LSTM-Model Forecasting af Finansielle Vækstaktier

AALBORG UNIVERSITET

Det Samfundsvidenskabelige Fakultet

Aalborg Universitet

Danmark

Titel:

LSTM-Model Forecasting af Finansielle Vækstaktier

Projekt:

Speciale

Forfatter:

Jimmi Lauritzen - 20134437

jnla13 @student.aau.dk

Vejleder:

Joachim Hagbart Madsen

Anslag: 102.828

Afleveringsdato:

10-01-2022

Abstract

Economist have been trying to forecast financial assets with various methods and different results where the main goal is to achieve the highest possible profit. This thesis examines the LSTM (Long Short Term Memory) model by comparing it to the traditional linear model, ARIMA. The goal is to examine which is superior for predicting the future stock price of three different growth stocks in different sectors using a one-step ahead forecasting method. The data was collected for 10 years from 2010 to 2020, where the first 8 years was used to train the models, followed by data for 1 year of validation and 1 year of forecasting. Two different architypes of LSTM were tested a 5-day and 10-day sequence length where the 10 sequence LSTM model seemed to outperform the 5-day model. To properly evaluate the LSTM model an ARIMA model for each stock was used as a baseline model. Both models was created as a univariate model where the LSTM model seemed to be more accurate compared to the ARIMA model. The LSTM models achieved a lower RMSE and MAPE and a diebold-marino test was performed and it concluded that the LSTM had a lower forecasting error. Although the overall results suggested that the LSTM outperformed the ARIMA model meaning that the non-linear models would provide better forecasts on predicting the future price of financial assets. It is however still clear that both models would need further improvements before being considered feasible for real life applications.

Indholdsfortegnelse

ABSTRACT ..... **III**

1 INDLEDNING **6**

1.1 PROBLEMFORMULERING 8
1.2 LITTERATUR REVIEW 8
1.3 AFGRÆNSNING 9
1.4 BIDRAG 10
1.5 AFHANDLINGENS STRUKTUR 10

2 TEORI **12**

2.1 AKTIEMARKEDET 12
2.2 DEN EFFICIENTE MARKEDSHYPOTESE 12

3 LINEÆRE MODELLER **15**

3.1 LINEÆRE MODELLER 15
3.1.1 Stationaritet *16*
3.2 ARIMA 16
3.2.1 Unit Root *17*
3.2.2 Akaike Information Criterion *18*
3.2.3 Augmented Dickey-Fuller Test *18*

4 IKKE-LINEÆRE MODELLER **20**

4.1 ARTIFICIAL INTELLIGIENCE 20
4.2 MACHINE LEARNING 21
4.3 ARTIFICIAL NEURAL NETWORK 22
4.3.1 Aktiveringsfunktioner *23*
4.3.2 Træning af ANN *26*
4.3.3 Stokastisk gradient & Back-propagation *27*
4.4 RECURRENT NEURAL NETWORK 29
4.5 LONG SHORT TERM MEMORY 32
4.5.1 Forget Gate *34*
4.5.2 Input Gate *35*
4.5.3 Output Gate *37*
4.6 HYPERPARAMETERS 37
4.7 OPSUMMERING 38

5 EVALUERINGSMULIGHEDER **40**

5.1 ROOT MEAN SQUARED ERROR 40
5.2 MEAN ABSOLUTE PERCENTAGE ERROR 40
5.3 PESARAN TIMMERMANN 41
5.4 DIEBOLD-MARINO 42

6 METODOLOGI OG DATA **44**

6.1 DATAGRUNDLAG 44
6.2 VÆKSTAKTIER OG P/E RATIO 45
6.3 DESKRIPTIV STATISTIK 48
6.4 ROLLING WINDOW 50

7 MODELUDVÆLGELSE 51

7.1 ARIMA 51
7.2 LSTM MODELKONSTRUKTION 52

8 EMPIRISKE RESULTATER 56

9 DISKUSSION 61

9.1 KRITIK AF LSTM 61
9.2 DATA PROBLEMET 62
9.3 SIGNAL-TO-NOICE 63
9.4 TILFØJELSE AF VARIABLER 63

10 KONKLUSION 65

11 LITTERATURLISTE 66

12 APPENDIKS 71

12.1 BILAG A: 71
12.2 BILAG B: 72

1 Indledning

I den finansielle verden er et af de mest veldiskuterede emner vedrørende profit, trends, risiko og maksimering, hvor det fra investorernes side er målet at opnå en profit ved bl.a. at kunne forudsige aktiekursen, hvilket er blevet forsøgt med yderst varierende succes. Det kan ikke anses som værende en nem eller simpel opgave at prædiktere aktiekursen, hvilket yderligere besværliggøres, når der findes, næsten uendelige metoder, at gøre dette på. Derudover kan der kan være forskel i investorernes præferencer i form af ønsket afkast og risikovillighed. Disciplinen at forudsige aktiekursen er omdiskuteret og forsøgt utallige gange, så er der fortsat bred uenighed om den mest efficiente metode, hvor der tidligere har været fokus på anvendelsen af økonometriske og statistiske metoder baseret på historiske data (Atsalakis & Valavanis, 2013).

Den stigende mængde af information som er tilgængelig både i form af hyppigere data men ligeledes med henblik på markedsrelaterede information, hvilket kan bidrage til skabelsen af nye investeringsstrategier i form af eventuelle porteføljesammensætninger eller strategier, denne ekstra kompleksibilitet, men hvorvidt effekten af nyhedsspredningen kan fungere som et værktøj er blevet debatteret i litteraturen af blandet andet Malkiel i 2003 som udtaler sig følgende:

> ” *The accepted view was that when information arises, the news spreads very quickly and is incorporated into the prices of securities without delay. Thus, neither techincal analysis, which is the study of past stock prices in an attempt to predict future prices, nor even fundamental analysis, which is the analysis of financial information such as company earnings and asset values to help investoes select* ”*undervalued*” *stocks, would enable an investor to achieve returns greater than those that could be obtained by holding a randomly selected portfolio of individual stocks, at least not with comparable risk."* (Malkiel, 2003)”

Så Malkiel beskriver markedet som værende et højest efficient marked, hvor priser og nyheder bliver øjeblikkeligt inkorporeret i markedet, hvilket altså umuliggør teknisk analyse og det at opnå et afkast højere end det marked præsterer og dette umuliggøre det, at prædiktere aktiekursen og man kan ikke, ifølge ham, finde et finansielt aktiv der er undervurderet, hvilket betyder at det kort sagt er nytteløs og umuligt at skabe et afkast over markedsafkastet. Denne tankegang er blevet diskuteret i litteraturen og tiltroen dertil er også formindsket, hvilket skyldes at økonomer har bevist at det til en vis grad er

muligt at prædiktere aktiekursen gennem forskellige mønstre i prisudviklingen for aktiverne (W. Lo & A. MacKinlay, 1999).

Igennem de senere år hvor udviklingen af computerteknologi samt den tidligere benævnte øgede mængde af data og nyhedsinformation fremmet brugen af kunstig intelligens eller AI (Artificial Intelligience) hvor en af metoderne er ML (Machine Learning). AI kan yderligere beskrives som computerteknologi der medfører at computeren kan foretage og udføre menneskelige opgaver, og i nogle tilfælde har evner, der bevæger sig udover det, der er menneskeligt muligt, det er dog vigtigt at pointere, at machine learning ikke skal forstås som ”man vs machine” men rettere ”man plus machine”, altså mand og maskine der arbejder sammen og ikke maskinen eller rettere teknologien der skal erstatte det menneskelige aspekt. Flere investorer og banker har allerede adopteret en machine learning tilgang til investering, hvor machine learning bliver benyttet som et redskab til at finde porteføljesammensætninger og til at foretage hurtige prædikteringer af aktiekursen, hvor der bliver handlet på baggrund af disse modeller (CNN, 2019).
Machine learnings evner til at prædiktere på komplekst data har skabt en konkurrence mod de gamle eksisterende lineære modeller såsom den anerkendte lineære økonometriske ARIMA model, hvilket også benævnes af Malkiel i 2003 hvor han pointerer at i særdeleshed at kortvarige effekter kan blive prædikteret, hvilket skyldes sofistikerede empiriske metoder såsom ML, men ydermere en øget tilgængelighed i form af databaser, hvilket selvfølgelig medvirker til modsigelsen af den tidligere benævnte efficiente markedshypotese (Malkiel, 2003).

Afhandlingen tager afsæt i vækst-aktier da disse har potentialet til at medføre en højere fremtidig indtjening i fremtiden og derfor kan gøre dem mere interessant at prædiktere på, hvor målet selvfølgelig er at finde den nye Amazon eller Tesla aktie før den tager fart. Dette munder ud i afhandlingens interesse i at forsøge at lave en komparativ analyse af ARIMA og LSTM modellerne og se hvorvidt de kan være et nyttigt redskab til at prædiktere vækstaktier og bidrage som et værktøj til udvælgelsen af finansielle aktiver og således opstår årsagen til efterfølgende problemformulering.

1.1 Problemformulering

Hvor anvendelig er lineære og ikke-lineære modeller til at prædiktere udviklingen af udvalgte vækst-aktier

1.2 Litteratur Review

I dette afsnit vil der blive foretaget en redegørelse for eksisterende litteratur der omhandler aktieforudsigelse, hvor der er anvendt ARIMA modeller eller ML modeller eller en kombination deraf.

Eftersom der ikke eksisterer en konsensus om den optimale fremgangsmåde at prædiktere aktiekursen og opsætte porteføljer er der sket meget forskning på området. Især i det nye århundrede er der foretaget meget forskning, hvor blandt andet ARIMA (AutoRegressive Integrated Moving Average) og LSTM (Long Short Term Memory) modeller er involveret. Et eksempel på dette er Ayodele et al. i deres artikel ”Stock Price Prediction using ARIMA Model” der benytter data fra NYSE (New York Stock Exchange) og Nigeria Stock Exchange (NSE) hvor konklusionen resulterede i at deres ARIMA model havde potentiale til short-term forudsigelser, altså modellen kunne prædiktere på kort sigt og de mener at modellen var jævnbyrdig med andre modeller, der prøver at forudsige aktiekursen (Ayodele et al, 2014).

Afeef et al. fandt i 2018 i deres artikel ”Forecasting Stock Prices through Univariate ARIMA Modeling”, hvor de anvendte data fra Oil & Gas Development Company Limited (OGDCL) som er et af de største firmaer i Pakistan. Dataet var daily closing price i en periode fra 2004 til 2018 med næsten 3632 observationer. Resultatet af studiet viser, at nogle af ARIMA modellerne, heriblandt ARIMA(1,1,1) påviste en stærk evne til at prædiktere på kort sigt (Afeef et al, 2018).

Namin et al. finder i deres komparative studie i artiklen ”A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM and BiLSTM”, at til trods for LSTM modellerne generelt outperformer de lineære modeller, så har valget af dataet afgørende effekt for modellernes evne til at forudsige aktiekursen. Der bliver i artiklen undersøgt både daglige, ugentlige og ikke mindst månedlige observationer hvor resultatet viste at ARIMA modellen klarede sig markant bedre på månedlige observationer kontra ARIMA modellen med daglige observationer. (Namin et al, 2019) Deorukhkar et al. forsøgte sig i 2019 at kombinere flere modelleringsmetoder i én. Formålet med deres model var at prøve at prædiktere aktiekursen den efterfølgende dag. Dataet anvendt i deres

forskning var afslutningsprisen den efterfølgende dag fra individuelle aktie fra NSE (National Stock Exchange). De formåede at kombinere LSTM, ARIMA og Sentiment Analysis, hvor modellen startede med at bruge historiske intuitioner fra ARIMA og LSTM-modellerne og kombinerede det med holdningen til den nuværende aktiemarkedssituation. Resultatet de opnåede var interessante i den forstand, at hybridmodellen formåede at slå alle de individuelle modeller, som også prøver at prædiktere aktiekursen for den efterfølgende dag (Deorukhar et al, 2019).

Qihang Ma foretog i 2020 i artiklen ”Comparison of ARIMA, ANN and LSTM for Stock Price Prediction” en komparativ analyse af de tre ovennævnte modeller hvor den lineære ARIMA model skulle konkurrere med de ikke-linære ANN og LSTM modellerne. Resultatet af studiet viste af LSTM modellen havde den bedste prædikteringsevne og udkonkurrerer både ARIMA og ANN-modellen. Der diskuteres i artiklen at fluktueringer af aktiepriserne ikke blot skyldes ændringer i tid og pris, men også socio-politiske og andre økonomiske faktorer, som ikke indgår direkte (Ma, 2020).

Sidra et Al. forsøgte i 2020 også med en ML-tilgang til det at forudsige aktiekursen, dataet omhandlende daglige prisændringer fra 2014-2020 og selve dataet var prisændringer fra NIFTY 50 indeks, som indeholder 50 vægtede gennemsnit af firmaer fra Indien. Deres bedste model opnår en RMSE på 0.03 hvorved konklusionen fremstår som, at LSTM var deres bedste model og outperformedede de andre modeller (Sidra et al, 2020).

1.3 Afgrænsning

Specialets primære mål er at se hvorvidt den nyere ikke-lineære tilgang i form af ML kan udkonkurrere den ældre økonometriske ARIMA-model. Eftersom der eksisterer mange former for ML i form af forskellige konstruerede netværk, så vil dette speciale afgrænse sig LSTM som den anvendte ML-metode og ARIMA-modellen som den udvalgte lineære model som benyttes som argumentations- og evalueringsgrundlag.

Der eksisterer en stor mængde af faktorer, der kan påvirke aktieprisen, men for ikke at overkomplicere de opstillede modeller og undgå potentielt overfittede modeller så afgrænses der, for brugen af dividende samt transaktionsomkostninger såsom mæglerhonorarer, handelsgebyrer mm. Eftersom der primært undersøges hvilken model der er fordelagtig til udvælgelsen af de udvalgte finansielle aktiver, så kigges der på individuelle vækstaktier med en ”Buy and Hold” strategi, hvor der udvælges

én eller flere aktier og disse holdes i en længere periode, hvilket også er inspireret af Warren Buffet tilgang til investering hvor han selv har udtalt af hans fortrukne periode at holde en aktie er for evigt (Graham, 2003). Slutteligt antages der for at undgå overkomplicerede og på baggrund af Buy and Hold strategien ingen shorting af de finansielle aktiver.

1.4 Bidrag

Selvom der, som der også er vist i afsnittet Litteratur Review, så er anvendelsen af lineære og ikke-lineære modeller ikke som sådan noget nyt, så hensigten med specialet er ikke at opfinde den dybe tallerken i form af nyt og banebrydende teori, men derimod tilføje noget til den eksisterende litteratur, specialet prøver at adskille sig fra tidligere projekter og litteratur i form af brugen af individuelle finansielle aktiver fra SP 500, hvor udvælgelsen af disse aktier kun sker i form af anvendelsen af deres P/E value, hvilket medfører at afhandlingen specifikt forsøger at undersøge LSTMs prædikteringsevner sammenlignet med ARIMA på vækstaktier. Det kan hertil tilknyttes, at da der findes tæt på uendelige mængder af fremgangsmåder at konstruere LSTM modeller på, så her vil der også være variation i selve modelkonstruktionen, hvor der i afhandlingen benyttes en mere avanceret form for optimering af LSTM-modellerne sammenlignet med meget andet eksisterende litteratur, hvilket sker i form af anvendelsen af en algoritme til at optimere modellens hyperparameter baseret på værdisæt bestemt af modelkonstruktøren, hvilket må formodes vil medføre nye resultater, som kan tilføjes til den eksisterende litteratur.

1.5 Afhandlingens struktur

Afhandlingen består af 10 overordnede afsnit, hvor der hidtil har været introduceret specialets formål og bidrag til den eksisterende litteratur.

Afsnit 1 – Giver en introduktion af emnet samt afhandlingens formål samt en afgræsning af emnet og bidrag til den eksisterende litteratur.

Afsnit 2 – Beskriver aktiemarked teori, for at give læseren en forståelse af, hvad aktiemarkedet er samt hvorfor aktiemarkedet er vigtigt, hvorefter der præsenteres den dertilhørende hyppigt debatterede aktiemarkedsteori den efficiente markedshypotese.

Afsnit 3 – Præsenterer det teoretiske fundament bag den traditionelle lineære ARIMA-model som har til hensigt af fungere som en baseline, altså en basis model, der kan benyttes som et vurderingskriterie for hvordan de opstillede LSTM-modeller klarer sig ift. at forecaste aktiekursudviklingen af vækstaktierne.

Afsnit 4 – Her præsenteres det teoretiske fundament til de ikke-lineære modeller, hvor der kort redegøres for, hvad Artificial Intelligence, altså kunstig intelligens er for en støbning, hvorefter der vil blive forklaret mere i dybden hvor ML bliver beskrevet samt nogle relevante modeller som giver indblik i LSTM modellens struktur.

Afsnit 5 – Redegøre for de evalueringsmuligheder der vil blive anvendt i afhandlingen, for at kunne sammenligne de lineære og ikke-lineære modeller.

Afsnit 6 – Fremsætter den metodologiske fremgangsmåde anvendt i specialet heriblandt det datamæssige grundlag, de udvalgte finansielle aktiver samt en dertilhørende deskriptiv statistik.

Afsnit 7 – Præsenterer udvælgelsen og begrundelsen deraf for de lineære og ikke-lineære modeller samt den endelige LSTM-konstruktionen.

Afsnit 8 - Præsenterer de empiriske resultater som specialet har opnået baseret på det udvalgte data samt metodologiske fremgangsmåde.

Afsnit 9 – Forsøger at fremlægge og diskutere andre metodologiske fremgangsmåder, der kunne have forbedret projektet, samt nogle af de problematikker der ligger til grund for brugen af neurale netværk.

Afsnit 10 - Præsenterer konklusionen som afrunder specialets resultater og konkluderer på de forudgående afsnit

2 Teori

I dette afsnit vil det teoretiske fundament blive præsenteret. Først vil der komme en introduktion til aktiemarkedet og teori dertil, deriblandt den efficiente markedshypotese, hvorefter en gennemgang af de lineære modeller gennemgås og slutteligt de ikke-lineære modeller.

2.1 Aktiemarkedet

Aktiemarkedet kan defineres som et sofistikeret sted at foretage handler, hvor den handlede vare er aktier. Aktiemarkedet bliver tit forbundet med økonomien som helhed, da markedet både kan fungere som et redskab for private forbrugere til at opnå en profit, som bidrager til deres privatøkonomi, hvilket medfører, at hvis aktiemarkedet stiger drastisk, så vil husholdningernes forbrug af varer også kunne tænkes at stige i takt med markedet, da deres forbrugsevne bliver forbedret, dette blev især argumenteret af Gurley and Shaw i 1967 der mener et øget privatforbrug samt et realøkonomisk velfærd i helhed kan forbindes med en finansiel udvikling. Virksomhederne kan benytte aktiemarkedet til at opnå kapital, som virksomhederne derefter kan benytte til udvidelser eller andet, hvor mængden af kapital som virksomhederne kan finde, afhænger af situationen på aktiemarkedet, hvor de vil kunne finde en større kapital, hvis aktiemarkedet blomstrer. Aktiemarkedets situationen kan anskues som en indikator på økonomisk vækst, da der vurderes at være en positiv korrelation mellem aktiemarkedet og økonomisk vækst (Najeb, 2013). Det er tydeligt for de fleste, at aktiemarkedet kan fluktuere, men hvordan og hvornår er altid hyppigt diskuteret i litteraturen, hvor en af de mest anerkendte beskrivelser af aktiemarkedet er baseret på ECM (Efficient Market Hypothesis), altså den efficiente markedshypotese.

