

Forecast Volatility in the Financial Markets

Miloš Obrenović i Lazar Kračunović

1 Uvod

Poslednjih nekoliko godina, razvoj novih procenitelja volatilnosti koji se oslanjaju na high-frequency podatke doveo je do novih načina preciznije procene volatilnosti. Bolja procena nestabilnosti tržišta i dinamike koju ona donosi postala je jedno od najznačajnijih pitanja u oblasti ekonometrije. Bolja procena volatilnosti dovodi do bolje predikcije buduće nestabilnosti (Patton & Sheppard, 2015) [7].

Prognoziranje volatilnosti, iako dugo korisno u upravljanju rizikom, postalo je sve važnije jer je volatilnost sada direktno utiče kroz swapove i futures ugovore. Volatilnost je takođe pozitivno povezana sa cenom opcije jer veće promene cena akcija ili druge imovine povećavaju verovatnoću da će te promene rezultirati opcijom koja je u novcu (in-the-money). Stoga, volatilnost igra ključnu ulogu u cenama opcija. Istovremeno, učesnici na tržištu mogu proceniti implicitnu volatilnost (IV) koju trgovci očekuju za kretanje osnovne imovine posmatrajući cenu opcije na tržištu.

Jedan od superiornijih modela je HAR (Heterogeneous Autoregressive Model), koji je predložen od strane Corsija u 2009 [6]. Teorijska osnova je Heterogena Marketna Hipoteza, koja tvrdi da je trejdovanje heterogeno, o ovome ćemo detaljnije kasnije. Sa razvojem veštačke inteligencije i širenjem domena njene primene, postavlja se pitanje da li nam može pomoći prilikom rešavanja ovog problema. Takođe, jedan od modela koji se pokazao veoma efikasnim u sekvencijalnom učenju je Long Short-Term Memory (LSTM) model. LSTM se istakao u obradi govora i pisanja, te bi mogao biti koristan i za procenu i predikciju volatilnosti na finansijskim tržištima.

Ultimativni cilj ovog rada je da prezentuje i upoređi jednostavan autoregresivni model (HAR) sa modelom zasnovanim na neuronskim mrežama (LSTM). Želimo da analiziramo različite horizonte predviđanja i procenimo njihovu tačnost out-of-sample.

Uzimajući u obzir Efikasnu Marketnu Hipotezu (EMH) (Fama, 1970), prema kojoj svi investitori imaju jednake šanse za informacije (isključujući insajderske informacije), tvrdimo da je nemoguće predvideti cenu akcija za sutra, budući da su cene pokazale sklonost random walk modelu (Malkiel, 1973).

Razlog za korišćenje modela mašinskog učenja je što su, u poređenju s tradicionalnim ekonometrijskim modelima, modeli mašinskog učenja fleksibilniji i pokazali su se efikasni u radu s nelinearnim vezama i multikolinearnošću. Za periode predviđanja koristimo dnevne, nedeljne i mesečne intervale. Ultimativni cilj istraživanja je da se ispita da li neuronske mreže mogu postići slične ili bolje rezultate u poređenju s tradicionalnim ekonometrijskim modelima.

Zbog mogućnosti poređenja rezultata, ovaj rad će pratiti istraživanje Pattona & Shepparda (2015) [7] u vezi sa selekcijom podataka, funkcijama i HAR modelima.

Limitacija ovog rada je što radimo sa podacima iz S&P 500 indeksa, budući da su nam oni bili dostupni. Svesni smo da future ugovori pružaju bolje predviđanje u odnosu na cene akcija koje mi koristimo.

2 Teorija

Na početku želimo da predstavimo najbitnije teorijske pojmove. Prvo želimo da objasnimo šta je i čemu služi volatilnost, kao i teorijsku osnovu iza modela koje ćemo koristiti.

2.1 Volatility

Volatilnost je suštinski apstraktan koncept koji se ne može direktno posmatrati, već zahteva određene apstrakcije i metode za njeno merenje i analizu. Ona predstavlja drugi moment (dispersiju) povraćaja, odnosno koliko se moguća zarada ili gubitak mogu razlikovati od proseka. Važno je napomenuti da veća volatilnost obično nosi veći rizik, ali isto tako može dovesti do potencijalno većih prinosa ili gubitaka.

