

pip install –r requirements.txt