2.2 Den Efficiente Markedshypotese

Idéen vedrørende den efficiente markedshypotese er ikke ny og har eksisteret siden Fama French i 1970 fremsatte teorien om den efficiente markedshypotese. Fama påviste en teori der belyste, at markedet var højest efficient og fulgte en random walk, han beskriver i sin teori, at når al tilgængelig information i tidspunktet t er inkorporeret, som sker øjeblikkelig, så vil informationen være ugyldig til at forcaste fremtidige værdier af aktiekursen, hvilket betyder at prisændringer sker igennem ny information, hvor informationen kan beskrives som en stokastisk proces hvilket derfor må betyde, at fremtidige afkast er drevet af uforudsigelighed, hvilket Fama beskriver vha. følgende ligning:

$$E\left(z_{j,t+1} \middle| \Phi_{\mathrm{t}}\right) = 0 \tag{1}$$

Hvilket betyder, at det forventede afkast af z på aktien j til tiden t+1, altså den fremtidige værdi af aktien, på basis af nyhedsinformationen, Φi tiden t, er lig 0 og dermed beskrives som ’fair game’ (Fama, 1970).

Ovenstående bliver yderligere understøttet af Malkiel i 1973 som også argumenterede for, at informationen bliver øjeblikkeligt absorberet af markedet og dermed øjeblikkelig inkorporeret i prisen.

Malkiel argumenterer yderligere i 2016, at eftersom nyheder bliver øjeblikkeligt absorberet i markedet, så efterlades der ikke flere nyheder som kan benyttes til at prædiktere aktiekursen, hvilket umuliggør det at prædiktere prisen baseret på spredningen af nyhedsinformation.

Malkiel understreger at selvom kriser opstår, såsom husholdningskrisen i 2007-2008, som kunne anses som en begrundelse, for at teorien ikke holder, så argumenteres der for, at netop disse episoder bekræfter teorien. Det virker måske irrationelt, at der opstår disse gnidninger i form af husholdningsboblen, men at markedet så efterfølgende selv korrigerer beviser, at det er efficient. På kort sigt kan der dog kan argumenteres for, at aktiekursen må følge en random walk da information på dagsbasis kan anses som værende tilfældig i sin natur, hvilket dermed gør det umuligt at vide hvilke informationer der bliver bragt om aktierne dagen efter, og dermed ikke noget man kan opbygge en model og prædiktere på (Enders. 2015).

En random walk kan defineres som et matematisk udtryk for en stokastisk trend. Det kan beskrives på ligningsform som:

$$Y_t = Y_{t-1} + \varepsilon_t \tag{2}$$

Hvor Y_t på aktiemarkedet ville beskrive den nuværende aktiepris, hvor Y_{t-1} beskriver prisen for forrige periode, hvor det på dagsbasis ville være prisen i går, hvor der slutteligt er et støj komponent, som er beskrevet som ε_t. Ud fra dette kan vi altså konkludere, at såfremt aktieprisen følger en random walk, så vil ændringer i den nuværende pris for aktien kun være afhængig af støj-komponenten, som altså medfører denne tilfældige ændring i prisen, hvor prisen formodes at være konstant, medmindre der forekommer disse tilfældige ændringer (Enders 2015).

Malkiel nævner selv i 2016 at aktiemarkedet ikke kan beskrives som en ren random walk og kritiserer selv teorien i 2003, hvor han nævner at markedet kan forekomme inefficient og nyheder, især på kort sigt, kan medføre en hype omkring en aktie, hvilket kan bidrage til en momentumeffekt, som får aktierne til hurtigt at stige i værdi, eller tilsvarende ved negative nyheder, at falde i værdi. Derudover kan der også argumenteres for hvorvidt markedet er i stand til at agere i en passende grad på nyhedsinformationen, hvor det kan forekomme, at markedet ikke reagerer kraftigt nok på nyhederne og skaber disse anomalier i aktieprisen og derfor kan det diskuteres hvorvidt aktiemarkedet reelt er et efficient marked (Malkiel 2003).

3 Lineære Modeller

Dette afsnit tager afsæt i de lineære modeller og vil beskrive det teoretiske grundlag samt modeller og tests der er nødvendigt at anvende i tidsserie økonometri.

3.1 Lineære modeller

Specialet benytter sig af tidsserie regression til forcasting af prisudviklingen af aktiekursen for de udvalgte aktiver. Tidsserie analyse anvendes til at udtrække meningsfulde statistiske sammenhænge i data, hvor tid betegnes som den underliggende faktor. Formålet med tidsserie regression er at prædiktere de fremtidige værdier, i dette tilfælde af aktiekursen, baseret på tidligere observerede værdier af aktiekursen. For at forudsige fremtidige værdier, baseret på tidligere observeret værdier. Tidsserie økonometri tager højde for den interne struktur deriblandt autokorrelation og trends, der kan være præget af sæson, hvor, såfremt man vil prædiktere tidsserie data, så er det en nødvendighed at datasættet besidder den tidsinvariante egenskab at være stationær. Hvis en tidsserie skal være stationær, kræves det at den besidder nogle karakteristika såsom at tidsseriens middelværdi, ubetinget varians og autokovarians skal være uafhængig af tid. Det er fordelagtigt at der ikke forekommer tilfældigheder i det anvendte data, især i stokastiske tidsserier skyldes disse tilfældigheder en white noice process, som kan beskrives som en samling af ukorrelerede variabler ε_t der har en middelværdi lig 0 samt en begrænset varians $\sigma^2\varepsilon$. Det antages at denne samling af ukorrelerede variabler er uafhængige og kan udtrykkes matematisk på følgende måde:

$$E(\varepsilon_t) = 0 \tag{3}$$

$$var(\varepsilon_t) = \sigma_\varepsilon^2 < \infty \tag{4}$$

Og white noice processen som nævnt tidligere kan beskrives som følgende, såfremt variablerne er normalfordelt.

$$\varepsilon_t \sim N(0, \sigma_t^2) \tag{5}$$

Såfremt der anvendes data og testes for stationaritet, hvor konklusionen lyder, at dataet ikke kan karakteriseres, som værende stationær, grundet at datasættet eventuelt kunne være sæsonpræget eller have en trend, hvilket vil medfører at der skabes en model hvor forecastest højest sandsynligt ville forringes og ugyldig da man går imod en af de basale antagelser for ARIMA modellen, men dette kan

imødekommes ved at differentiere datasættet og opnå stationaritet, så selv et ikke-stationært datasæt kan blive anvendeligt til forecasting. (Enders, 2015)

3.1.1 Stationaritet

Stationaritet kan blive defineret som en stationær proces hvor gennemsnittet, varians samt autokorrelation ikke afhænger af tiden t. Såfremt dataet kan karakteriseres som stationær må der derfor ikke forekomme trends eller sæsonprægede fluktueringer derudover skal variansen og kovariansen være konstant. En stationær tidsserie kan yderligere defineres matematisk på følgende måde:

y_t er stationær hvis følgende er gældende $E(y_t^2) < \infty$ samt at hverken $E(y_t^2)$ eller kovariansen $cov(y_t, y_{t+k})$ er tidsafhængig, hvor k er alle heltal.

Såfremt at den forventede værdi af y_t er konstant og derfor ikke tidsafhængig må det altså være gældende at $E(y_t) = \mu$

For at variansen skal være konstant kan det tilsvarende beskrives matematisk ved følgende: $var(y_t) = \sigma^2$

Og slutteligt må det være gældende for kovariansen, såfremt den er konstant, at i ethvert tidspunkt i tiden t skal afstanden mellem y_t og y_{t+k} være tilsvarende til afstanden k. Det kan beskrives matematisk som følgende $cov(y_t, y_{t+k}) = k$, hvilket altså betyder at kovariansen ikke ændres over tid og dermed kan blive karakteriseret som uafhængig af tiden t. (Enders, 2015)

3.2 ARIMA

En af de hyppigste anvendte modeller til at prædiktere en stationær tidsserie er en AutoRegressive Integrated Moving Average herefter benævnt ARIMA og kan beskrives som en ARIMA(p,d,q), hvor p er modellens antal af laggede perioder, d er antallet af gange dataet er differentieret for at opnå stationaritet, q fortæller om modellens antal af moving average dele, det kan hertil bemærkes at en ARIMA (p,0,q) proces er tilsvarende til ARMA processen, da der ikke forekommer et differentieret led, såfremt at $d \geq 1$ må det altså betyde at tidsseriedataet ikke har været stationær og derfor er blevet differentieret d gange for at opnå stationaritet og tidserien er derfor integrated af ordren d og forklares som $x_t \sim I(d)$.

AR(p)-delen er en lineær komponent som gør at modellens prædiktering er baseret på tidligere værdier, altså laggede værdier af variablen, dette kan beskrives følgende:

$$AR(p) = y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \varepsilon_t \tag{6}$$

Hvor p angiver antallet af laggede perioder, altså hvor mange perioder bliver der kigget tilbage, y_t er den afhængige variabel, α_0 er en konstant og α_i er regressionens parametre og ε_t er den tidligere beskrevet white noice proces. (Enders, 2015)

MA(q)-delen er en komponent som prædiktere forecastet baseret på tidligere observerede fejlled og dette kan beskrives matematisk som:

$$MA(q) = x_t \sum_{i=0}^{q} \beta_i \varepsilon_{t-i} \tag{7}$$

Hvor q er lags, β_i er parametre og ε_t er en white noice proces. (Enders, 2015)

Hvor ARIMA modellen kombinerer alle tre elementer og ARIMA modellen kan beskrives matematisk på følgende form:

$$y_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i y_{t-i} + \sum_{i=0}^{p} \beta_i \varepsilon_{t-i} \tag{8}$$

(Enders, 2015)

3.2.1 Unit Root

En af de årsager til at datasættet skal differentieres skyldes at AR komponentet har et punkt på enhedscirklen, og derfor har noget som kendetegnes som en unit root, hvilket medfører at tidsserien derfor skal differentieres d gange indtil den bliver stationær.

En unit root kan forklares som en stokastisk trend i tidsserie økonometri som forekommer i AR eller ARMA tidsserier, som også kan blive forklaret som en random walk med en drift, hvilket betyder at tidsserien har et mønster, som er uforudsigelig og derved umuliggøre det at prædiktere et mønster,

hvilket tydeligvis er ufordelagtigt, når man vil forsøge eksempelvis at prædiktere en aktiekurs. (Enders, 2015)

3.2.2 Akaike Information Criterion

Når man skal vælge hvilken orden af ARIMA(p,d,q) kan der vælges et uendelig antal af p og q, men vælges der for mange forekommer det at man overfitter modellen, hvilket vil medføre at modellen er god til at forklare selve træningsdataet, men dårlig til at prædiktere den fremtidige kurs, hvorimod et for lavt antal af p og q kan medføre en model som har en høj sum af SSR (sum of squarred residuals) som medføre at modellen vil være svag. Ydermere påviste Granger og Newbold i 1974, at en spurious regression bestående af to uafhængige random drifts vil producere det man kalder en spuriøs regression, som er en regression med en høj R^2, altså forklaringsgrad, og derudover en signifikant t-statistik, så outputtet vil se fantastisk ud men uden nogen økonomisk relevans. Granger og Newbold fandt ud af, at hvis man kiggede på least-squares (mindst kvaderet estimater) belyser det at estimaterne ikke er konsistente, og derfor vil de statistiske inferens ikke kunne anses som værende gyldige (Enders, 2015).

En af de mest anvendte metoder til at vælge den rigtige mængde af p og q for at undgå at overfitte modellen eller spuriøse regressioner er at benytte sig af AIC, som står for The Akaike Information Criterion, som tager højde for både risikoen ved at overfitte modellen men samtidigt også risikoen ved at underfitte modellen, hvor en lavere AIC-score er målet og dermed forbundet med den bedste model. AIC kan beskrives på ligningsform som følgende:

$$AIC = Tln(sum\ of\ squared\ residuals) + 2n \tag{9}$$

Hvor T er antallet af observationer og n er antallet af estimerede parametre, altså antallet af p+q+k, hvor k er en mulig konstant (Enders, 2015).

3.2.3 Augmented Dickey-Fuller Test

En Augmented Dickey-Fuller, herefter benævnt ADF, er en mode der kan anvendes til at tjekke datasættet for stationaritet og om hvorvidt der befinder sig en tilstedeværelse af en unit root. ADF testen anses som en udvidelse af den klassiske DF test, hvor ADF testen tager højde for en højere orden, hvor den klassiske DF test er til lavere orden, så kan ADF testen håndtere en ARMA(p,q). ADF-testen kan beskrives vha. følgende ligning:

$$\Delta y_t = \gamma y_{t-1} + \sum_{i=2}^{n} \beta_i \Delta y_{t-i+1} + \varepsilon_t \tag{10}$$

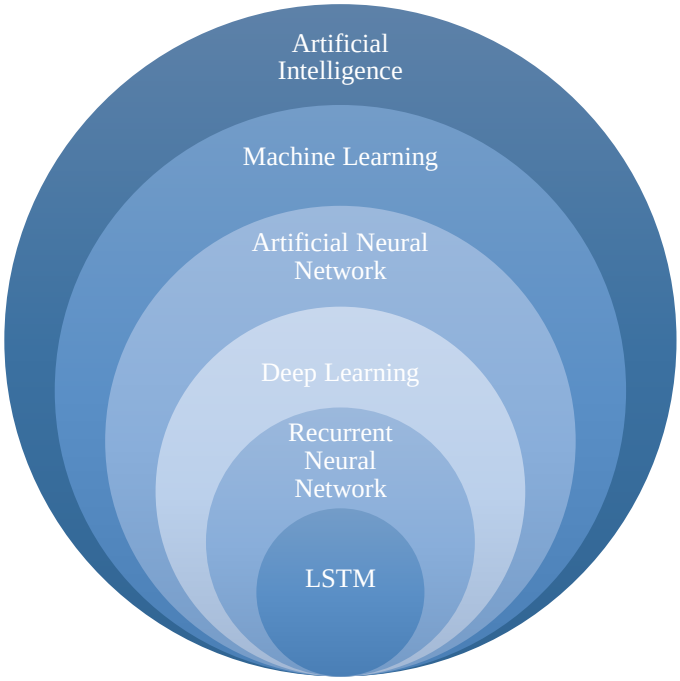
Hvor γ angiver om hvorvidt der en unit root tilstede, Δ angiver ændringen i tidsserien y_t i tidspunktet t. Fremgangsmåden når man foretager en ADF test er at der opstilles en nulhypotese, hvor man antager at såfremt $\gamma = 0$ så betyder dette, at der tilstedeværelse af en unit root og dermed vil datasættet ikke være stationært. Ligeledes hvis $\gamma < 0$ så er der ingen unit root til stede. Såfremt det er muligt at forkaste sin H0 om at $\gamma = 0$ så ville det være plausibelt at modellere ARMA(p,q) (Enders, 2015).

4 Ikke-lineære Modeller

Dette afsnit vil forklare det teoretiske grundlag for Long Short Term Memory (LSTM) modellen, men for fyldestgørende at kunne forstå LSTM er det essentielt at starte overordnet med AI og bevæge os til neurale netværk og hvilke elementer der indgår herunder.

Der er forsøgt at lave et figurativt overblik over hvad AI er for en skabning samt hvilke underkategorier der stammer fra AI.

Figur 1: Gruppering Af Kunstig Intelligens



4.1 Artificial Intelligience

Det kan være svært at forstå AI og hvad det er for en størrelse. Definitionen af AI beskrives bl.a. John McCarthy, der ofte er krediteret som skaberen af AI og en af de første personer der virkelig arbejdede med AI. Han beskriver AI således:

"It is the science and engineering of making intelligent machines, especially intelligent computer programs." – John McCarthy 2007

Til trods for at AI først virkelig har fået sit gennembrud i de senere år, som bl.a. skyldes de forhenværende begrundelser såsom teknologi og computerkraft, så har AI eksisteret i mange år, hvor

opstarten af AI vurderes at startede med John McCarthy som er krediteret som skaberen af AI tilbage i 1955, hvor formålet var at skabe en maskine der kunne tænke, lige så abstrakt som det der er kendetegnet ved den menneskelige natur, hvortil maskinen også skulle kunne løse problemstillinger og lære på egen hånd. (Childs, 2011). Til trods for manglende computerkraft og sofistikerede programmer så tog AI fart i 1980 det blev anvendt i et industrielt omfang (Russel & Norvig, 2010).

4.2 Machine Learning

Machine learning anses som værende en udspringer fra AI hvor AI havde til formål at prøve at skabe en maskine, der kunne simulere og agere menneskelig, hvor formålet med ML var at automatisere bearbejdningen af komplekst data, hvor maskinen kunne lære nye regelsæt og baseret på de nye regelsæt, drage konklusioner, hvor anvendelsen af ML er opgaver såsom klassificering og prædiktering. Hvilke anvendelser af ML man vil anvende, er baseret på hvilken form for machine learning der hentydes til. Overordnet skelnes der mellem to kategorier af ML, der eksisterer supervised (SML) og unsupervised (UML). SML er den hyppigste anvendt teknik til at træne sine neurale netværk. Det som adskiller de to, er at UML er anvendelig til data uden konkrete labels, hvorimod SML fungerer bedst til labels, såsom en prisændring for en aktie. (Ayodele, 2010).

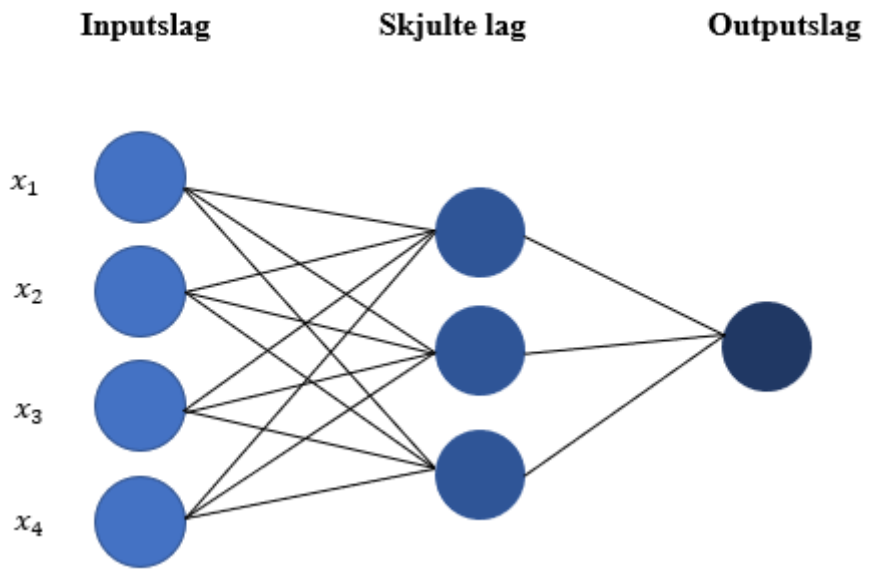
Det kan dog være svært at skelne mellem de to former for machine learning, da det kan forekomme kompliceret at vurdere om hvorvidt variablen anses som en feature eller et mål givet af en person. Et klassisk eksempel på UMl er opgaver hvor målet er at finde den bedste præsentation af dataet, hvor programmet skal forsøge at finde det bedste udtryk for datasættet samtidigt med at den adlyder angivne begrænsninger. Så kort sagt vil man prøve at finde et mere sparsommeligt udtryk for x. (Goodfellow et al 2016).