Koncept "Sharpe ratio" (Sharpe, 1966) predstavlja meru koja upoređuje višak prinosa sa volatilnošću, pri čemu se smatra da su veći povrat i manja volatilnost poželjni (što rezultira većim Sharpe ratiom). Međutim, važno je imati na umu da visok povrat često prati visoku volatilnost, što može smanjiti Sharpe ratio. Sharpe ratio se obično izračunava na osnovu prošle (ex post) volatilnosti, ali idealno bi bilo koristiti "buduću" volatilnost za analizu rizika imovine.

U kontekstu predviđanja, dok tvrdimo da je predviđanje cena kao pronalaženje igle u plastu sena, predviđanje volatilnosti bi se moglo uporediti s procenom veličine tog plasta sena. Ne zahteva se tačno predviđanje pojedinačnih cena, već razumevanje distribucije tih cena i njenih fluktuacija tokom vremena.

Ovaj pristup naglašava složenost i važnost analize volatilnosti na finansijskim tržištima, kao i potrebu za efikasnim modelima i metodama kako bismo bolje razumeli i upravljali rizikom u investicionim odlukama.

2.2 Stilizovane činjenice

Volatilnost ima neke vremenski zavisne osobine koje olakšavaju predviđanje, ali isto tako ima i osobine koje predstavljaju izazov. Jedna od najvažnijih činjenica je da volatilnost prinosa na kapital varira tokom vremena. Cont (2001) [4] definiše stilizovane činjenice kao "uobičajene kod širokog spektra instrumenata, tržišta i vremenskih perioda". Sada ćemo definisati neke od najvažnijih stilizovanih činjenica:

2.2.1 Horizontalna zavisnost

Horizontalna zavisnost odnosi se na stilizovane činjenice koje se javljaju u vremenskim serijama finansijskih instrumenata. Neke od tih činjenica uključuju:

1. **Mean-reversion (vraćanje ka srednjoj vrednosti):** Ova pojava se odnosi na tendenciju vrednosti akcije da se vrati ka dugoročnoj srednjoj vrednosti nakon što se dogodi neka promena ili vest koja utiče na cenu. Na primer, ako dođe do naglog porasta ili pada cene akcije zbog neke vesti, očekuje se da će cena s vremenom ponovo konvergirati ka svojoj dugoročnoj srednjoj vrednosti, što takođe utiče na volatilnost.

2. **Volatility clustering (klasterovanje volatilnosti):** Ova pojava se odnosi na trend da se periodi visoke volatilnosti obično smenjuju s periodima niske volatilnosti, formirajući "klaster" u vremenskim serijama. Ovi klasteri često nastaju kao odgovor na ekonomske ili tržišne vesti koje utiču na ponašanje tržišta.

3. **Long-memory (upornost):** Upornost se odnosi na postojanje dugoročne zavisnosti u vremenskim serijama, gde trenutni povraćaji mogu biti značajno povezani s prošlim povraćajima. Ova karakteristika je često prisutna u visoko-frekventnim finansijskim podacima i tokom perioda kao što su vikendi.

2.2.2 Vertikalna zavisnost

Vertikalna zavisnost se odnosi na skokove u volatilnosti i izboru estimatora koji se koriste za procenu volatilnosti na različitim frekvencijama (dnevne, nedeljne, mesečne), što može direktno uticati na merenje volatilnosti. Neki od aspekata vertikalne zavisnosti uključuju:

1. **Leptokurtosis (debeli repovi):** Ova pojava se odnosi na to da distribucija povraćaja često ima "debele repove" u odnosu na normalnu distribuciju, što ukazuje na veću verovatnoću ekstremnih događaja koji mogu izazvati visoku volatilnost.

2. **Leverage effect (asimetrija):** Asimetrija se odnosi na negativnu korelaciju između volatilnosti i prinosa. To znači da su dani s negativnim prinosom obično praćeni visokom volatilnošću, dok dani s pozitivnim prinosom imaju tendenciju ka nižoj volatilnosti.

Ove osobine volatilnosti su važne za razumevanje ponašanja finansijskih tržišta i imaju značajan uticaj na način na koji se procenjuje rizik i donose investicione odluke.

2.3 Procenitelji volatilnosti

Volatilnost je inherentno neuočljiva i zahteva određene apstrakcije i metode za merenje i analizu. Veliki broj istraživanja koristi potpuno specificirane modele uslovljene srednjim vrednostima i uslovnim varijansama kako bi procenili i analizirali latentne varijable. Jedan način posmatranja volatilnosti jeste kroz parametarske modele poput GARCH-a. Međutim, korišćenje ovih parametarskih modela predstavlja problem jer podrazumeva da je najviše jedan model tačan, ali nijedan nije potpuno precizan. Takođe

je bitno napomenuti da ovi modeli često ne mogu dobro da reprodukuju "stylized facts" empirijskih podataka.

Drugi pristup je korišćenje direktnih indikatora poput kvadrata povraćaja u određenim intervalima. Međutim, ovo može dovesti do problema sa "šumom" (engl. *noise*), gde je jačina grešaka mnogo veća od samog signala. Da bismo to izbegli, istraživači pokušavaju da aproksimiraju trenutnu volatilnost posmatranjem visoko-frekventnih intervala. Ovaj metod, poznat kao realizovana volatilnost, zasnovan je na sumi kvadrata povraćaja tokom jednog dana. (Andersen, 2001) [1]

Mana ovih modela je ta što često ne mogu da replikuju empirijske karakteristike podataka navedenih ranije [6].

Sada ćemo reći da je posmatrana log-price p_0, \dots, p_n at $n+1$ u jednako podeljenim intervalima $[0, t]$, pa realizovanu volatilnost možemo predstaviti na ovaj način :

$$RV = \sum_{i=1}^n r_i^2 \rightarrow [p, p]_t, \quad \text{as } n \rightarrow \infty,$$

gde je $r_i = p_i - p_{i-1}$.

Dalje u radu Barndorff-Nielsen-a (2008) [2] pokazano je da realizovan volatiliti možemo dekomponovati na pozitivno i negativnu semivarijansu.

$$RS^+ = \sum_{i=1}^n r_i^2 \mathbf{I}\{r_i > 0\}, \quad RS^- = \sum_{i=1}^n r_i^2 \mathbf{I}\{r_i < 0\},$$

gde je $\mathbf{I}\{\cdot\}$ indikatorska funkcija.

Patton & Sheppard (2015) su pokazali da ovom dekompozicijom možemo dobiti tačnije rezultate. U ovom radu ćemo koristiti oba benchmark modela.

2.4 Heterogeni autoregresivni modeli

Heterogeni autoregresivni modeli se zasnivaju na hipotezi heterogenog tržišta u trgovanju [10, 6]. Ova hipoteza proizlazi iz činjenice da je volatilnost pozitivno korelisana sa prisustvom na tržištu i obimom trgovanja. Ideja je da na heterogenom tržištu svi učesnici primaju i procenjuju informacije u različitim vremenskim horizontima, na primer, trgovci koji često zatvaraju svoje pozicije na kraju dana.

Hipoteza heterogenog tržišta sugerise da učesnici na tržištu imaju različite strategije i vremenske horizonte kada reaguju na informacije. Na primer, kratkoročni trgovci mogu reagovati brže na trenutne informacije, dok dugoročni investitori mogu imati duže vremenske horizonte u proceni tržišnih uslova.

Heterogeni autoregresivni modeli uključuju ove različite vremenske horizonte i strategije reagovanja u svoje specifikacije, što omogućava bolje modeliranje volatilnosti na tržištima sa različitim tipovima učesnika i njihovim preferencama.

2.5 HAR Model

Najobičniji model pretpostavlja da se volatilnost može aproksimirati sa samo 3 komponente različitih vremenskih intervala. Jedna je dnevna, druga nedeljna i treća mesečna.

$$RV_t = \frac{1}{20} \sum_{i=5}^{20} RV_{t-i}^d, \quad RV_t^w = \frac{1}{5} \sum_{i=1}^5 RV_{t-i}^d, \quad RV_t^m = \frac{1}{21} \sum_{i=1}^{21} RV_{t-i}^d.$$

Kao i u radu Patton-a i Sheppard-a, zavisna varijabla za višednevno predviđanje je definisana kao prosečna suma volatilnosti tokom h dana

$$\bar{y}_{h,t+h} = \frac{1}{h} \sum_{i=1}^h y_{t+i},$$

gde je $y = RV$.