I dette speciale vil der derfor blive anvendt Supervised ML, hvor algoritmen bliver præsenteret for dele af input- og output data, hvormed det kan træne og derefter kommer med prædikteringer baseret på dens træning (Ayodele, 2010). Begrundelsen for valget af machine learning bliver yderligere understøttet af Chong et Al 2017 som argumenter for at SML er særdeles egnet til aktiemarkedsforudsigelser grundet dets egenskaber og natur.

4.3 Artificial Neural Network

Et Artificial Neural Network (ANN) eller kunstigt neurale netværk er en estimationsmodel som anvendes til at finde et system i datasæt, hvor det kan klassificere eller prædiktere på nyt data, baseret på det data som modellen er blevet trænet på. ANN kan anses som værende en mere kompleks model sammenlignet med klassiske SML-metoder. ANN er inspireret af vores forståelse af den menneskelige hjerne. ANN består af neuroner som modtager og derefter bearbejder information. Et klassisk eksempel på en ANN er et Feedforward Neural Network også kendt som (FNN). Hvor formålet med en FNN er at approksimere en funktion hvilket sker ved at passere information igennem et netværk fra venstre til højre (Goodfellow et al, 2016).

Figur 2: Simple Feed-Forward ANN



Egen fremstilling baseret på (Gupta. 2017)

Som det ses i ovenstående fremstilling, så er inputs og outputslaget forbundet via et skjult lag, hvor det illustreres at inputslaget er bestående af fire neuroner og det skjulte lag har tre neuroner, hvor det ses hvordan de tre forskellige lag i modellen er forbundet, hvor mængden af neuroner i inputslaget er tilsvarende antallet af forklarende variable. Hver neuron modtager dataet samt en forklarende variabel og dertilhørende vægt som beskriver hvor vigtig variablen er for modellen, og slutteligt et outputslag som består af en enkelt neuron (Goodfellow et al, 2016).

Begrundelsen for navnet skjulte lag skyldes at dette lærings-lag foregår på en ikke-visuel måde, altså en form for black box, hvilket gør det umuligt prævist at bevidne hvad der foregår inde i modellen, hvilket står som stor kontrast til den økonometriske ARIMA model, hvor alt kan forklares, ses og tydeligt matematisk formuleres. Det kan derfor være svært at kortlægge og præcist forklare processen fra input til output og der kan selv i enkelte modeller og enkelte forecast forekomme millioner af forskellige kombinationer i form af vægte mellem de forskellige lag. (Molnar, 2019). En af de unikke egenskaber ved ANNs er at det selv lærer fra inputsdataet ved at ændre på parametre for derved at opnå den optimale prædikteringsevne og dette sker ved back-propagation som beskrives i et senere afsnit.

4.3.1 Aktiveringsfunktioner

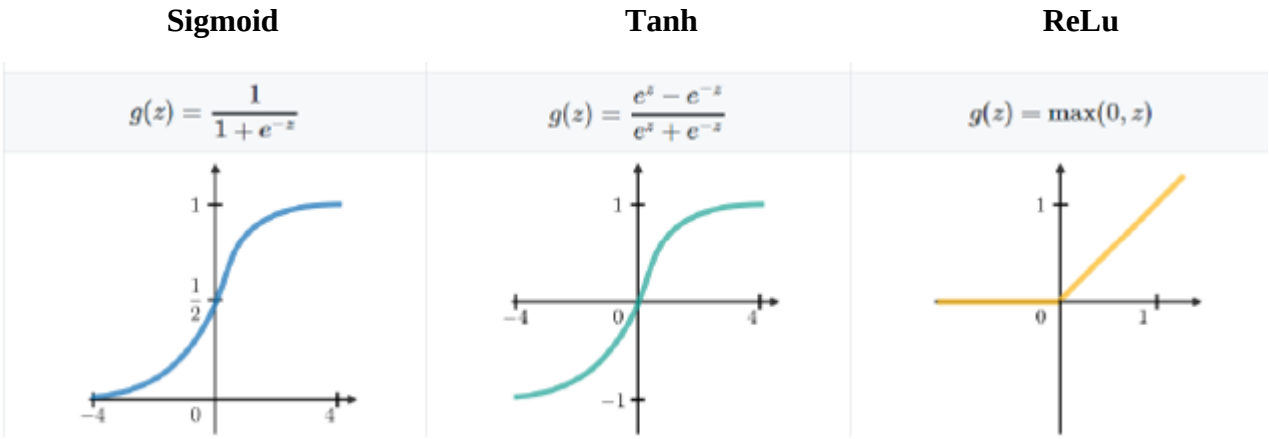
Aktiveringsfunktionen er den funktion, der i det skjulte lag transformerer informationen der bliver givet. Selvom aktiveringsfunktionen perception ikke er den hyppigste i moderne ML anvendelse men derimod en sigmoid funktion, så er det vigtigt at forstå perception funktionen for at kunne forstå sigmoid funktionen. Perception har et antal af binære inputs x_j for det j'te input hvor vigtigheden af neuronens binære ouput 0 eller 1 forekommer baseret på hvorvidt den vægtede sum er større eller mindre end en opsat grænseværdier og dette kan beskrives som $\sum_j w_j x_j$ og dette kan udtrykkes på matrix-form som (Nielsen. 2019).

$$Output = \begin{cases} 0\ hvis \sum_j w_j x_j \ \leq \ grænseværdi \\ 1\ hvis \sum_j w_j x_j > grænseværdi \end{cases} = \begin{cases} 0\ hvis\ w \cdot x + b \leq 0 \\ 1\ hvis\ w \cdot x + b > 0 \end{cases} \tag{11}$$

Problemet med denne grænsebaserede aktiveringsfunktion medfører, at det neurale netværk ikke kan lære de ikke-lineære sammenhænge hvilket medfører komplikationer, når der skal prædikeres på komplekst data, hvor der her ofte er indforstået, at der forekommer nogle ikke-lineære sammenhænge. Derudover medfører den binære grænsebaserede aktiveringsfunktion også, at små ændringer i eksempelvis bias eller vægtene, at neuronerne i modellen går fra at aktivere, til ikke at aktivere, hvilket selvføre må medføre at resultatet ændres drastisk (Nielsen. 2019).

Der kan anvendes forskellige aktiveringsfunktioner som udvælges af modelkonstruktøren, hvor de hyppigste anvendte er illustreret i efterfølgende figur.

Figur 3: Aktiveringsfunktioner



(Egen fremstilling, baseret på Teng et al,. 2020)

Sigmoid-funktionen fungerer binært, så outputtet er enten 0 eller 1 og er ikke-lineær. Sigmoid funktionen, modsat den tidligere beskrevet perception, er at den er bedre til at foretage mindre ændringer i outputtet ved ændring af vægtene. Sigmoid-funktionens problem derimod er, at den bliver påvirket af noget som hedder vanishing gradient, som medfører at funktionen efter et vist punkt nægter at lære mere og derfor har et loft på hvor meget det potentielt kan lære. En logistisk sigmoid-funktion kan beskrives matematisk som:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{12}$$

Hvor x er et givent input, hvor dette input tager en værdi i intervallet 0 til 1.

I det neurale netværk hvor sigmoid-funktionen består af variablen, z, som kan skrives på ligningsform som:

$$z = \sum_{j=1}^{n} (w_j x_j) + b \tag{13}$$

Hvor z udregnes fra forrige lag x med den tilhørende vægt og b er et bias-led.

hvor outputtet bestemmes af neuronerne som anvender sigmoid-funktionen som først beregner z fra inputs fra tidligere lag af x og de dertilhørende vægte w, hvor sigmoid-funktionen transformerer z som outputtet, så

$$O= \sigma(x) = \frac{1}{1+e^{-x}} \tag{14}$$

Tanh-funktionen er en skaleret sigmoid-funktion med et interval indenfor -1 til 1. Tanh-funktionens fordel er at negative inputs vil fremstå som meget negative og alle 0-inputs vil stadig være tæt på 0. Tanh-funktionen anvendes primært til klassificeringsproblemer. Ligesom Sigmoid-funktionen så lider tanh-funktionen også af vanishing gradient problemet, hvor den har et loft på dens indlæringsevne og når et punkt, hvor den stopper med at lære.

RELU-funktionen også kaldet Rectified Linear Units (ReLu), som er blevet den standart og fortrukne aktiveringsfunktion i de fleste NN (Neurale Netværk) og DL(Deep learning) modeller. Ligesom Sigimoid-funktionen så transformeres den vægtede sum af input og bias som kan skrives på ligningsform som:

$$O = f(z) = \begin{bmatrix} 0 \; hvis \; z < 0 \\ z \; ellers \end{bmatrix} = \max(0, z) \tag{15}$$

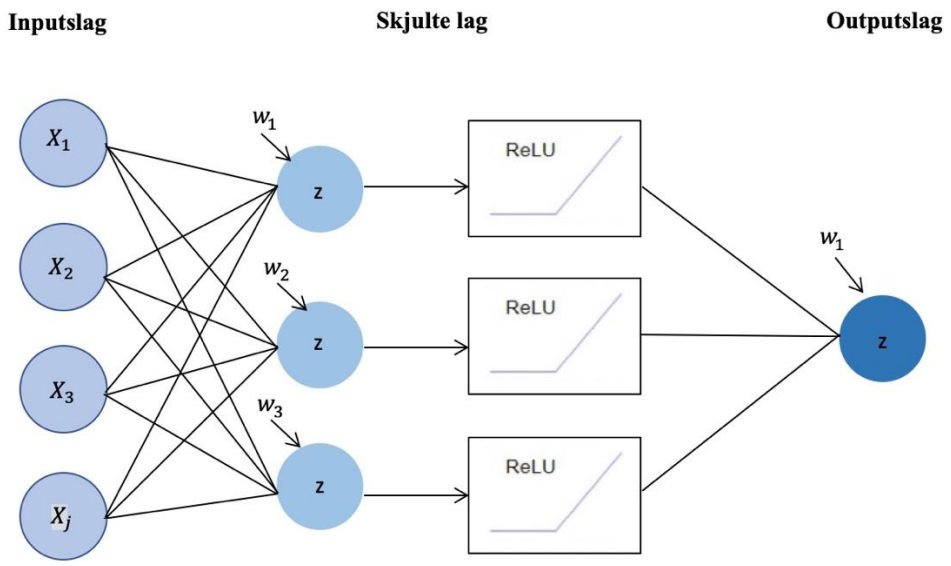
ReLu giver outputs i intervallet 0 til uendeligt. Som det også ses grafisk på figur 3, så er der ingen negative værdier såfremt ReLu-funktionen anvendes, hvilket skyldes at negative z-værdier får værdien 0, og dette kan medføre en mindsket evne for modellen at træne på datasættet (Sharma, S, 2017). En af egenskaberne ved ReLu-funktionen minder om den lineære-aktiveringsfunktion, dette betyder også at netværket er hurtigt til at lære og optimere på de inputs den får givet, hvilket gør anvendelsen af ReLu-funktionen efficient at anvende og den tillader anvendelse af back-propagation (Goodfellow et al, 2016).

Output

Figur 4 opsummerer konstruktionen af et Feed-Forward Neuralt Netværk, hvor der modsat den helt simple model er inkluderet et bias-led samt de processer der tidligere er beskrevet, hvor der først beregnes neuronens inputvariabel z, som den vægtede sum af x_j hvor bias-leddet, b_j, bliver adderet

hvorefter den transformeres baseret på den anvendte aktiveringsfunktion, hvor der i både figuren er vist en ReLu-funktion som tidligere beskrevet, var blevet den fortrukne aktiveringsfunktion. Slutteligt, som der også er illustreres transformeres z vha. aktiveringsfunktionen til den endelige prædiktion (Goodfellow et al, 2016).

Figur 4: Feed-Forward Neuralt Netværk



4.3.2 Træning af ANN

Når man skal træne en ANN startes der ofte med at man opdeler datasættet, hvor modellen får et træningssæt, som modellen træner på og lærer mønstrene i datasættet at kende, derudover splittes der også op, så der er et testsæt som anvendes til at se hvordan modellen klarer sig, hvorefter dens prædikteringsevne skal evalueres på nyt data, som den ikke har set under træningen og dette kan være en indikator på hvor god modellen er. I nogle tilfælde vil datasættet også blive splittet op i tre, hvor det sidste datasplit er et valideringssæt, som har til hensigt at teste for overfitting af netværksparametrene, hvilket betyder, at såfremt netværket tilpasser sig for meget træningssættets mønstre, så vil dens performance være god på træningssættet men være dårlig til at prædiktere og evaluere nyt data uden præcis samme mønstre.

Når man starter træningen af ANNs så bliver vægtene og bias tilfældigt valgt, og disse vægte og bias bliver ændret under modellens træning ofte i form af en tabsfunktion, som evaluerer modellens performance. Tabsfunktionen indikerer hvor langt væk det rigtige svar er for det prædikterede. Et

eksempel på en tabsfunktion, er RMSE som beskrevet i et senere afsnit. Formålet med træning af ANNs er at finde et lokalt minimum for tabsfunktionen og dette sker i form af back-propagation (Goodfellow et al, 2016).

Teoretisk er det optimale, når man vil udregne det lokale minimum for tabsfunktion som bliver beskrevet i dybden i efterfølgende afsnit, så er fremgangsmåden at udregne det individuelt for hver neuron i netværket, men da denne proces er både tidskrævende og ikke særlig efficient og samtidigt kræver meget computerkraft, så kan der anvendes en optimeringsfunktion. En måde til at håndtere det at effektivisere processen er at opdele dataet i mindre samples som i konstruktionen kaldes for ’batches’ hvor man sender disse igennem netværket, hvor hensigten er at modellen skal udregne tabsfunktionen for udpluk af datasættet, hvorefter modellen ender i samme lokale minimum for funktionen på en hurtigere måde end ved at køre det for hele datasættet.

For at effektivisere træningen af ANN kan man yderligere tilføje en momentums-funktion hvor der i specialet er anvendt Adaptive Momentum Estimation (ADAM) og kan ses som en læringsrateoptimerings-funktion, som effektivt og hurtigt formår at finde de korrekte vægtninger og bias-led og kan anses som en forlængelse af den stokastiske gradient, herefter bare benævnt gradient, som bliver forklaret i efterfølgende afsnit sammen med back-propagation. (Bushaev, V. 2018)

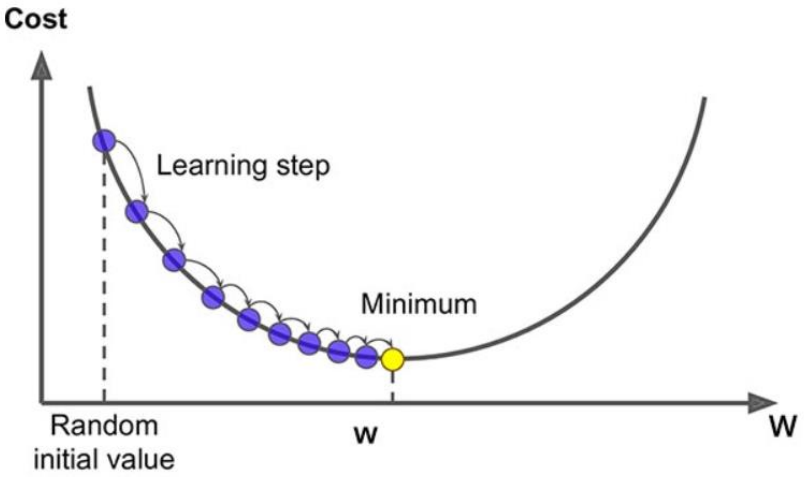
4.3.3 Stokastisk gradient & Back-propagation

En af årsagerne til at ANNs lære over tid sker igennem Back-propagation (BP), for at forstå BP kan man kigge på outputslaget som tidligere beskrevet, hvor der kun er en enkelt neuron. Som tidligere beskrevet, så kører processen i et neuralt netværk normalvis fra venstre til højre, så hver observation er kørt igennem de forskellige lag og prædikteringen kommer ud som outputtet, hvor ouputtet kan ændres ved at ændre på vægtene, bias og input som ses i ligning 16. Når et ANN kommer med outputs i form af prædikteringer, så er det muligt at udregne en tabsfunktion og derved beregne hvordan modellen skal ændres for at nå det lokale minimum, denne proces foregår vha. en gradient som kan beskrives som en optimeringsalgoritme, som er ved til at finde det lokale minimum, hvor gradienten viser hvilken retning inputsene skal ændres, for at nå det lokale minimum og denne proces gentages altså utallige gange indtil man rammer minimum. Det matematiske bevis anses som værende yderst kompliceret og derudover ligge udenfor afhandlingens emnefelt. Men tabsfunktionen kan på matematisk vis formuleres på følgende ligningsform:

$$C(w,b) = \frac{1}{2n}\sum_{x}\left|\left|y(x) - a\right|\right|^2 \tag{16}$$

Hvor w er en samling af alle vægtene fra netværket, b er alle bias-ledene a er et vektor af alle outputs for hvor x er træningsinputtet og y(x) er det ønskede output og n er det totale antal af træningsinputs. Det kan ses at vektoren a er afhængig af x,w og b. Den kvaderede tabsfunktion er også tit betegnet som MSE. Da C(w,b) er positiv da alle term er positive og dermed kan C ikke tage negativ form. Ydermere når tabsfunktionen er lille når C(w,b) ≈ 0 når y(x) er approximeret lig med outputtet a, for alle trænings inputs x. Gradient-algoritmen har derfor klaret det godt såfremt den har været i stand til at finde vægte og bias-led så C(w,b) ≈ 0. Ligeledes klarer gradient-algoritmen det dårligt hvis C(w,b) har en stor værdi og dette betyder også at y(x) ikke er tæt på at være lig med outputtet a. (Nielsen. 2019). Gradient-algoritmen kan ses afbilledet på nedenstående figur 5.

Figur 5: Gradient Descent



Note: Viser de skridt en NN tager for at mindske tabsfunktionen. (Nielsen, M. 2019)

Hvor der ses et startpunkt ud fra den oprindelige værdi, hvor den igennem forskellige iterationer nærmer sig det lokale minimum markeret med gult og det er så den endelige værdi hvor tabsfunktionen er minimeret. Det ses også at der i starten fortages større skridt, hvor skridtstørrelsen kan forklares som: $Skridtstørrelse = hældning \cdot Lærringsrate$
og når denne skridtstørrelse når så tæt på 0 som overhovedet muligt, så vil den være så tæt på det lokale minimum hvorefter gradient-algoritmen stopper med at foretage flere skridt.