Konačna jednačina koju ćemo koristiti je:

$$\hat{y}_{h,t+h} = y_{t+i} = \mu + \phi_d y_t + \phi_w \frac{1}{4} \sum_{i=1}^4 y_{t-i} + \phi_m \frac{1}{17} \sum_{i=5}^{21} y_{t-i} + \epsilon$$

dok zamenom semi-varijansama dobijamo:

$$\hat{y}_{h,t+h} = \frac{1}{h} \sum_{i=1}^h y_{t+i} = \mu + \phi_d^- y_t^- + \phi_d^+ y_t^+ + \phi_w \frac{1}{4} \sum_{i=1}^4 y_{t-i} + \phi_m \frac{1}{17} \sum_{i=5}^{21} y_{t-i} + \epsilon$$

2.6 Neuronske mreže

Ovaj odeljak ima za cilj da da pregled funkcionalnosti neuronskih mreža. Diskusija o najosnovnijoj verziji neuronskih mreža, takozvane neuronske mreže sa naprednim prenosom podataka - "feed-forward". Ona prati diskusiju o rekurentnim neuronskim mrežama zajedno sa njenim razrađenijim oblikom LSTM neuronske mreže. U daljem tekstu, ANN(Artificial Neural Network) će označavati skraćenicu za neuronsku mrežu.

ANN je definisana ulaznim slojem, brojem skrivenih slojeva i izlaznim slojem. Pored toga, svaki neuron u svakom sloju je povezan sa svakim neuronom u sledećem sloju sa matricom težina. Štaviše, aktivacione funkcije moraju biti definisane za svaki sloj. Sve u svemu, proces daljeg prenosa mreže može se definisati kao :

$$\begin{aligned} a^1 &= \Phi_1(\mathbf{W}_1^T \mathbf{x} + \mathbf{b}) \\ a^l &= \Phi_l(\mathbf{W}_l^T \mathbf{a}^{(l-1)} + \mathbf{b}) \\ y &= \Phi_Y(\mathbf{W}_Y^T \mathbf{a}^{(l-1)} + \mathbf{b}) \end{aligned}$$

gde \mathbf{x} predstavlja ulazni vektor, \mathbf{a}^l aktivaciju l-tog skrivenog sloja, \mathbf{W} odgovarajuća matrica težina, a \mathbf{b} je vektor pristrasnosti, što se može tumačiti kao verovatnoća da se neuron aktivira. Štaviše, proces je definisan aktivacionom funkcijom označenom sa Φ

Različiti slojevi mogu imati različite aktivacione funkcije, u zavisnosti od toga kako bi trebalo da izgleda izlaz. Za potrebe regresije (što je primenljiv slučaj u ovom radu) aktivaciona funkcija izlaznog sloja je linearna, dok je za svrhe klasifikacije aktivaciona funkcija izlaznog sloja sigmoidna funkcija. Preostale aktivacione funkcije skrivenih slojeva moraju biti odabrane tako da se performanse mreže maksimiziraju. (Nielsen, 2015) [11] O najvažnijim aktivacionim funkcijama se govori kasnije u tekstu.

Do sada smo razmatrali samo "feed-forward" NN u kojima je tok informacija konstantan od ulaznog do izlaznog sloja. Glavni nedostatak neuronskih mreža sa direktnim prenosom podataka je nepostojanje memorije u modelu. Rekurentne neuronske mreže (RNN) pokušavaju da prevaziđu ovaj nedostatak integracijom petlji u mrežu (Olah, 2015) [12]. RNN su prilično slični uobičajenim NN-ovima sa prosleđivanjem, osim što je dodata dodatna ulazna funkcija, odnosno izlaz prethodne instance. Matematički gledano, cela mreža se može opisati sledećim skupom jednačina. (Geron, 2019) [8] :

$$\begin{aligned} \mathbf{a}_t^1 &= \Phi_1(\mathbf{x}_t \mathbf{W}_x + \mathbf{a}_{t-1}^1 \mathbf{W}_{a_{t-1}^1} + \mathbf{b}) \\ \mathbf{a}_t^l &= \Phi_l(\mathbf{a}_t^{(l-1)} \mathbf{W}_{a_t^{(l-1)}} + \mathbf{a}_{t-1}^l \mathbf{W}_{a_{t-1}^l} + \mathbf{b}) \\ \mathbf{y}_t &= \Phi_y(\mathbf{a}_t^{(l-1)} \mathbf{W}_{a_t^{(l-1)}} + \mathbf{y}_{t-1} \mathbf{W}_{y_{t-1}} + \mathbf{b}) \end{aligned}$$