Back-propagation (BP) blev oprindeligt introduceret i 1970, men vigtigheden af anvendelsen af BP blev først tydeligt efter 1986 efter en artikel af Rummelhart et al. De fandt ud af, at BP i flere forskellige neurale netværk var markant hurtigere til at lære end de forhenværende metoder som kommer i spil, når tabsfunktionen har en høj værdi, som gradienten prøver at minimere igennem ændring af parametrene som beskrevet i ligning 16, hvor problemet består i, at jo dybere modellen er, altså jo flere lag der findes, desto mere kompleks bliver modellen, da alle lag er afhængige af forrige lag, så fungerer BP som en metode til at gå modsatte vej i netværket, hvor BP kortlægger fejlene fra output til input med de vægtninger og bias man benyttede til at finde det rigtige input for at finde det lokale minimum og på denne måde kan BP medvirke til at finde de passende vægtninger eller bias-led for hvert led for at minimere fejlen, så man kan sige at BP og gradient algoritmerne arbejder sammen om at træne netværket så netværkets parametre ændres minimalt hver gang for at opnå det optimale punkt for modellen, hvor modellens præcision er så god som muligt samtidigt med at tabsfunktionen holdes så lav som mulig og dette kaldes også for lærringsraten og beskriver altså hvor hurtigt modellen er til at ændre vægtene og bias-ledene når den møder ny information. (Nielsen, 2019)

4.4 Recurrent Neural Network

Recurrent Neural Network (RNN) er en afgrening af ANN som kom frem i slutningen af 1980'erne, efter Rummelhart et al i 1986, som tidligere nævnt, kom frem med en artikel der fremhævede vigtigheden samt anvendelsen af BP. RNN kan forstås som et neuralt netværk der kan forstå autoregressive egenskaber som forklares i forrige afsnit 3. ANN's kendetegnes som et neuralt netværk, der kan forstå og bearbejde sekvensdata, hvor sekvensdata kan beskrives som den data hvor der forekommer en tidsdimension, hvor vigtigheden af sekvensen vil være afgørende. Det der gør RNN specialet fra tidligere de neurale netværk, som tidligere er beskrevet, er at dette netværk er tilbagevendende. Dette kan også ses for ligningen af et klassisk eksempel på det dynamiske system som kan gives ved ligningen:

$$s^{(t)} = f(s^{(t-1)}; \theta \tag{17}$$

Hvor s^t henviser til stadiet i tiden t, hvorefter man kan aflæse at definitionen af s ved tiden t refererer til den samme definitionen fra forrige periode, altså t-1, hvilket betyder at den nuværende definition

bliver baseret på den forrige periode sammen med en parameter θ. Dette betyder altså at det skjulte lag i en RNN model vil værdien af noderne blive baseret på tidligere værdi samt en parameter.

Såfremt det dynamiske system indeholder et eksternt signal, så kan det udtrykkes:

$$s^{(t)} = f(s^{(t-1)}, x^{(t)}; \theta \tag{18}$$

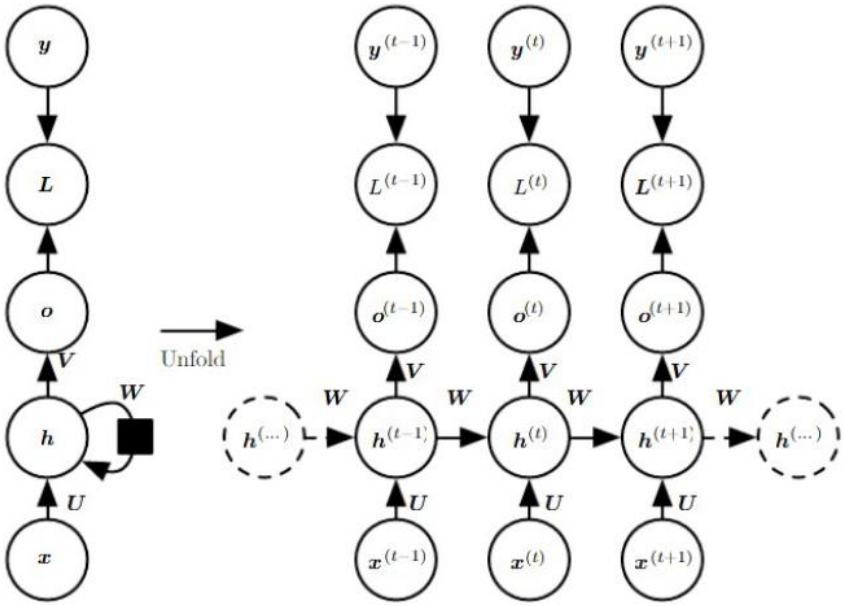
Hvor det eneste nye fra ligning 18 er det eksterne signal $x^{(t)}$.

I ovenstående ligning har den nuværende definition af s i tiden t hele informationen om sekvensen, så for at forstå det skjulte lag i en RNN model så kan vi omskrive ligning 18 for at finde definitionen af værdierne i det skjulte lag. Omskrivningen af ligning 18 sker ved at omskrive variablen $s^{(t)}$ til $h^{(t)}$ for at indikere at det er "hidden", altså skjult. Så neuronerne i det skjulte lag kan defineres matematisk således:

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta \tag{19}$$

Et eksempel på et RNN kan illustreres ved figur 6

Figur 6: RNN med input og output



Note: Venstre side viser RNN hvor højresiden er samme system bare mere detaljeret. (Goodfellow et al,. 2016)

De to ovenstående figurer er egentlig det samme netværk, mens det højerestående er ’foldet’ ud så det har samme struktur som de tidligere beskrevet neurale netværk. Som det ses på venstre-siden så modtager modellen et sekvensinput med x-værdier, hvorefter det skjulte lag h opdateres og dette h-lag fortsætter med at udvikle sig, så definitionen af h hele tiden bliver fornyet baseret på forrige definitioner af h og påvirker derfor de fremtidige stadier af h, denne form for recurrent, altså tilbagevendende, bliver markeret i modellen som en sort boks. Baseret på den information modellen modtager fra det skjulte lag, h, så medfører dette en outputs-sekvens, o. Når modellen trænes, så opstår der en tabsfunktion, som også beskrevet i afsnit 4.3.3, hvor værdien af tabsfunktionen L afhænger af forskellen mellem de realiserede værdier y og prædikterede værdier $\hat{y}$.

RNN modellen har et input som har skjulte forbindelser som kan beskrives ved en vægtmatrice U, som er en skjult-til-skjult tilbagevendende forbindelse, som kan beskrives ved en vægtmatrice W og samt et skjult-til-output matrice V, og disse matricer kan også kan aflæses grafisk på venstreside af modellen. RNN modellen anvender, som de tidligere neurale netværk også aktiveringsfunktioner og tabsfunktioner, men udskiller sig i den forstand, at RNN arbejder med sekvensdata og derfor regnes tabsfunktionen baseret på en sekvens og ikke hele datasættet.

Outputtet i af neuronerne fra det skjulte lag kan beskrives på ligningsform:

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)} \tag{20}$$

$$h_t = \tanh\left((a^t)\right) \tag{21}$$

$$o^{(t)} = c + Vh^{(t)} \tag{22}$$

Hvor der ses at der i eksemplet anvendes en tanh aktiveringsfunktion som tidligere beskrevet, derudover, c og b er bias-vektorer, samt de tidligere nævnte vægtmatricer, U,V,W for henholdsvis input-skjult lag, skjult-ouput lag og slutteligt skjult-skjult lag er der h_t som er neuronerne i det skjulte lag, b er bias, og hvor W og U er de tidligere beskrevet vægtmatricer. Først udregnes $a^{(t)}$ med udgangspunkt i den valgte aktiveringsfunktion samt parametrene fra bias-leddet b samt vægtmatricen W og neuronen fra forrige skjulte lag, altså $h^{(t-1)}$ samt vægtmatrice U og inputs sekvensen, x fra nuværende periode. Outputtet findes ved bias-leddet c samt vægtmatricen V der er ganget på nuværende tidspunkt t af $h^{(t)}$.

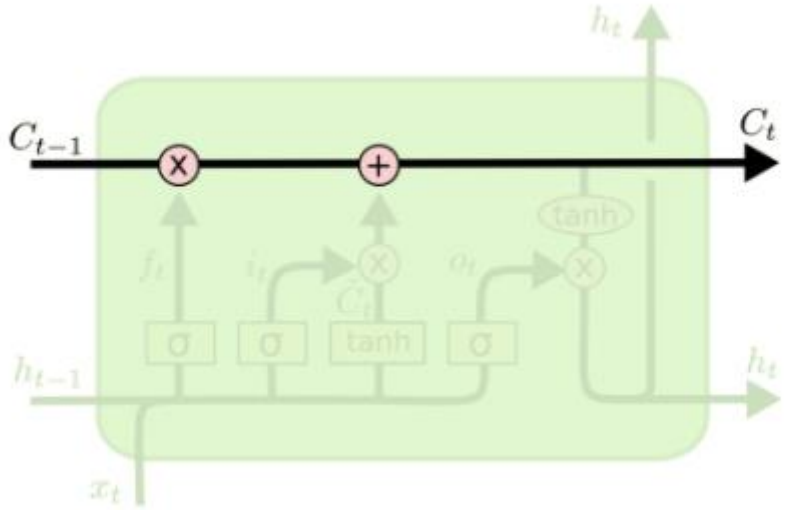
Problemet med anvendelsen af RNN kan også ses i figur 6 hvor vægtmatricerne U,W og V ikke kun bliver tildelt den nuværende inputssekvens $x^{(t)}$ men også forrige og fremtidige værdier i sekvensen dette medfører at RNN bevarer en hukommelsen af sekvensen, men dette medføre et problem da modellen lider af 'short-term memory' altså en kort hukommelse. Dette problem kaldes for the vanishing gradient problem, hvilket sker på baggrund af, at modellen ikke efficient kan anvendes til at ændre vægtene og bias vha. back-propagation som tidligere modeller kunne udnytte under træning. Gradienterne ved en RNN har en tendens til at forsvinde deri opstår et udtryk kendt som 'the vanishing gradient problem', kort sagt så bliver gradient-værdierne meget små og bevæger sig mod 0 og dette implicerer at modellens tabsfunktionen ikke kan minimeres, hvor RNN forholdsvis hurtigt støder på dette problem og derfor kan RNN modeller normalt ikke lære langsigtede mønstre. (Goodfellow et al, 2016)

4.5 Long Short Term Memory

For at imødekomme 'the vanishing gradient problem' som beskrevet i forrige afsnit så blev der i 1997 præsenteret en model af Hochreiter & Schmidhuber som kan anses som værende en videreudviklet RNN-model, Long Short Term Memory (LSTM) modellen som udnytter en cellestruktur som modsat RNN har en hukommelse der strækker sig over kortsigtet såvel som langsigtet. (Goodfellow et al, 2016). Følgende afsnit med gennemgangen af LSTMs processer med LSTM gates vil være baseret på Christopher Olahs artikel fra 27 August 2015 "Understanding LSTM Networks"

En af de essentielle fundamenter ved LSTM ligger i et celle stadie, som fungerer som et transportbånd som løber igennem en lige linje igennem hele kæden, hvor der forekommer nogle mindre linære interaktioner, dette kan ses illustreret på figur 7.

Figur 7: Cellestadiet



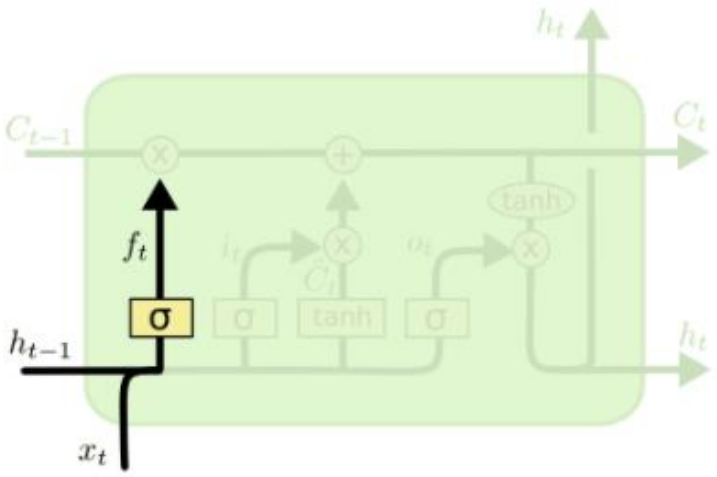
(Olah, C. 2015)

Informationen kan nemt og effektivt suse uændret igennem modellen, men LSTM har egenskaben til at ændre informationen og dette sker ved enten at tilføje information til cellestadiet eller fjerne information og denne proces bliver styret af celleporte, hvis hensigt er hjælpe cellestadiet, i den forstand, at de vurderer hvorvidt den information der bliver introduceret er relevant eller ligegyldig. Såfremt informationen er ligegyldig så informationen bliver fjernet fra cellestadiet og ligeledes hvis informationen er vigtig, vil informationen blive husket, og heri opstår hukommelses-delen i LSTM-modellen. Celleportene består af sigmoid-funktioner, som også beskrevet i afsnit 4.3.1 samt figur 3, hvor der bliver givet en værdi i intervallet 0;1 hvor såfremt værdien er 1 så vil hele informationen passere og er værdien 0, så vil ingen information blive passeret videre. LSTM består af tre porte og disse porte er følgende:

- Forget Gate
- Input Gate
- Output Gate

4.5.1 Forget Gate

Figur 8: Forget Gate



(Olah, C. 2015)

Det første skridt i LSTM-modellen er at vurdere hvilken information der skal passere og hvilket der skal kasseres for cellestadiet C_t, og dette sker i ”Forget-gate”. Beslutningen bliver udført af et sigmoid-lag som baseres på nuværende inputtsekvens i tiden t, x_t ,samt outputssekvensen fra forrige stadie, h_{t-1}, hvor der kommer et sigmoid-output for hvert tal i cellestadiet C_{t-1} hvorefter outputtet, som er, hvor meget information der skal passeres og glemmes, kan blive vurderet. Denne proces kan beskrives matematisk på ligningsform som:

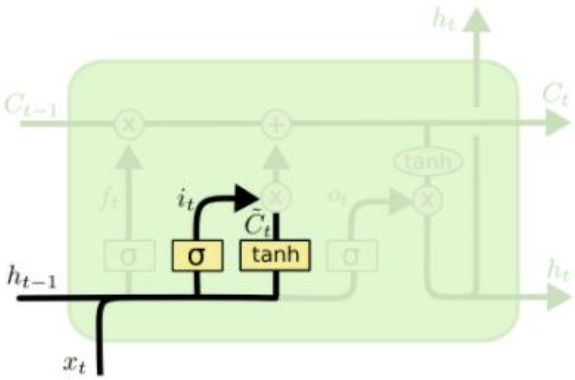
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{23}$$

Hvor f_t er outputtet fra forget-porten, σ er en sigmoid-funktion W_f og b_f er henholdsvis vægte og bias-vægte. (Olah, C. 2015)

4.5.2 Input Gate

Del 1

Figur 9: Input gate del 1



(Olah, C. 2015)

Det efterfølgende skridt i LSTM er at beslutte hvilken information der skal opdateres til cellestadiet c_t. Denne proces foregår som en to-trins proces i input-porten. Det første informationen bliver udsat for er et sigmoid-lag, σ, der bestemmer hvilken information, der skal opdateres baseret på deres værdi mellem 0;1, hvor b og W er henholdsvis bias og vægtmatricer. Hvor h_{t-1} er forrige stadies værdi og x_t er nuværende inputsvariabel x_t.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{24}$$

Simultant møder informationen et tanh-lag der skaber og transformerer informationen med samme fremgangsmåde som i ligning 24, denne alternative vektor med nye værdier for cellestadiet navngives $\widetilde{C_t}$ og udtrykkes på ligningsform som:

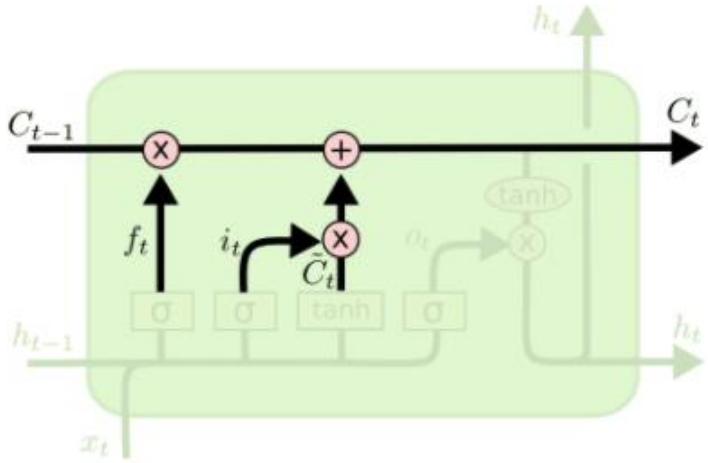
$$\widetilde{C_t} = \tanh\,(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{25}$$

Dernæst forekommer en kombination af input-porten og tahn-laget der vurderer hvilken information der skal videregives til cellestadiet C_t. Outputtet fra input-porten kan defineres i_t som på ligningsform tilsvarende ligning 25 og defineres derfor på ligningsformen:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{26}$$

Del 2

Figur 10: Input gate del 2



(Olah, C. 2015)

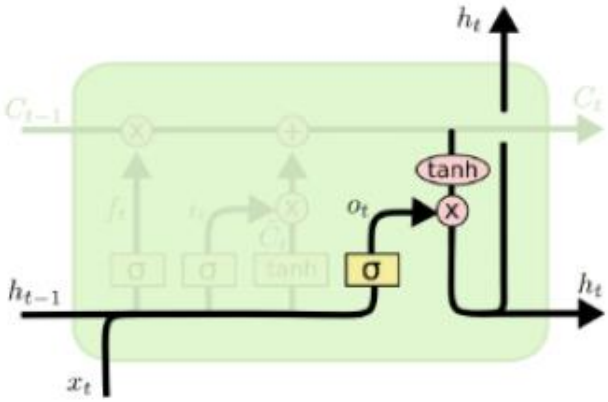
Trin et afgjorde hvad der skulle udføres og nu skal denne proces så udføres. Dette sker ved at det forrige cellestadie, C_{t-1} skal opdateres til det nuværende cellestadie C_t. Det gamle cellestadie C_{t-1} bliver opdateret ved at anvende følgende komponenter, outputtet fra forget-porten f_t , output fra input-porten, i_t, værdierne fra forrige cellestadie, c_{t-1} samt vektoren med de nye potentielle værdier for cellestadiet, $\widetilde{C_t}$. Dette kan udtrykkes på ligningsform:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t} \tag{27}$$

Først glemmes det gamle irrelevante information ved at multiplicere outputtet fra forget-porten f_t med forrige cellestadie C_{t-1} hvorefter ny information tilføjes ved at multiplicere outport-porten i_t multipliceres med vektoren potentielle værdier for cellestadiet $\widetilde{C_t}$. Cellestadiets værdier er opdateret og sendes videre til den sidste port i LSTM-systemet.