Ovo implicira da je \mathbf{y}_t funkcija od \mathbf{x}_t i \mathbf{y}_{t-1} , što je opet funkcija \mathbf{x}_{t-1} i \mathbf{y}_{t-2} . Ovo zauzvrat implicira da je \mathbf{y}_t funkcija svih ulaza od vremena $t = 0$ i stoga mreža ima neku vrstu sećanja na prošlost. (Geron, 2019) [8] Ako neuron ima oblik sećanja, naziva se memorijska ćelija. Rekurentni neuron je osnovna memorijska ćelija jer se njegovo pamćenje sastoji od prošlosti izlaza. Poslednjih decenija izmišljeni su složeniji oblici memorijskih ćelija. Jedna od tih naprednih ćelija je ćelija kratkoročne memorije (Long short-term memory cell).

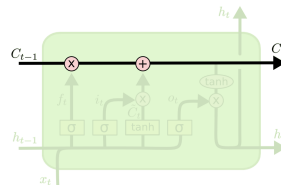
2.7 Long Short Term Memory Cell

Ćeliju dugoročne memorije uveo je 1997. godine Sepp Hochreiter & Jurgen Schmidhuber, a unapređena je tokom poslednje dve decenije. Ćelija je stekla popularnost zbog svojih performansi, brze konvergencije treninga i sposobnosti da pronade dugoročne zavisnosti u podacima. (Geron, 2019). [8]

Hochreiter & Schmidhuber (1997) [9] ističu da sa konvencionalnim metodama učenja za RNN, kao što je nazadno širenje kroz vreme, signali greške su skloni ili da nestanu ili da eksplodiraju. Stoga su došli do

nove rekurentne mrežne arhitekture (nazvane LSTM) u kombinaciji sa odgovarajućim algoritmom učenja zasnovanim na gradijentu (Hochreiter & Schmidhuber, 1997). Sledeće objašnjenje LSTM neuronskih mreža zasnovano je na Olahu (2015) [12], sa ciljem da se ovoj prilično složenoj memorijskoj ćeliji pristupi na intuitivniji način.

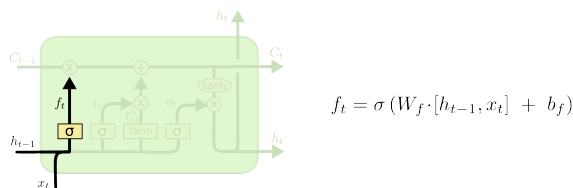
Jedan od ključnih aspekata LSTM ćelije je takozvano stanje ćelije, koje prolazi kroz ćeliju sa različitim tačkama interakcije. Ove tačke interakcije se zovu kapije (gates), jer regulišu tok informacija uklanjanjem ili dodavanjem informacija stanju ćelije. Stanje ćelije je lepo predstavljeno na slici 1.1.



Slika 1.1 : Cell State

Prva interakcija stanja ćelije odlučuje o tome koje informacije treba odbaciti iz stanja ćelije. Ovu odluku donosi sigmoidni sloj koji se naziva sloj kapije zaborava (**forget gate layer**). Ovaj sloj uzima nove ulaze \mathbf{x}_t zajedno sa izlazom iz prethodnog sloja \mathbf{h}_{t-1} kao ulaze i izbacuje vektor brojeva između nule i jedan za svaki element u stanju ćelije \mathbf{C}_{t-1} . Matematički gledano, ovo je predstavljeno u sledećoj jednačini:

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}^T \mathbf{x}_t + \mathbf{W}_{hf}^T \mathbf{h}_{t-1} + \mathbf{b}_f)$$