4.5.3 Output Gate

Figur 11: Output gate



(Olah, C. 2015)

Sidste del af LSTM processen er at afgøre hvad outputtet, o_t, skal være og dette sker naturligvis i output-porten. Outputtet vil være baseret på cellestadiet men vil forekomme som en filteret version af cellestadiet. Først mødes informationen af et sigmoid-lag der bestemmer hvilken del af cellestadiet C_t der skal passere til output baseret på sigmoid-værdien mellem 0;1, hvilket matematisk kan forklares som:

$$o_t = \sigma(W_0[h_{t-1}, x_t] + b_0 \tag{28}$$

Hvorefter cellestadiet transformeres ved at det bliver kørt igennem en tanh-aktiveringsfunktion hvilket skalerer værdierne i intervallet -1;1 hvorefter det transformerede cellestadie bliver multipliceret med outputtet fra output-porten som illustreres i figur 11, og dette udtrykkes på ligningsform som:

$$h_t = o_t \cdot \tanh\,(C_t) \tag{29}$$

4.6 Hyperparameters

En essentiel del af det at forbedre det neurale netværk, er at styre modellens adfærdsmønster og disse indstillinger foregår igennem det, som er kaldet for ’hyperparametres’ som kan beskrives som hyperparameter, hvis værdi benyttes til at kontrollere modellens indlæringsevne. Det skal her

bemærkes at disse hyperparametre ikke styres af de parametrene der kan trænes såsom bias og vægte. Antallet af parametrene er udvalgt af modelkonstruktøren og ændres ikke under træningen af modellen. I eksempel med en ANN-model så vil disse parametre indebære antallet af neuroner og antallet af lag og hvilken aktiveringsfunktion der skal anvendes i modellen. Det er vigtigt at kunne skelne mellem hyperparameter og model parametre, hvor hyperparameterne er med til at styre hele modellen og bestemmes af modelkonstruktøren og disse bliver optimeret før selve træningen af modellen begynder. Nogle af de væsentlige hyperparameter, der er med til at styre modellens adfærdsmønster indebærer:

- Antallet af neuroner og skjulte lag
- Læringsrate
- Aktiveringsfunktionen
- Epoker & Batch Størrelse

Antallet af neuroner og skjulte lag afgør hvor mange inputsdimensioner der medtages i modellen samt skjulte lag, som er med til at give modellen mere dybde og kompleksitet. Formålet med valg af skjulte lag involverer at give modellen nok kompleksitet til at kunne forestå sammenhænge i datasættet uden at være så kompleks, at den lærer unødvendige ting hvilket vil implicere, at modellen overfitter og derved ville have en høj præcision på træningssættet, men en lav præcision på valideringssættet. Aktiveringsfunktionen er vigtig da den er med til at afgøre om hvorvidt modellen kan forstå ikke-lineære forhold, eksempelvis sigmoid-funktionen som tidligere beskrevet i afsnit 4.3.1).

Epoker afgør hvor mange gange optimeringsalgoritmen skal køres for at minimere tabsfunktionen og lærringsraten, som tidligere forklaret i afsnit 4.3.3, beskriver hvor hurtigt modellen skal justere vægtene og bias-ledene i modellen efter hver epoke (Goodfellow et al, 2016).

4.7 Opsummering

Der findes mange forskellige typer af neurale netværk med fordele og ulemper, men det de alle har tilfælles er at de er inspireret af den menneskelige hjerne og består af neuroner, som påvirkes af hinanden igennem vægtede forbindelser i netværket. Neuronerne blive transformeret baseret på den valgte aktiveringsfunktion, som hjælper med at ændre inputtet til outputtet. Vægtene og bias-leddene i modellerne bliver justeret vha. to algoritmer der arbejder sammen, gradienten og back-propagation som prøver at justere vægtene og bias-leddene, for at finde det lokale minimum for tabsfunktionen.

Problemet med RNN er at the vanishing gradient problem opstår, hvilket betyder at modellen relativt hurtigt stopper med at lære og minimere tabsfunktionen og derfor har problemer med lange sekvenser og kendetegnes for at have en korttidshukommelse hvorefter LSTM-modellen, som også er den anvendte model for afhandlingen introduceres.

LSTM-modellen blev introduceret af Hochreiter & Schmidhuber i 1997 kunne imødekomme the vanishing gradient problem, og led modsat den traditionelle RNN-model ikke af korttidshukommelse. LSTM-modellen er bestående af et cellestadie og tre porte, forget-porten, inputporten og output-porten, hvor portenes funktioner er hele tiden med til at afgøre hvilken information, der skal sendes igennem systemet. Informationen bliver alle transformeret igennem en sigmoud-funktion, som giver informationssættet en værdi i intervallet 0;1, som beskriver hvor vigtig informationen er, og hvor meget der skal videregives til cellestadiet. Når informationen bliver kørt igennem systemet, vil cellestadiet hele tiden blive opdateret med ny information samtidigt med at glemme den ligegyldige information, hvilket giver modellen den egenskab at lære long-term adfærdsmønstre. LSTM-modellens kan konstrueres på uendelige måder og metoder og dette kan opdeles i to forskellige kategorier, parametre såsom bias, input og vægt og hyperparametre som justeres og vælges før modellen begynder at træne på træningsdataet for at lære dets adfærdsmønstre.

Hyperparametrene involverer valg af neuroner, antallet af skjulte lag og aktiveringsfunktioner og batch size og disse udvælges af modelkonstruktøren med den hensigt at vælge den optimale konstruktion for at give modellen den bedste mulighed for at lære adfærdsmønstrene.

5 Evalueringsmuligheder

Der findes mange forskellige evalueringsmuligheder der kan anvende som et værktøj til konkret at vurdere modellernes performance af forcastet. Specialet tager udgangspunkt i RMSE og MAPE som tabsfunktioner, der beskriver modellernes evne til at forudsige prisen for de udvalgte finansielle aktiver. Pesaran And Timmerman testen benyttes til at vurdere modellernes evne til at forudsige retningen for de givne aktiver. Slutteligt benyttes der en Diebold-Marino test for hvorvidt modellerne er statiske forskellige i nøjagtigheden af deres forecast.

5.1 Root Mean Squared Error

Root Mean Squared Error, herefter benævnt RMSE, benyttes til at teste modellens evne til at prædiktere, i afhandlingens tilfælde aktiekursen, hvor RMSE yderligere er benyttet som tabsfunktionen til træning af de ikke-lineære modeller. RMSE beskriver standardafvigelsen i residualerne som beskriver hvor langt regressionslinjen er fra de observerede værdier. RMSE ligger høj vægt på større afvigelser og mindre vægt på små afvigelser, grundet det kvadrerede forhold. RMSE kan forklares som en kvaderet udgave af MSE og beskrives på ligningsform som:

$$RMSE = \sqrt{MSE} = \sqrt{\sum_{t=1}^{n} \frac{(y_t - \widehat{y_t})^2}{n}} \tag{30}$$

Hvor y_t beskriver den faktiske værdi i tiden t, $(\widehat{y_t})$ beskriver den prædikterede værdi i tiden t og n angiver antallet af prædikteringer (Enders, 2015).

5.2 Mean Absolute Percentage Error

Mean Absolute Percentage Error herefter benævnt, MAPE, måler, som navnet også hentyder til, den absolutte forskel i procent mellem den observerede værdi y_t i tiden t og den faktiske værdi $\widehat{y_t}$ i tiden t. Formålet med at udregne MAPE skyldes at den vægter afvigelserne ligeligt uanset størrelsen. Derudover at udregne gennemsnitsafvigelsen mellem forcastede og observerede værdier i procent, så kan statistikken også benyttes som et værktøj til at evaluere de afhængige variable på tværs, i

specialets tilfælde, de forskellige udvalgte finansielle aktiver. MAPE kan matematisk beskrives på ligningsform som:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \left| \frac{y_t - \widehat{y_t}}{y_t} \right| \tag{31}$$

(Enders, 2015)

5.3 Pesaran Timmermann

Som et supplement til at teste for hvilken model der er mest præcis i sin prædiktering baseret på førnævnte evalueringskriterier, så kan der også testes for modellerne evner til at forudsige retningen, i specialets tilfælde, om hvorvidt modellerne er i stand til at binært forudsige om aktiekursen vil stige eller falde, og testen giver dermed en belysning af hvor gode modellerne er

hver især til at afgøre dette, hvor der ikke tages udgangspunkt i hvor langt de prædikterede værdier er fra den observerede værdi, men blot retningen derpå. Denne test blev udviklet i 1992 af Pesaran og Timmermann. Testen kan anvendes på ikke-stationære normalfordelinger så længe y_t og x_t er symmetrisk fordelt omkring nul. Tages der udgangspunkt i en tidsserie y_t og et forecast af tidsserien x_t så kan testen defineres på ligningsform som følgende:

$$S_n = \frac{\hat{P} - \widehat{P_*}}{\left[v\hat{a}r\left(\hat{P}\right) - v\hat{a}r\left(\widehat{P_*}\right)\right]^{\frac{1}{2}}} \sim N(0,1) \tag{32}$$

Hvortil følgende ligninger anvendes til at beskrive ovenstående ligning:

$$\hat{P} = n^{-1}\sum_{t=1}^{n} Z_t(y_t, x_t) \tag{33}$$

$$v\hat{a}r\left(\hat{P}\right) = n^{-1}\widehat{P_*} \cdot (1 - \widehat{P_*}) \tag{34}$$

$$\widehat{P_*} = \widehat{P_y}\widehat{P_x} + (1 - \widehat{P_y})(1 - \widehat{P_x}) \tag{35}$$

$$\hat{var}(\hat{P}_*) = n^{-1}(2\hat{P}_y - 1)^2\hat{P}_x(1-\hat{P}_x) + n^{-1}(2\hat{P}_x - 1)^2\hat{P}_y(1-\hat{P}_y) + 4n^{-2}\hat{P}_y\hat{P}_x(1-\widehat{P_y})(1-\hat{P}_y) \tag{36}$$

$$\hat{P}_y = n^{-1}\sum_{t=1}^{n} Z(y_t) \tag{37}$$

$$\hat{P}_x = n^{-1}\sum_{t=1}^{n} Z(y_x) \tag{38}$$

$$Z(\cdot) = \begin{matrix} 1\ hvis \cdot > 0 \\ 0 = ellers \end{matrix} \tag{39}$$

I ovenstående ligning er henholdsvis $\hat{P}_y$ og $\hat{P}_x$ sandsynligheden for at y_t og x_t er større end nul. $\hat{P}$ beskriver derfor den andel gange at de observerede værdier og prædikterede værdier har haft samme retning. $\hat{P}_*$ kan defineres som sandsynligheden for at y_t og x_t er større end nul adderet med sandsynligheden for, at de ikke er større end nul. (Pesaran & Timmermann, 1992)

5.4 Diebold-Marino

Udover at kigge på modellernes forecastings performance ved afvigelsen mellem prædikterede og faktiske værdier, så anvendes der en Diabold-Marino test i specialet. DM (Diebold-Marino), testen anvendes som et redskab til at belyse hvorvidt der er signifikant forskel i accuracy, altså præcisionen, på tværs af modellerne. Eftersom specialet kun benytter one-step ahead forecasts, så vil DM-testen blive beskrevet ud fra et one-step ahead forecast perspektiv.

I en traditionel tabsfunktion så kan differencen fra periode i, ved anvendelse af model 1 mod model 2 beskrives som, $d_i = g(e_{1i}) - g(e_{2i})$ og middelværdiens tabsfunktion på tværs af modellerne kan findes ved:

$$\bar{d} = \frac{1}{H}\sum_{i=i}^{H}[g(e_{1i}) - g(e_{2i})] \tag{40}$$

Hvor H = antallet af forecast.

Såfremt man antager at $h_0 = Equal\ forecast\ accuracy$, så vil $\bar{d}$ = 0, hvor det antages at $\bar{d}$ følger en normalfordeling og derved forventes det ikke at modellernes prædikteringsfejl i sig selv følger en normalfordeling. Hvis vi kender variansen af $\bar{d}, altså\ var(\bar{d})$, så kan der opstilles et forholdet $\frac{\bar{d}}{\sqrt{var(\bar{d})}}$ hvorefter der kan testes nulhypotesen ved hjælp af en t-test med H-1 frihedsgrader. Variansen af $\bar{d}$ kan estimeres ved:

$$var(\bar{d}) = [\gamma_0 + 2\gamma_1 + \cdots + 2\gamma_q](H-1)^{-1} \tag{41}$$

Hvor γ_i denoterer den i'te autokovarians ved sekvensen d_t hvis man antager den første værdi q, ved kovarians y_i er forskellige fra nul, så vil der kunne forekomme et scenarie hvor y_i er negativ og derved vil standardafvigelsen af $\bar{d}$ blive negativ hvortil Harvey et al, argumenterede for brugen af den kvaderede standardafvigelse og dermed konstruerede Diebold-Marino test statistikken kan udregnes og angives på ligningsform således:

$$DM = \frac{\bar{d}}{\sqrt{\frac{[\gamma_0 + 2\gamma_1 + \cdots + 2\gamma_q]}{(H-1)}}} \tag{42}$$

(Enders 2015)

6 Metodologi og data

Afsnittet vil præsentere det data som danner fundamentet for de empiriske resultater og behandlingen deraf. Dataet vil blive beskrevet både grafisk og vha. en deskriptiv statistik.

6.1 Datagrundlag

Datasætter som danner udgangspunkt i konstruktionen af både de lineære og ikke-lineære modeller tager afsæt i finansielle aktiver som indgår i S&P500-indekset. S&P 500 indekset følger de 500 største amerikanske virksomheder. S&P blev oprindeligt introduceret i 1959 af Standard & Poor. For at indgå i S&P 500 er der nogle kriterier der skal opnås. Det kræver at virksomheden er lokaliseret i USA samt have en markedsværdi på 13,1 billion dollars. Ydermere skal mindst 50% af virksomhedens aktier være tilgængelig for offentligheden og aktieprisen skal minimum være en dollar per aktie. Mindst halvdelen af virksomhedens indtægter skal være fra USA og derudover skal virksomhederne have mindst 4 kvartalers positive indtjening (K, Amadeo. 2021). Specialet har derfor valgt 3 vækstaktier der indgår i S&P 500 indekset og de er valgt baseret på deres Price-Earning ratio, hvortil aktierne skal have en højere Price-Earning ratio end selve indekset de er fra. Derudover er der valgt aktiver fra forskellige sektorer, og dataet er hentet ned vha. Rstudio, direkte fra Yahoo Finance hvor alle finansielle aktiver har en ticker, som er et symbol der kendetegner aktien, eksempelvis er $AMZN den finansielle ticker for Amazon.com aktien.

***Amazon.com ($AMZN)** Er en Amerikansk internethandelsvirksomhed med hovedsæde i Seattle

*** American Tower ($AMT)** American Tower Corporation er en amerikansk ejendomsinvesterings fond med hovedsæde i Boston. American Tower Corporation er derudover også ejer af og operatør af trådløs radiokommunikationsinfrastruktrer i flere lande over hele verden.

*** Merck & Co. ($MRK**) Merck & Co Inc. Anses som værende et forsknings-drevet medicinalselskab. Merck & Co. er blandt verdens største medicinalfirma koncerner med hovedsæde i New Jersey.

6.2 Vækstaktier og P/E Ratio

For at forstå hvad der menes når der bliver nævnt vækstaktier, så arbejdes der ud fra følgende definition fra Corporate Finance Institute

"Growth Stocks are stocks that come with a substantially higher growth rate compared to the mean growth rate prevailing in the market. It means that the stock grows at a faster rate than the average stock in the market, consequently generating earnings at a faster rate"- CFI

Vækstaktier kan altså defineres som aktier, der vækster højere end gennemsnittet i det marked de befinder sig i, hvormed den hurtigere vækstrate må formodes at medføre en højere indtjening. Vækstaktier er ofte forbundet med at være relativt dyre aktier, grundet deres vækstende natur, hvilket kan medføre, at de vil være korrekt placeret i markedet, hvor fordelen ved at investere i vækstaktier ligger i, at man forudsætter at de fortsætter deres vækst hvormed man så opnår en bedre indtjening. Fordelene ved vækstaktierene fremfor deres modsætning, som er valueaktierne er, at de er relativt mindre risikable, da de vurderes til at være mere robuste overfor økonomiske udsving. Måden hvorpå man kan afgøre om det er en vækst eller value-aktie findes ved bestemte finansielle nøgletal, hvor et af dem, som specialet tager udgangspunkt i er P/E (Price/Earning) ratioen (CFI).

P/E ratioen kan defineres som et nøgletal der beskriver virksomhedens aktuelle aktiekurs i forhold til deres indtjening per aktie og kan skrives på ligningsform som:

$$\frac{Price}{Earning} = \frac{P}{E} = \frac{Markedsværdi\ pr\ aktie\ (aktiekurs)}{Indtjening\ pr\ aktie} \tag{43}$$

Der findes naturligt mange forskellige nøgletal, porteføljesammensætning og strategier når det kommer til investering, men næsten 80% af alle investorer adspurgt af Bank of America benytter sig af P/E-ratioen og anses som værende det førende anvendte nøgletal til udvælgelsen af aktier, ydermere har vækstaktierne domineret aktiemarkederne hvorimod valueaktierne med lave P/E values har underpræsteret siden 2010 (Fitzgerald, 2019). Til trods for P/E ratioen er relativt simpel at udregne så skelnes der mellem tre forskellige måder at vurdere price/earnings på.

***TTM** – Trailing Twelve Months som navner også antyder, så vurderes virksomhedens indtjening baseret på de sidste 12 måneder hvor fordelen her er, at der benyttes historisk data for forrige år og er en af de hyppigste anvendte metoder. Hvilket også er afhandlingens fremgangsmåde til at udregne P/E værdien for aktierne.

***Forward Earnings**: Her udregnes det baseret på et estimat af virksomhedens fremtidige indtjening.

***Shiller P/E** Benytter sig af gennemsnitlig indtjening over en periode, ofte over en periode på 10 år hvormed der justeres for inflation.

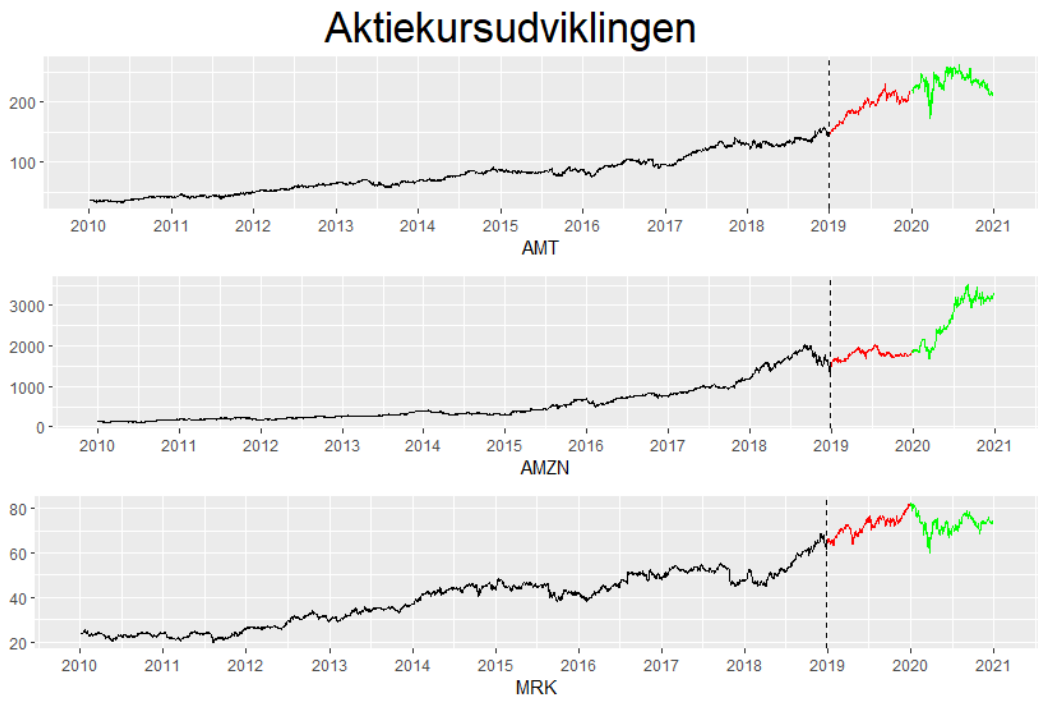
For at aktierne skulle defineres som vækstaktier sammenlignes de med et benchmark, som ofte er anvendt indenfor investering, hvor dette benchmark altså fungerer som et sammenligningsgrundlag for virksomhedens performance, hvor det altså forventes at de udvalgte aktiver skal have en P/E værdi som er højere end benchmarket. Eftersom fokuspunktet af de udvalgte aktier er baseret på USA's aktiemarked og de alle indgår i S&P 500, så virker dette, som det umiddelbare naturlige valg som benchmark.

Tabel 1: P/E for udvalgte aktiver

Company	Amazon.com	Merck & Co	American Tower Corporation	S&P 500
P/E Ratio	91,11	27,52	86,97	23,27

Note: Price Earning værdier for de udvalgte aktier, egen fremstilling baseret på R koderne

Figur 12: Aktiekursudviklingen for udvalgte aktiver



Note: x-aksen = datoer, y-aksen = aktiekursen, Egen fremstilling baseret på R koderne

Til trods for at specialet ikke benytter sig direkte af porteføljesammensætninger er der stadig forsøgt at skabe spredning i udvalget af aktier, da man normalt som investor forsøger at skabe spredning i sin portefølje igennem diversificering af aktiver, jf. Harry Markowitz moderne porteføljeteori, så er aktierne udover P/E værdien forsøgt at blive udvalgt med det kriterie, at ikke har samme sektortilhørsforhold. Som det kan ses på figur 12, så er der markeret tre forskellige farver, hvor dette indikerer datasættets opdeling hvoraf perioden forløber sig fra 2010-01-04 til 2020-12-29 hvoraf de første 8 år, det som er markeret med sort, indikerer træningssættet, hvor den bedste model for henholdsvis ARIMA og LSTM modellen bliver fundet, hvorefter de bliver trænet på et testdata som forløber sig fra 2019-2020 og slutteligt forsøges et forecast af aktiekursen fra 2020-2021.

Datasplittet er derfor delt op i 2766 observationer for hele perioden, hvoraf test og forecast træningssættene er delt op i hhv. 251 og 252 observationer hvoraf dette efterlader 2263 observationer til træningsdatasættet. Denne metode hvor man opdeler datasættet på og tester modellens prædikteringsevne kendetegnes som en out-of-sample strategi som vurderes som værende mere troværdig, da modellerne skal forsøge, at forudsige kursen ud fra data de ikke har set før, hvilket gør det mere virkelighedstro (Eurostat, 2021).

Dataet som anvendes er dagligt data af aktiverne hvor adjusted closing price er valgt, som er den justeret aktiepris, og begrundelsen for valget af denne skyldes, at der i specialet ikke anvendes koder til at tage højde for dividender og andre transaktionsomkostninger, hvorfor adjusted closing price virker som den optimale til besvarelsen af afhandlingens problemformulering. Der afviges i specialet for at udføre LSTM og ARIMA forecasts på selve aktieindekset da aktier fluktuerer og har et forskelligt forhold, hvor der håbes at give modellernes forecast performance et mere realistisk afbillede af hvor anvendelige de er ift. udvælgelsen af aktiver samt forudsige aktiekursen.

Eksempelvis kan Amazons standardafvigelse forekomme høj og urealistisk for et indeks hvortil MRK og AMTs minimum og maksimum værdi forekommer relativt lave, selv sammenlignet med AMZN, hvortil det ikke giver et sandfærdigt afbillede såfremt man valgte at inddrage et indeks. Middelværdien er markant højere end MRK og AMT hvilket blot skyldes at aktiekursen er markant højere det samme udtryk gør sig gældende for standardafvigelsen af Amazon hvilket blot tyder på, at de har haft en høj aktiekursudvikling, det giver dog ikke et udtryk af trends der må forekomme i tidsserierne.

Skewness indikerer hvorvidt fordelingen af observationerne er fordelt omkring middelværdien, hvor værdien nul vil antages at være normalfordelt, det ses at samtlige har en positiv skewness hvilket

betyder de må være højreskæv, hvilket også grafisk kan ses i figur 12. Kurtosis giver indtryk af halernes fordeling hvor en positiv kurtosis tyder på at der forekommer mange outliers i tidsserien. Jarque Bera værdien benyttes som en goodness-of-fit test for at se hvorvidt tidsserien er normalfordelt, hvilket forekommer tydeligt da værdien af JB-testen er 2,2e-16 for samtlige aktiver hvilket betyder at H0 om normalfordeling forkastes, hvilket bekræftes af både skewness, kurtosis og grafisk på figur 12.

6.3 Deskriptiv statistik

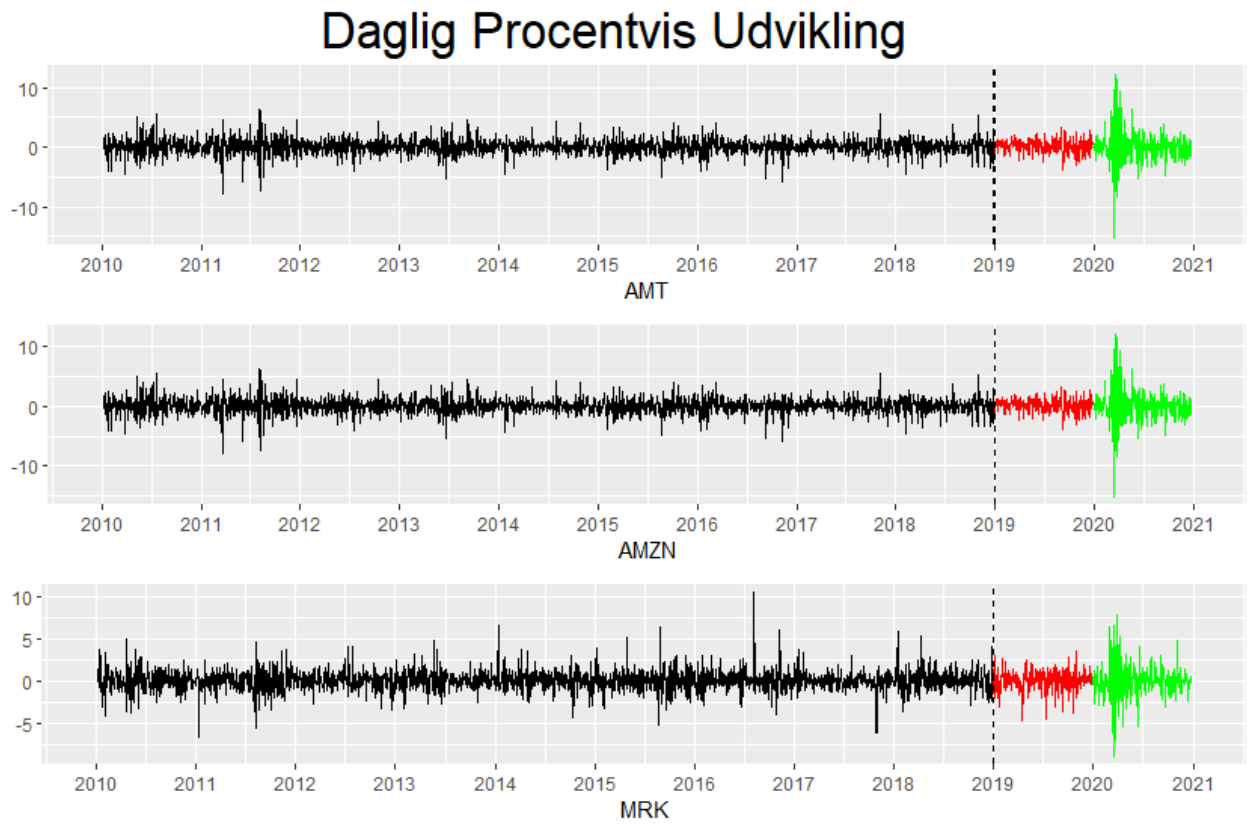
Tabel 2: Deskriptiv Statistik

	AMZN	MRK	AMT
Middelværdi	856,73	44,89	104,04
Standardafvigelse	813,20	16,92	60,25
Maksimum	3531,45	82,14	263,28
Minimum	108,61	20,04	32,27
Skewness	1,29	0,40	1,02
Kurtosis	0,87	-0,85	-0,03
Jarque Bera	2,2e-16	2,2e-16	2,2e-16
ADF P-værdi	0,01	0,01	0,01

Note: Viser beskrivende statistik af datasætte. Egen fremstilling baseret på R koderne.

ADF-testen for datasættet havde høje værdier omkring 40-80% for samtlige aktiver og dette ville medføre komplikationer, da det betyder at datasættet ikke vil være stationær, hvilket som beskrevet i afsnit 3 er en nødvendighed for ARIMA-modellen. Eftersom stationaritet er en nødvendighed for den lineære regressionsmodel, og eftersom der altid vil være trend i tidsseriedata der er baseret på aktier, hvilket kan argumenteres for, da aktier generelt stiger over længere perioder og derfor må have en positiv trend og derfor vil kriterierne for stationaritet ikke være opfyldt. Derfor blev datasættet differentieret hvorefter ADF-testen foretages igen hvorefter samtlige p-værdi er 0,01 hvortil der må konkluderes at såfremt tidsserien, er differentieret så er der stationaritet.

Figur 13: Daglig Udvikling i Procent



Note: x-aksen = datoer, y-aksen = aktiekursen. Træningsperiden er sort, rød er træningsperioden og grøn er forcastperioden. Egen fremstilling baseret på R koderne.

Figur 13 illustrerer tidserien efter at være differentieret af første orden, hvor det daglige afkast er omregnet til procent. Datasættet er vist splittet op som figur 12 hvor den sorte linje markerer træningsdatasættets forløb, den røde illustrerer testperioden og den grønne viser prædikteringsperioden. De største daglige ændringer forekommer i prædikteringsperioden, hvor alle aktier udviser høj volatilitet, hvilket kunne mistænkes at skyldes den globale pandemi, Corona-perioden, som tidsperioden forløber sig i. Hvilket må formodes vil skabe større usikkerhed og for nogle virksomheders tilfælde konkurs. Dette kunne medføre at modellernes evne bliver svækket da perioden de skal forecaste på, afviger fra de forrige to perioder.

6.4 Rolling Window

I empirisk analyse kan det forekomme, at der ikke er nok data tilgængeligt, hvilket kan medføre komplikationer hvortil et Rolling Window (rullende vindue) kan benyttes for at imødekomme dette problem. Ofte så kræver sofistikerede modeller meget data med mange parametre der skal modelleres, hvilket medvirker til nødvendigheden for et større datasæt. Rolling Window metoden er hyppigt anvendt i praksis, til trods for manglende metoder ofte er negligeret i litteraturen. RW, (Rolling Window), metoden benyttes til test og validering af opstillede modeller, ofte når der forekommer tidsserier. Når der skal opstilleres modeller og analyseres på finansielt data, så er de hyppige antagelser ofte at parametrene for modellen er tidsuafhængige, som det også er tilfældet ved ARIMA-modellen. Dette kan forekomme urealistisk og nær utopisk, eftersom de økonomiske omgivelser ændres, så må parametrene også ændres for at tilpasse sig den nuværende økonomiske situation. For at imødegå dette problem kan RW benyttes, hvor hensigten er at estimere modellens parametre ved at benytte sig af et rullende vindue der går igennem en fast størrelse af stikprøven. Såfremt parametrene er konstante over tid, så vil estimatet for det rullende vindue ikke ændre sig markant, men hvorimod ændres under valideringen af stikprøven, så vil estimatet af det rullende vindue opfange ustabiliteten og dermed opfange det og ændre sit estimat derefter. Der er i specialet anvendt rullende vinduer både til LSTM og ARIMA-modellerne. (Kidbrooke Advisory, 2017)

7 Modeludvælgelse

Præsenterer den endelige opsætning af de lineære og ikke-lineære modeller, som er anvendt med henblik på bedst muligt at kunne prædiktere one-step ahead forecast på de tre udvalgte aktiver.

7.1 ARiMA

For at udvælge den optimale ARIMA model baseres det på AIC som beskrevet i afsnit (X). Modellen bliver trænet med én afhængig variabel, den justerede aktiepris, hvorefter datasættet bliver splittet op, som beskrevet i afsnit (X). Hvor træningssættet benyttes til at træne modellen for at finde den model som har den laveste AIC, som klaret i afsnit (X), der er i tabel 7.1 udvalgt de bedste modeller, der kan yderligere henvises til bilag A. for et overblik over samtlige modeller med tilsvarende AIC værdier. AMTs bedste model er en ARIMA(1,1,1)* hvor * indikerer det er en ARIMA model med drift, hvor både AMZN og MRKs bedste AIC score findes ved en ARIMA(0,1,0)*, hvilket betyder at der i tidsserien ikke forekommer AR eller MA-led men derimod kun en koefficient i form af driften.

Tabel 3: AIC for ARIMA

ARIMA	AMT	AMZN	MRK
Arima(1,1,1)*	**6598.58**	18495.99	3127.816
Arima(0,1,0)*	6608.191	**18493.92**	**3124.787**

Note: Laveste AIC-værdier for ARIMA-modellen, hvor den laveste er markeret med fed. Egen fremstilling baseret på R koderne. Se Bilag A under appendix for den AIC værdier for samtlige ARIMA modeller.

Efter modellen er blevet udvalgt baseret på AIC kan out-of-sample strategien benyttes vha. testdatasættet som vil blive anvendt til at udvælge den model som opnår den laveste RMSE/MAPE værdi. Testen for de laveste RMSE og MAPE værdier vises i tabel 4. Hvor der er afrundet til tre decimaler og fire decimaler såfremt tallene skulle være ens, hvilket bliver gjort for at vise forskellen i RMSE og MAPE-værdierne. Såfremt der skulle forekomme et scenarie hvor den ene model har en lavere MAPE men samtidigt højere end RMSE, så afgøres dette vha. tabel 3 som viser AIC, hvor modellen med laveste AIC vil blive anvendt til forecasting af aktiekursen, dette gør sig eksempelvis gældende for AMT som har en lavere RMSE i ARIMA(1,1,1)* men samtidigt en højere MAPE sammenlignet med ARIMA(0,1,0)* hvor der vil blive valgt ARIMA(1,1,1)* da den har en lavere AIC

værdi. Ligeledes vil AMZN blive valgt baseret på AIC-værdien grundet forholdet mellem RMSE og MAPE.

Tabel 4: RMSE og MAPE for ARIMA

	AMT		AMZN		MRK	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
ARIMA(1,1,1)*	**2,337**	0,896	25,236	**1,0635**	**0,832**	**0,8630**
ARIMA(0,1,0)*	2,343	**0,894**	**25,251**	1,0637	0,833	0,8633

Note: RMSE og MAPE værdier for ARIMA modellerne, laveste værdier markeret med fed. Egen fremstilling baseret på R koderne

Baseret på tabel 3 og 4 kan ARIMA modellens endelige struktur sammenfattes, AMT vil benytte sig af en ARIMA(1,1,1)* model, da RMSE forekommer lavt kombineret med den laveste AIC værdi, hvilket betyder der i modellen vil være både et AR og MA led og derudover være differentieret af første orden, for at opnå stationaritet. AMZN og MRK vil begge benytte af en ARIMA(0,1,0)*, hvilket kendetegnes som en random walk med en drift, hvor den eneste koefficient bliver driften, begrundelsen for valg af ARIMA model bliver også yderligere understøttet af RMSE og MAPE værdier i tabel 4.

7.2 LSTM Modelkonstruktion

Når der skal konstrueres den ’perfekte’ LSTM model, så er den største fordel også den største ulempe. Modellen har et enormt potentiale på nær uendeligt med muligheder, begrænsningen ligger blandt andet i computerkraft men også at der ikke er konsensus om det perfekte antal af neuroner, lag osv. I forsøget på at jagte den optimale model følges samme fremgangsmåde som ved ARIMA-modellen hvor LSTM-modellerne oprindeligt blev trænet på træningsdataet, hvor der trænes 150 modeller for hver af de udvalgte aktier for henholdsvis en 5-sekvens og 10-sekvens datalængde, hvilket vil være tilsvarende at benytte sig af lags i ARIMA-modellen. Tuningen af modellerne medfører at 900 modeller er blevet trænet og derefter bliver testet på testdatasættet hvorefter der udvælges den model med laveste RMSE og MAPE-værdi som så benyttes til forecasting.Hvor det forventes at 10-sekvens modellen vil præstere bedst, grundet LSTM’s struktur men også set ud fra et teoretisk synspunkt som

beskrevet i afsnit 4. Tabel 5 viser RMSE og MAPE-værdierne for den bedste LSTM 5-sekvens og 10-sekvens model.

Tabel 5: RMSE og MAPE Værdier

	AMT		AMZN		MRK	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
5-sekvensdata	7,653	4,817	126,56	7,428	4,31	**1,904**
10-sekvensdata	**6,465**	**2,985**	**113**	**4,581**	**4,271**	1,967

Note: RMSE og MAPE værdier for LSTM modellerne. Egen fremstilling baseret på R koderne.

Grundet at opsætningen af LSTM-model hvor der indgår neuroner, lag, dropout, aktiveringsfunktioner som blandt andet beskrevet i afsnit 4, så vil der forsøges, i kontrast til tidligere eksisterende litteratur, at anvende en mere automatiseret tilgang til det, at optimere modellen, hvilket sker ved først og fremmest at finde de optimale hyperparametre ved brug af KerasTuner[1] som er en pakke i R, hvor der i den eksisterende litteratur hidtil har manuelt forsøgt optimere parametrene, hvilket har resulteret i få tuning af parametre, så vil dette foregå ud fra pre-definerede værdier for hyperparametrene hvor algoritmen forsøger at optimere LSTM-modellen indenfor dens mulige hyperparameter framework.