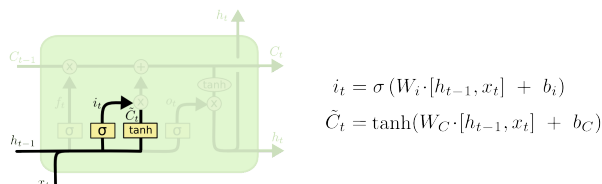


Slika 1.2 : Forget Gate

U sledećem koraku, ćelija mora da odluči koje nove informacije će biti uskladištene u stanju ćelije. Ovaj proces zahteva dva uzastopna koraka. Prvo, sigmoidni sloj, nazvan sloj ulaznih kapija (**input gate layer**), odlučuje koje vrednosti će se ažurirati. Drugo, $\tanh()$ -sloj stvara vektor novih vrednosti kandidata, koji bi potencijalno mogli biti dodati stanju ćelije. Matematički gledano, proces je opisan na sledeći način:

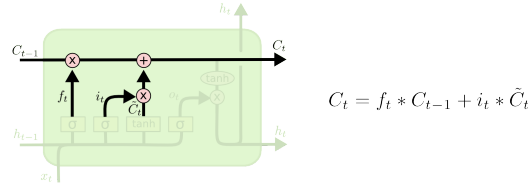
$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}^T \mathbf{x}_t + \mathbf{W}_{hi}^T \mathbf{h}_{t-1} + \mathbf{b}_i)$$

$$\mathbf{g}_t = \tanh(\mathbf{W}_{xg}^T \mathbf{x}_t + \mathbf{W}_{hg}^T \mathbf{h}_{t-1} + \mathbf{b}_g)$$



Slika 1.3 : Input Gate

U sledećem koraku, prethodno stanje ćelije se ažurira interakcijom sa kapijama. Prvo, izlaz kapije za zaborav (forget gate layer) se množi sa stanjem ćelije, čime se zaboravljaju delovi koje smo ranije odlučili da zaboravimo. Drugo, skalirane nove vrednosti kandidata se dodaju stanju ćelije. matematički govoreći:



Slika 1.4 : Output Gate

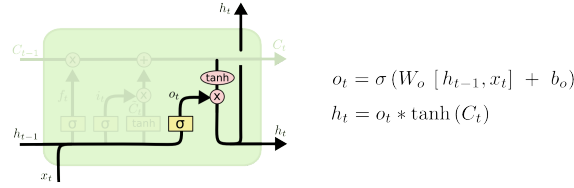
$$\mathbf{c}_t = \mathbf{f}_t \oplus \mathbf{c}_{t-1} + \mathbf{i}_t \oplus \mathbf{g}_t$$

U poslednjem koraku, ćelija odlučuje šta će prikazati, što je filtrirana verzija stanja ćelije.

U prvom koraku, uvodi se novi sigmoidni sloj koji uzima nove ulaze x_t zajedno sa izlazom iz prethodnog sloja h_{t-1} kao ulaze i odlučuje o tome šta će stanje ćelije da izađe. U drugom koraku, guramo stanje ćelije kroz $\tanh()$ funkciju, da bismo nametnuli vrednosti u opseg od -1 i 1, a zatim ga pomnožili sa izlazom prethodno pomenute sigmoidne kapije. Ova dva procesa su formalno objašnjena sa dve jednačine:

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}^T \mathbf{x}_t + \mathbf{W}_{ho}^T \mathbf{h}_{t-1} + \mathbf{b}_o)$$

$$\mathbf{y}_t = \mathbf{o}_t \oplus \tanh(\mathbf{c}_t)$$



Slika 1.5 : LSTM Output

3 Metodologija

3.1 Tok projekta

Kao osnova ovog istraživanja poslužili su nam podaci tick-by-tick. Koristili smo uzorke podataka SPY u intervalima od 5 minuta tokom određenog perioda. Da bismo dobili relevantne mere dnevnog realizovanog volatiliteta, koristili smo uzorke i iz njih izveli karakteristike. Na kraju smo uklonili ekstremne vrednosti, transformisali i skalirali podatke za svaki model posebno, nakon čega smo ih trenirali i validirali.

3.2 Podaci

Koristili smo "besplatne" sirove podatke koje smo pronašli na internetu, obzirom da su visokofrekventni podaci skupi. Pronašli smo dataset koji sadrži S&P500 Trust Exchange Traded Fund (ETF), sa uzorcima u intervalima od 5 minuta. Podatke smo filtrirali da obuhvate samo vremenski period od 09:30 do 16:00, što je u skladu sa metodologijom Patton & Sheppard (2015).

3.3 Uzorkovanje i izrada karakteristika

Budući da nismo imali pristup visokofrekventnim podacima, koristili smo podatke uzorkovane u intervalima od 5 minuta, što se razlikuje od pristupa Pattona & Shepparda. Svesni smo da ovakav način uzorkovanja može uticati na kvalitet podataka, jer jednako razdvojeni podaci mogu da umanje razlike između visoko i nisko frekventnih intervala. Nakon definisanja podataka, pripremljeni podaci za obradu uključuju realizovani volatilitet, realizovane pozitivne i negativne semivarijanse, kao i **bipower** varijansu za svaki dan u analiziranom vremenskom opsegu. Takođe, jednostavno možemo izračunati prosečni realizovani volatilitet na nedeljnom i mesečnom nivou.

3.4 Preprocesiranje podataka

Nakon što smo izračunali sve potrebne vrednosti za naše modele, vreme je za njihovu obradu. Prvi korak je uklanjanje ekstremnih vrednosti iz vremenskih serija, zatim reskaliranje i transformacija podataka, a poslednji korak je podela podataka na trening, validacioni i test skup.

3.4.1 Jump detection i transformacija

Merenje realizovane volatilnosti kao suma kvadrata varijansi logaritmovanih vrednosti hvata skokove u cenama i volatilitetu. Prema Corsiju (2010)[5], predloženo je korišćenje threshold-based pristupa za detekciju skokova. Za prag se koristi standardna devijacija prethodnih 200 dana, definisan kao rolling window. Ovom procedurom je eliminisano oko 3% podataka.

Prag za skokove se koristi u svim metodama. Prilikom korišćenja HAR modela, predikcije se vrše i sa i bez logaritamskih transformacija, dok se neuronske mreže treniraju na logaritmovanim i normalizovanim podacima.

3.4.2 Training, Validation, Testing sets

Kao što možemo videti izabrani smo malo veće skupove za testiranje i validaciju, razlog tome je taj što smo hteli da vidimo kako se modeli ponašaju na dužim vremenskim intervalima. Pored toga moramo da naglasimo da smo prilikom treniranja HAR modela koristili i trening skup i validacioni skup kao trening skup, svesni smo da možemo doći do preprilagođavanja i da je ocena greške na trening skupu optimistična.

Table 1: Podela na trening, test i validacioni skup

Skup	Pocetak	Kraj	Duzina	Duzina (procenti)
Training	01.01.2006	12.31.2012	1,704	$\approx 60\%$
Validation	01.01.2013	12.31.2015	734	$\approx 20\%$
Testing	01.01.2016	12.31.2018	742	$\approx 20\%$