Hyperparameterne er afgørende for resultatet, som også forklaret i afsnit 4.6, da de medvirker til at styre adfærdsmønstret i modellen. Størrelsen af batch size er faldet på 64, hvilket stemmer overens med (Goodfellow et al, 2016) der argumenter for at batch størrelser i potenser af to præsterer bedst, hvor 16-32 batches primært bliver anvendt til store modeller. Den gængse aktiveringsfunktion som anvendes til samtlige LSTM-modeller vil være ReLu-funktionen som beskrevet i afsnit4.3.1. Til at bestemme selve modellens parametre benyttes der en hyperband funktion som kan anses som værende en algoritme der automatiserer processen at finde den optimale model, dette foregår ligeledes indenfor et parameter-interval der bestemmes af modelkonstruktøren. Antallet af epoker under Hyperband afgør hvor mange gange hver model skal trænes før den går videre og tester den nye model, måden

[1] O'Malley, Tom and Bursztein, Elie and Long, James and Chollet, Fran C and Jin, Heifeng and Invernizzi, Luca & Others - 2019

dette foregår på, er i specialets tilfælde, et valg mellem 32-600 neuroner, en dropout rate på hhv. 10,20 og 30%. Hvor tabsfunktionen er MAPE. Derudover for at overskueliggøre modellerne så anvendes der kun modeller med 1 til 3 lag. Ydermere eftersom 10-sekvens modellen outperformede 5-sekvens modellen for samtlige aktier, hvormed MRK modellens 5-sekvens var den som kom tættest på performance af 10-sekvens, hvilket også ses i tabel 5, derfor vil 10-sekvens modellen være den udvalgte model til prædiktering. Det er vigtigt at tydeliggøre at følgende optimale LSTM-konstruktion kun anses som værende optimal for det anvendte datasæt, samt de estimerede parameter og anvendte hyperparamtre.

Figur 14: LSTM struktur for AMT

Den bedste konstruktion for AMT består af et LSTM lag, to dense lag, hvor et dense lag er det samme som et FNN lag som er forklaret i afsnit 4, og et dropout lag på 10%.

Figur 15: LSTM Struktur for AMZN

Den bedste konstruktion for AMZN består af et LSTM lag, to dense lag samt et dropout lag på 10%.

Figur 16: LSTM Struktur for MRK

Den bedste konstruktion for MRK består af et LSTM lag, to dense lag, hvor et dense lag er det samme som et feedforward lag som er forklaret i afsnit 4, et dropout lag på 10%.

Modellerne virker overordnet til at følge samme struktur hvor forskellen ligger især ved antallet af neuroner for hver model, hvor AMZN også skiller sig ud ved at have et dropout på 20% modsat de andre som kun har et 10% dropout. Derudover kan det virke iøjefaldende at samtlige modeller valgte det maksimale antal af lag i modeller, hvilket kan skyldes at modellen, grundet dens få lag kan fremstå grådig ift. at optimere aktiviteterne i det skjulte lag for at opnå den laveste tabsfunktion (Bengio et al,. 1994).

8 Empiriske resultater

Dette afsnit præsenterer de empiriske resultater med udgangspunkt i afhandlingens forudgående teoretiske samt metodologiske fundament

For at skabe et sammenligningsgrundlag er der i afhandlingen udført forskellige tests for at have nogle indikatorer på hvilke modeller, der har den bedste performance ift. at forudsige aktiekursen for de udvalgte vækstaktier. Formålet er at skabe et retvisende billede og udnævne en klar vinder af de lineære og ikke-lineære modeller. I tabel 6 ses evalueringskriterierne opstillet i afsnit 5. Hvor der er blevet udregnet en RMSE og MAPE-værdi for henholdsvis ARIMA og LSTM modellen. AMT aktien virker iøjnefaldende da både RMSE og MAPE virker ekstreme for ARIMA, hvilket kunne tyde på, at modellernes prædikteringsevne er svag. For Amazon aktien virker der til at være et væsentlig skel mellem ARIMA og LSTM-modellens performance, hvor det er tydeligt at LSTM modellen har klaret sig bedst. For MRK aktien virker det til, at der også er relativt tæt mellem RMSE og MAPE, for ARIMA og LSTM værdierne, især sammenlignet med AMT, men det er dog stadig tydeligt, at LSTM forekommer som den klare vinder. Den overordnede konsensus må derfor være at LSTM-modellens RMSE og MAPE-værdier tilsiger at LSTM-modellen mod alt forventning laver mindre fejl i dens prædiktering, hvilket også vil blive yderligere drøftet i efterfølgende diskussionsafsnit. Eftersom der er anvendt daglige data, hvor der forekommer hyppige observationer, så virker værdierne ekstreme, især for AMT aktien.

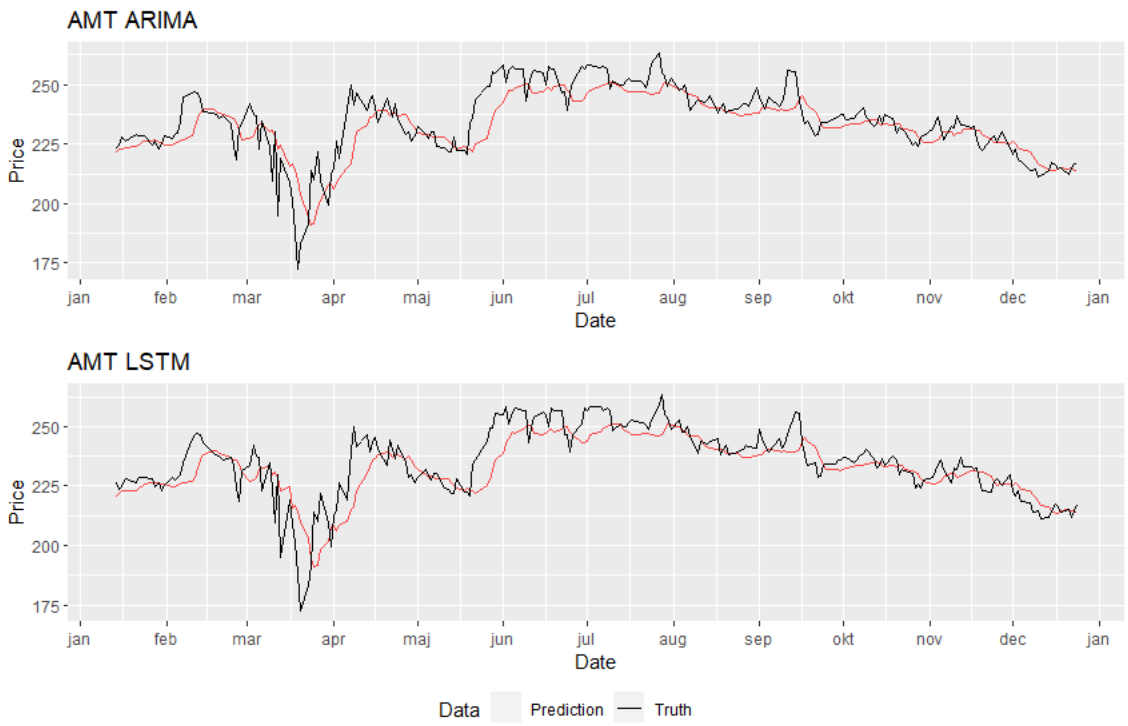
Tabel 6: RMSE og MAPE værdier for prædikteringsperioden

	RMSE		MAPE	
	Arima	LSTM	ARIMA	LSTM
AMT	15,910	**8,695**	4,817	**2,654**
AMZN	319	**215**	9,42	**6,18**
MRK	3,578	**2,051**	3,793	2,218

Note: Viser RMSE og MAPE værdierne i forecastingsperioden 29-12-2019 til 29-12-2020, laveste værdier markeret med fed. Egen Fremstilling baseret på R koderne.

Der er i efterfølgende figur 17-19 givet et grafisk overblik over modellernes prædikteringsevner for både LSTM og ARIMA-modellerne for alle tre aktier.

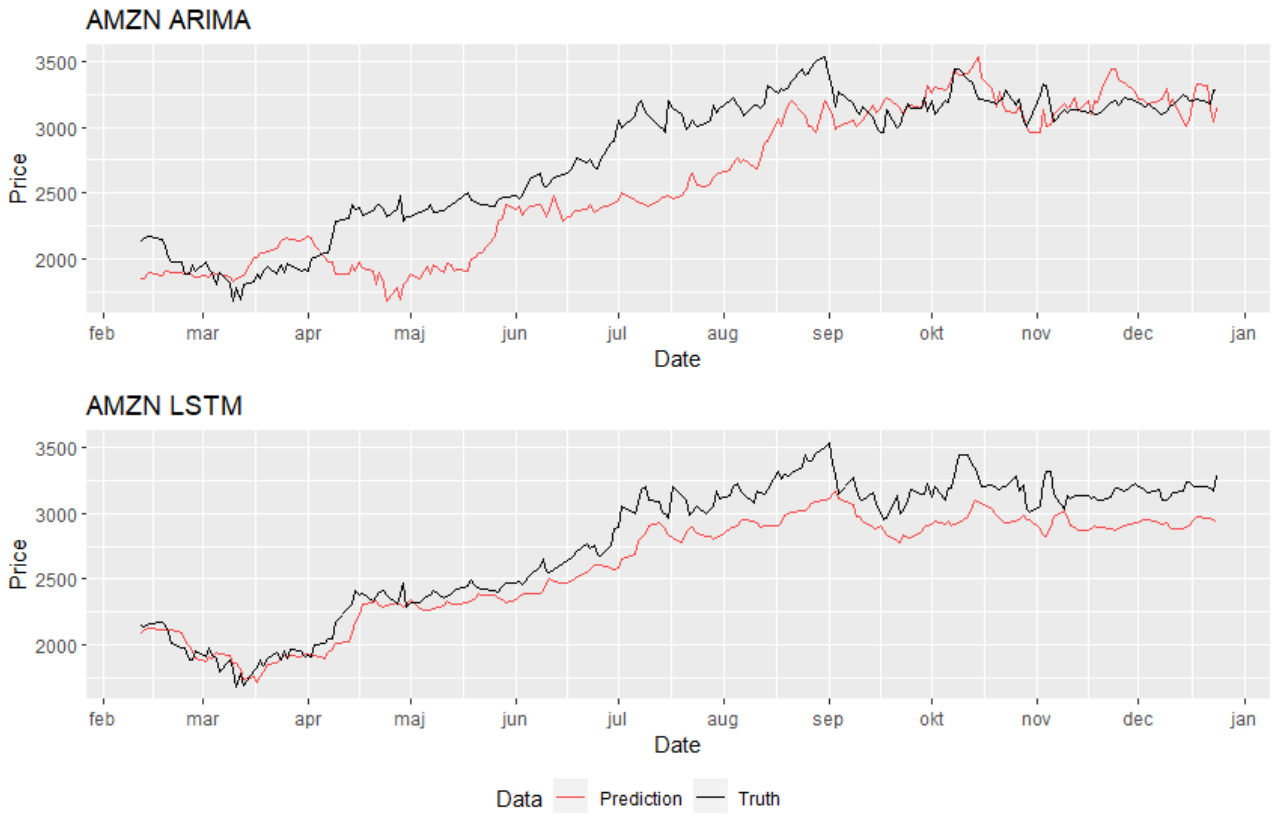
Figur 17: Forecast for AMT



Note: x-aksen = datoer, y-aksen = aktiekursen, observerede værdier er den sorte linje og prædikterede værdier den røde. Egen fremstilling baseret på R koderne.

Som også understøttes af tabel 6 så virker både LSTM og ARIMA modellen til at prædiktere kursen i nogen grad, hvor det ses at de generelt følger kursen, dog med det forbehold at det virker til at det forekommer tydeligt at perioderne er lagged og derfor virker til hele tiden at være steps bagud i forhold til den observerede aktiekurs. Der forekommer dog nogle udsving i aktiekursen og volatilitetet som ingen af modellerne opfanger.

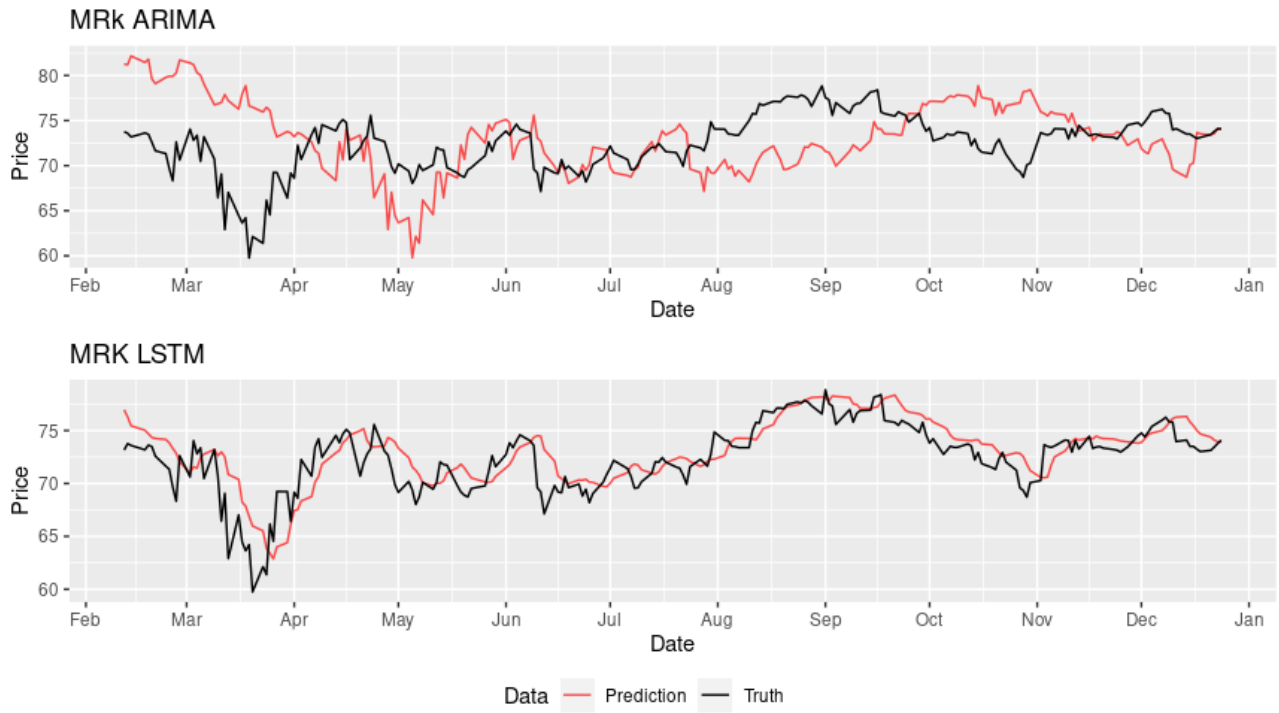
Figur 18: Forecast for AMZN



Note: x-aksen = datoer, y-aksen = aktiekursen, observerede værdier er den sorte linje og prædikterede værdier den røde. Egen fremstilling baseret på R koderne.

Som også understøttes af tabel 6 så virker det til at LSTM-modellens prædikteringsevne er overlegen sammenlignet med ARIMA-modellen, hvilket ses tydeligt på figur 18, at ARIMA modellens forcastede værdier, virker til, at være langt fra de observerede værdier og kun har prædikterede værdier der er tilnærmelsesvis tæt på fra oktober til december, hvorimod det virker til at LSTM-modellen gætter for konservativt i sit forecast og derved altid befinder sig under kurven for de observerede værdier.

Figur 19: Forecast for MRK



Note: x-aksen = datoer, y-aksen = aktiekursen, observerede værdier er den sorte linje og prædikterede værdier den røde. Egen fremstilling baseret på R koderne.

Det grafiske billede af forecastet for MRK illustrere noget der minder lidt om AMZN-aktien, hvorimod dens bedste forecastingsperiode sker fra juni til august. Det virker tilsyneladende til at ARIMA-modellen i næsten hele perioden, at være langt fra de observerede værdier med undtagelse af juni til august, hvilket stemmer overens med tabel 6. Baseret på tabel 6 burde modellernes prædikteringsevner være relativt tætte, hvorimod det ses tydeligt at LSTM outperformer ARIMA modellen i høj grad, eftersom LSTM i nogen grad føler de observerede værdier, dog virker modellen til at være for langsom til at fortage ændringer og derfor altid halter efter de observerede værdier.

Tabel 7: Diebold-Marino Test

H_0: Hypotese	AMT	AMZN	MRK
M1=M2	0,000	0,000	0,000
M1>M2	0,000	0,000	0,000
M2<M1	1	1	1

Note: Viser resultatet fra Diebold-Marino testen, M = Model, LSTM-modellen er benchmark (M1) modellen. Egen fremstilling baseret på R koderne.

Ovenstående tabel 7 som fremsætter Diebold-Marino testen virker ret sandsynlig og bliver yderligere valideret af tabel 6 samt figur 17-19 hvor der baseret på RMSE og MAPE-værdierne samt den grafiske oversigt at LSTM overordnet virker bedre i sin prædiktering. Der bliver for samtlige aktier testet hvorvidt modellernes forecast errors, altså prædiktionsnøjagtighed er ens, hvormed samtlige h_0 hypoteser afvises, og derfor kan der konkluderes at der er statistisk belæg for at den ene model tilsyneladende er bedre end den anden. Hvorefter der testes for hvorvidt Model 1 (benchmark) modellen har højere forecastingsfejl end Model 2 (ARIMA). Dette forkastes i samtlige hypoteser, og derfor må man næsten antage at det eneste mulige scenarie er, at model 2 er den bedste model, hvilket slutteligt testes for, hvormed der forekommer en P-værdi på 1, altså 100% hvormed vi med alt sandsynlighed, baseret på testen kan konkludere at LSTM modellen har lavere prædikteringsfejl ned ARIMA modellen.

Tabel 8: Pesaran Timmerman Test

	AMT		AMZN		MRK	
H0: Gætter Retning	ARIMA	LSTM	ARIMA	LSTM	ARIMA	LSTM
P-værdi	0,000	0,000	0,000	0,000	0,000	0,000

Note: Viser resultatet fra Pesaran Timmerman testen. Egen fremstilling baseret på R koderne.

Tabel 8 viser resultatet af Pesaran Timmerman testen hvor hensigten er at teste modellernes evne til at prædiktere retningen i binær form, altså om de med nogen sandsynlighed kan vurdere hvorvidt aktiekursen for de udvalgte vækstaktier vil stige eller falde, hvor der i afhandlingen er benyttet one-step ahead forecast, så derfor lyder testen på om de kan forudsige aktiekursen for den efterfølgende dag. Tabel 8 viser et tydeligt resultat hvormed at samtlige H_0 forkastes, hvilket betyder, at til trods for at LSTM-modellen baseret på RMSE, og Diebold-Marino testen ikke virker til og formår at med en vis statistisk sandsynlighed at kunne forudsige aktiekursen for den efterfølgende dag.

9 Diskussion

I de forudgående er der givet et teoretisk fundament samt et metodologisk overblik efterfulgt af resultaterne som blev baseret på de forudgående afsnit. I dette afsnit vil der forsøges diskuteres nogle af de udfordringer ved anvendelsen i særdeleshed ANNs. Derudover vil der blive præsenteret og diskuteret for alternative metodologiske valg som potentielt kunne forbedre resultaterne.

9.1 Kritik af LSTM

Som det fremstår af analysen, så virker LSTM til at slå ARIMA på samtlige fronter, rent grafisk kan der ses at LSTM var markant bedre til at forudsige kursen, hvilket illustreres på figur 17-19 hvilket blev suppleret af tabellerne fra forudgående afsnit, hvor samtlige RMSE og MAPE værdier var lavere for LSTM modellen, hvormed DM testen også anerkendte at nøjagtigheden for LSTM modellens fejlled var lavere. Ingen af modellerne virker dog som overlegne ift. at udvælge vækstaktier. Fordelen ved ARIMA-modellen er dens implementerbarhed hvor LSTM kan virke som en uoverskuelig jagt på de optimale parametre uden en egentlig konsensus om hvorvidt dette er opnået. Dette blev imødekommet i projektet ved at anvende KerasTuner til at automatisere processen, men selv med sådan et redskab defineres der grænser og dermed må det næsten formodes at den optimale model for det udvalgte data stadig findes og ikke er opnået i specialet. Modsat ARIMA hvor alt er tydeligt og kan forklares matematisk så kan ANNs natur forekomme som en ”Black-box”, altså en sort boks, dette skyldes at det kan forekomme umuligt at få et direkte indblik i parametrene og vægtene som modellerne sætter inde i de skjulte lag, hvormed det ikke fremstår klart om variablen har lineære eller ikke-lineære parametre, hvor der i dybe og komplekse netværk kan være op til 64 millioner parametre, eksempelvis i billede genkendelse. Selv hvis der opstilles to netværk med præcis samme modelkonstruktion i form af læringsrate, antallet af neuroner og skjulte lag, så kan modellerne opnå forskellige resultater, hvilket kendetegnes som churn, som kan forståes som to ens modellernes forventede difference i procent. Dette skyldes at der er i ANNs forekommer nogle tilfældige forhold i form af startværdierne under træningen, hvilket skal forsøges mindskes. Dette tilfældige forhold anses ikke som værende en fejl af modellen, men nærmere en af de nødvendige karakteristika som medfører at ANNs har evnen til at lære. (Cormier et al., 2016)

Hvis der tages udgangspunkt i et investeringsperspektiv, kan det forekomme skræmmende at have et output som ikke kan direkte forklares og dette må derfor næsten garantere en høj risiko eftersom resultat kan direkte aflæses men vejen dertil foregår igennem denne sorte boks hvor der foretages

ændringer under træningen af modellen uden at modelkonstruktøren eller andre har et præcist indtryk af hvad der sker. LSTM modellens formåen i praksis er også et diskuteret emne, da modellerne i den finansielle verden ofte skal efterleve nogle overordnede krav men også krav som hvert firma individuelt opstiller, hvilket MLs operationer kan have svært ved at efterleve og dermed kan brugen og validiteten af ANNs diskuteres. Udover den praktiske anvendelse af ANNs så kritiseres de også for at have en naturlig tilbøjelighed til at overfitte eller underfitte under træningen af dataet. Det kan derfor være altafgørende af evaluere variansen mellem træning og testsættet, hvor en traditionel k cross-validerings procedure som ofte er brugt til at backteste modeller ikke virker for ML modellerne (Sen et al., 2021).

9.2 Data Problemet

Et andet diskuteret emne indenfor ML med udgangspunkt i den finansielle verden, da specialet i sin natur kigger på udvælgelser af vækstaktier, er data problemet. Selvom finansielt data er yderst tilgængelig i form af aktiekursen, volumen og meget mere, så kan der nemt argumenteres for at det finansielle data er mindre datasæt når det sættes i sammenligning med de data-hungrende ANNs og især DL modeller, hvor et mindre datasæt naturligt må medføre at modellerne bliver mindre trænet og derfor ikke konstrueret korrekt, hvormed man opnår et resultat der er sub-optimalt i forhold til, i specialets tilfælde, forecasting performance (Israel et al., 2020).

Der er i datasættets fulde periode over de 10 år indsamlet daglige data hvilket har medført 2766 observationer, hvilket selvfølgelig under ingen omstændigheder kan kategoriseres som big data og går under de her mindre datasæt som nævnes i Israel et al., hvor der i specialet er benyttet RW for at imødekomme dette problem, men dette kan muligvis stadig have medført at især LSTMs modeller har haft et sub-optimalt resultat og dermed kunne have præsteret markant bedre såfremt der havde været anvendt time-data eller intraday data. Det skal her også nævnes at der er tidsbegrænsning for især LSTM-model optimering som tager markant længere at tune og opstille end ARIMA-modellen, hvor ARIMA-modellen kun tager 10 minutter at opstille og maksimere efter alle koderne er lavet, hvorimod LSTM med den computerkraft der har været tilgængelig i specialet, så tager det en dag at tune LSTM modellerne, hvilket selvfølgelig er problematisk når der kigges på dagsdata. En yderligere Komplikationen ved at anvende observationer på time-basis eller endda intraday skyldes at transaktionsomkostningerne ville blive øget markant, såfremt der skulle ageres på de signaler modellerne kommer med, hvor det må formodes at såfremt en investor ville skulle agere på alle signalerne fra en time-model eller intraday model så vil transaktionsomkostningerne stige og dermed

må profitten forsvinde. Der kan derfor også debatteres hvorvidt det er anvendeligt at benytte sig af dagligt data som anvendt i specialet, hvor der potentielt kunne fordelagtigt have været anvendt månedlige data, men her ville der have været endnu færre observationer og dermed ville datasættet i større grad have været indbegrebet af small data problemet som nævnt tidligere. (Israel et al., 2020)

9.3 Signal-to-Noice

Udover den datamæssige udfordring forekommer der også yderligere komplikationer da finansielt data er kendetegnet ved lav signal-to-noice ratio, hvor ML modeller ofte trives i data med høj signal-to-noice, hvor grundet modellernes egenskaber er i stand til at lære de signaler der forekommer. ML modellerne lærer fra inputsdataets variable for at finde de signaler der medfører signalet. Med udgangspunkt i den finansielle verden så opstår den lave signal-to-noice på grund af de basale markedskræfter der er i spil i form af konkurrence og profitmaksimering. Såfremt der er information og signaler i markedet, så vil aktørerne, altså investorerne, udnytte disse signaler og handle derefter, hvilket betyder at såfremt der er signaler som kan benyttes til at investere, så vil disse signaler indtil markedet tilpasser sig og værdien af signalet forsvinder og dermed bliver inkorporeret i aktiekursen. Dette bliver yderligere understøttet af det Eugene Fama's teori der blev introduceret i afsnit 2, den efficiente markedshypotese (Israel et al., 2020).

Såfremt ML-modellen lærer de signaler som forekommer i det finansielle markeder, så vil signalet ændret sig eftersom investorernes præference ændres, da de agerer på nye signaler og dermed lærer modellen hele tiden lærer forældede signaler, hvilket hvormed ML konstant skal gentrænes for at lære de nye signaler, som kan virke som en evig process, hvor ML-modellen altid vil være konstant bagud, hvilket stemmer overens med den efficiente markedshypotese, der postulerer at det virker umuligt at slå markedet på lang sigt, da information i form af signaler altid være inkorporeret i aktiekursen og må derfor må aktierne altid formodes at handles til den retmæssige pris.

9.4 Tilføjelse af Variabler

Der er i specialet benyttet én variable, prisen, hvorimod tilføjelse af flere forklarende variabler vil potentielt kunne øge nøjagtigheden i modellerne, hvor det også virker tydeligt at LSTM og ARIMA-modellerne ikke fungerer overlegent som en univariable model. Der kunne fordelagtigt have været foretaget en Granger kausalitets test til at udvælgelse af relevante forklarende variable. Granger

Kausalitets testen blev oprindeligt introduceret af Granger i 1969. Testen forsøger at teste hvorvidt en tidsserie kan benyttes til at forklare en anden tidsserie og testen beviser om der eksisterer en såkaldt granger-kausalitet. Eftersom de udvalgte vækstaktier som er benyttet i specialet alle indgår i S&P 500 så virker det oplagt at teste for om S&P 500 indekset mod de udvalgte aktier, eftersom det kunne virke indlysende at såfremt S&P 500 indekset stiger, så kunne det tyde på at aktierne i indekset ligeledes må stige.

Formålet med udvælgelse af aktiver og aktiekursforudsigelse med henblik på afkast bliver dog argumenteret for at være et small data problem. Tager der udgangspunkt i en simpel regression som kan opstilles matematisk som:

$$Y_t = \sum_{t=1}^{N} \beta_i x_{i,t-1} \tag{44}$$

Hvor t=1

Hvor Y_t er den fremtidige værdi der ønskes prædikteret og på højresiden er de forklarende variable, så kan data blive opdelt i small og big data afhængig af hvor rigt en model der kan opstilles i form af hvor mange relevante forklarende variable der kan benyttes, i specialet tilfælde i form af S&P 500 aktiekursen eller lignende. Hvor der argumenteres for at antallet af uafhængige observationer på venstre siden er denoteret af T, hvor til trods for tilføjelse af flere forklarende variable i form af enten NLP (Natural Language Processing) og sentiment analyse, hvor der gives en score og ud fra den score kan der vurderes holdningen til et givent aktiv. Blot vil sentiment analyse og S&P 500 indekset stadig være variabler på højresiden, hvorimod at variablerne på venstresiden primært består af forrige observationer for Y. Hvor der argumenteres for at modellernes prædikteringsevne vil være det samme uanset om der benyttes N-antal af forklarende variabler, da antallet af effektive estimater der kan anvendes, er baseret på antallet af observationer for Y og dermed er det et small data problem og derfor vil antallet af variabler ikke nødvendigvis gavne specialet og hjælpe modellernes prædikeringsevne (Israel et al., 2020).

10 Konklusion

Dette afsnit har til hensigt at kort præsentere specialets resultater og svar på problemformuleringen som blev opstillet i afsnit 1.

Specialets formål var at besvare følgende problemformulering:

Hvor anvendelig er lineære og ikke-lineære modeller til at prædiktere udviklingen af udvalgte vækst-aktier

Specialet har kigget på ARIMA som den traditionelle lineære model og en ikke-lineære LSTM model. Modellerne er yderst forskellige i sin natur, da ARIMA modellen har rødder i den økonometriske verden hvorimod LSTM-modellen afstedkommer fra AI. Specialet har benyttet sig af tre udvalgte vækstaktier, som blev udvalgt baseret på deres Price-Earning ratio. De udvalgte aktier er Amazon.com (AMZN), American Tower Corporation (AMT) og Merck & Co. (MRK). Hvor der er blevet opstillet adskillige tests for at kunne have fælles evalueringskriterier for begge modeller. Ordnet kan det konkluderes baseret på det anvendte data samt den metodologiske fremgangsmåde at LSTM-modellen med 10 sekvenser virker mere fordelagtigt til at skulle prædiktere udviklingen af aktiekursen for de fornævnte aktier. LSTM modellen opnåede en lavere RMSE og MAPE samtidigt med at DM-testen bekræftede at LSTM havde lavere forecastings-fejl end ARIMA-modellen. Baseret på PT-testen kan det dog ikke statistisk konkluderes at hverken LSTM eller ARIMA kan anvendes til at binært forudsige aktiekursen.

Den daglige procentvise ændring i aktiekursen viste dog at der forekom større udsving og dermed mere volatilitet i forecastings-perioden, hvilket kan have svækket modellernes evne til prædiktering.

Der kan dog konkluderes at selvom ingen af modellerne virker overlegne i deres prædikteringsevne så virker LSTM modellen til at være markant bedre end ARIMA men grundet at man ikke præcist ved hvordan LSTM når til det output den når, så kan det ikke anbefaledes at benytte nogle af de modeller som blev opstillet i specialet til prædiktering af aktiekursen uden nogle markante forbedringer.

11 Litteraturliste

Afeef, M., Ihsan, A. & Zada, H. (2018) “Forecasting Stock Prices through Univariate ARIMA Modeling” - Besøgt 04-11-2021: https://tinyurl.com/2p9prfk6

Atsalakis, G. S. and Valavanis, K. P. (2013) “Surveying Stock Market Forecasting Techniques - Part I: Conventional Methods”. Computation Optimization in Economics and Finance Research Compendium, Nova Science Publishers Inc.

Ayodele, A,A., Adewumi, A. & Ayo, C. (2014). ”Stock Price Prediction using the ARIMA model”. Conference on Computer Modelling and Simulation, Cambridge University. - Besøgt 04-11-2021: https://tinyurl.com/4s7rcytp

Ayodele, A, T. (2010). “Types of Machine Learning Algoirthms”. University of Portsmouth. – Besøgt 04-11-2021: https://tinyurl.com/32buujuj

Bengio, Y, Simard, Y,P. and Fraconi, P. (1994) “Learning long-term dependencies with gradient descent is difficult” - Besøgt 04-11-2021: https://tinyurl.com/2p9c6hcx

Bushaev, V. (2018) ”Adam – latest trends in deep learning optimization” - Besøgt 04-11-2021: https://tinyurl.com/3n44x3b2

Chong, E., Han, C. & Park, C. F. (2017). ”Deep Learning Networks for Stock Market Analysis and Prediction: Methodology, Data Representation, and Case Studies. Durham University Business School – Besøgt 04-11-2021: https://tinyurl.com/2p8sjmp3

Cormier, Q., Fard, M. M., Canini, K & Gupta, M. (2016) “Launch and Iterate: Reducing Prediction Churn”. Advances in Neural Information Processing systems. - Besøgt 04-11-2021: https://tinyurl.com/4mwrzfh7

Corporate Finance Institute - Besøgt 04-11-2021: https://tinyurl.com/5ft94w2n

Egan, M. (2019) “How Elite Investors use artificial intelligence and machine learning to gain an edge”. CNN - Besøgt 04-11-2021: https://tinyurl.com/4m3kspp5

Enders, W. (2015). Applied econometric time series. Wiley, 4th edition.

Eurostat. ”Glossary:In-sample vs. out-of-sample forecasts”. - Besøgt 04-11-2021: https://tinyurl.com/2s4phktu

Fama, E. F. (1970). “Efficient capital markets: A Review of Theory and Empirical Work”.
The Journal of Finance.

Fitzgerald, M. (2019) ”Everyone still relies on a stock’s P-E ratio to invest, but a study show it’s bunk” Cnbc. - Besøgt 04-11-2021: https://tinyurl.com/bdz6kwde

Goodfellow, I. Bengio, Y. and Courville, A(2016). “Deep Learning”. MIT Press - Besøgt 04-11-2021: https://tinyurl.com/3rc5dn7b

Graham, B and J, Zweig. (2003) “The Intelligent Investor” Revised edition.

Granger, C,W.J (1969) “Investigating Casual Relations By Economic Models and Cross-Spectral Methods”. Econometrica vol 37, No. 3. - Besøgt 04-11-2021: https://tinyurl.com/2p96v8fc

Gupta, V. (2017) ”Understanding Feedforward Neural Networks” - Besøgt 04-11-2021: https://tinyurl.com/2p8bfk9h

Gurley, J. and Shaw, E. (1967) “Financial Structure and Economic Development”. Economic Development and Culture

Israel, R,. Kelly, B & Moskowitz T. ”Can Machines “Learn” Finance?”. Journal of Investment Management,. Vol 18, No. 2. – Besøgt 04-11-2021: https://tinyurl.com/2p99drbc

K, Amadeo. (2021) “The S&P 500 and How It Works” - Besøgt 04-11-2021: https://tinyurl.com/2p8cm3eb

Najeeb, M.H Masoud. (2013) “The Impact of Stock Market Performance upon Economic Growth”. Journal of Economics and Financial issues. - Besøgt 04-11-2021: https://tinyurl.com/4ba5bdks

Namin, S. S., Tavakoli, N. and Namin, A. S. (2019), *A Comparative Analysis of Forecasting Financial*
Time Series Using ARIMA, LSTM, and BiLSTM. - Besøgt 04-11-2021: https://tinyurl.com/yc2ah8hn

Nielsen, M. (2019). Neural networks and deep learning. - Besøgt 04-11-2021: https://tinyurl.com/2p8dcku8

Ma, Q. (2020). Comparison of ARIMA, ANN and LSTM for Stock Price Prediction. Sun Yat University. - Besøgt 04-11-2021: https://tinyurl.com/328zbdub

McCarthy, J. (2007)." What is artificial intelligence?" - Besøgt 04-11-2021: https://tinyurl.com/2p8a7hcu

Mehtab, S., Sen, J. & Dutta, A. (2020). Stock Price Prediction Using Machine Learning
And LSTM-Based Deep Learning Models. Department of Data Science and Artificial Intelligence, Praxis Business School. - Besøgt 04-11-2021: https://tinyurl.com/47xc9wvm

O'Malley, Tom and Bursztein, Elie and Long, James and Chollet, Fran C and Jin, Heifeng and Invernizzi, Luca. (2019): "KerasTuner" - Besøgt 04-11-2021: https://tinyurl.com/mw3a3m2t

Pesaran, M. H. and Timmermann, A. (1992), *A Simple Nonparametric Test of Predictive Performance*". Journal of Business and Economic Statistics 10.

Russel S and Norvig, P. (2010) "Artificial Intelligence A Modern Approach" 3rd edition. Pearson Education.

Sen, J., Sen, R. & Dutta, A. "Machine Learning in Finance-Emerging Trends and Challenges." Department of Data Science, Praxis Business School. – Besøgt 04-11-2021: https://tinyurl.com/2p89s23p

Sharma, S. (2017). "Activation Functions in Neural Networks" - Besøgt 04-11-2021: https://tinyurl.com/57epmcnt

W. Lo, A. and MacKinlay, A. C. (1999). "A Non-Random Walk Down Wall Street". Princeton University Press, 5th edition.

Z, Teng, S, Teng, J, Zhang, G, Chen & F, Cui. (2020) “Structural Damage Detection Based on Real-Time Vibration Signal and Convolutional Neural Network” - Besøgt 04-11-2021: https://tinyurl.com/5n8759vu

12 Appendiks

12.1 Bilag A:

Bilag A: Udvidet AIC for ARIMA modellerne.

Model	AMT	AMZN	MRK
ARIMA(0,1,0)*	6608.28	**18488,58**	**3123.33**
ARIMA (1,1,0)*	6602.49	18489.64	3125.07
ARIMA(2,1,2)	6608.28	18493.92	3123.34
ARIMA(0,1,1)*	6602.11	18493.98	3125.04
ARIMA(0,1,0)	6606.12	18495.76	3124.71
ARIMA(1,1,1)*	**6598.57**	18495.99	3125.3
ARIMA(2,1,2)*	6601.69	18496.3	3125,35
ARIMA(0,1,2)*	6602.88	18495.98	3125.11
ARIMA(2,1,1)*	6598.33	18498	3124.35
ARIMA(1,1,1)	6598.59	18498.05	3125.3

Note: Den laveste AIC score er markeret med fed og bliver anvendt som den bedste ARIMA model.

12.2 Bilag B:

Bilag B: List over anvendte forkortelser

AI:	**Artificial Intelligence**	**ADF:**	**Augmented Dickey Fuller**
NN:	**Neurale Netværk**	**RW:**	**Rolling Window**
ML:	**Machine Learning**	**RNN:**	**Recurrent Neural Network**
LSTM:	**Long Short Term Memory**	**DL:**	**Deep Learning**
ANN:	**Artificial Neural Network**		
FNN:	**Feed Forward Neural Network**		
ARIMA:	**AutoRegressive Integrated Moving Average**		
RMSE:	**Root Mean Squared Error**		
MAPE:	**Mean Absolute Percentage Error**		
SML:	**Supervised Machine Learning**		
UML:	**Unsupervised Machine Learning**		
DM:	**Diebold-Marino**		
PT:	**Pesaran Timmerman**		
ReLu:	**Rectified Linear Unit**		
PE:	**Price/Earning**		
BP:	**Back Propropagation**		
AMZN:	**Amazon.com**		
AMT:	**American Tower Corporation**		
MRK:	**Merck & Co**		
AIC:	**Akaike Information Criterion**		