3.5 Trening neuronske mreže

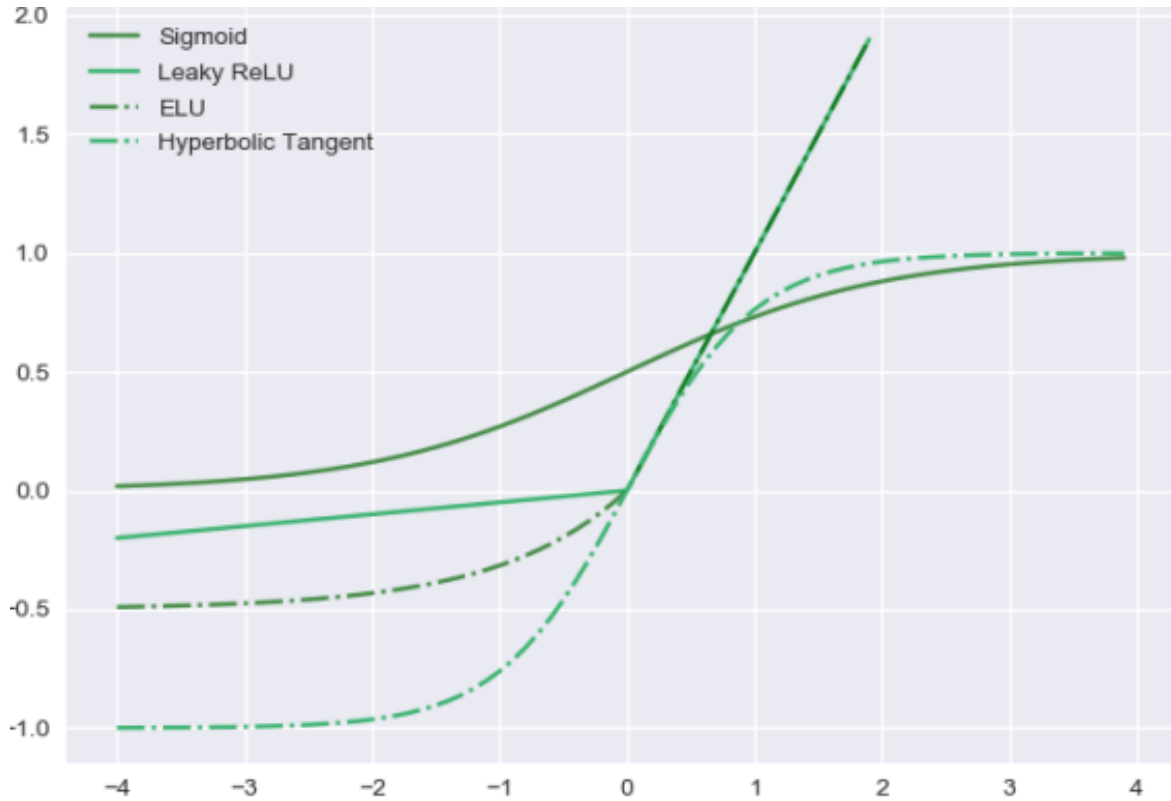
3.5.1 Podešavanje hiperparametara

Hiperparametri igraju ključnu ulogu u arhitekturi i treniranju neuronskih mreža. Bez obzira da li model radi prosečno ili izuzetno dobro, često se svodi na pravi izbor hiperparametara. Ovaj odeljak naglašava odluku o izboru hiperparametara za primenu. Izbor hiperparametara se zasniva na dva različita pristupa. Za neke od hiperparametara, pregled literature čini osnovnu osnovu za izbor. Za druge, izbor sa najboljim učinkom se istražuje uz pomoć algoritama optimizacije. Konkretno, za pronalaženje odgovarajućih rešenja, primenjuje se pretraga mreže.

Hiperparametri o kojima se govori u ovom odeljku su aktivacija i funkcija gubitka.

Aktivaciona funkcija : Funkcija aktivacije za svaki neuron je izuzetno važna za performanse modela. Dugo vremena je funkcija logističke aktivacije zajedno sa hiperboličkom tangentom bila istaknuti izbor za aktivacione funkcije. Međutim, problem sa obe ove funkcije je u tome što se one zasićuju u ekstremima, što znači da ako ulazi postaju veliki (i pozitivni ili negativni), funkcija se zasićuje na 0 (-1 za $\tanh(x)$) i 1. Ovo zauzvrat implicira da je izvod u ovim tačkama blizu nule, što uzrokuje problem nestajanja gradijenta. (Geron, 2019) [8] Slika 2.1 pokazuje da su repovi obe funkcije spljošteni, pa stoga izvod postaje nula.

Ovo pitanje je dovelo do razvoja efikasnijih aktivacionih funkcija, **Rectified Linear Unit (ReLU)** i **Exponential Linear Unit (ELU)**. ReLU se smatra najboljom funkcijom za aktivaciju jer je ublažio problem nestajanja gradijenta. Međutim, funkcija aktivacije ReLU je nenegativna što implicira srednju aktivaciju različitu od nule. Ovo se zauzvrat pojavljuje kao izvor pristrasnosti u mreži. (Clevert, Unterthiner, Hochreiter, 2015) . Alternativni metod za ReLU predložili su Clevert i Unterthiner (2015) u



Slika 2.1 : Aktivacione funkcije

radu „Fast and Accurate Deep Network Learning by Exponential Linear Units“ [3]. Funkcija ELU aktivacije je definisana kao:

$$ELU_{\alpha}(z) = \begin{cases} \alpha(\exp(z) - 1), & z \leq 0 \\ z, & z \geq 0 \end{cases}$$

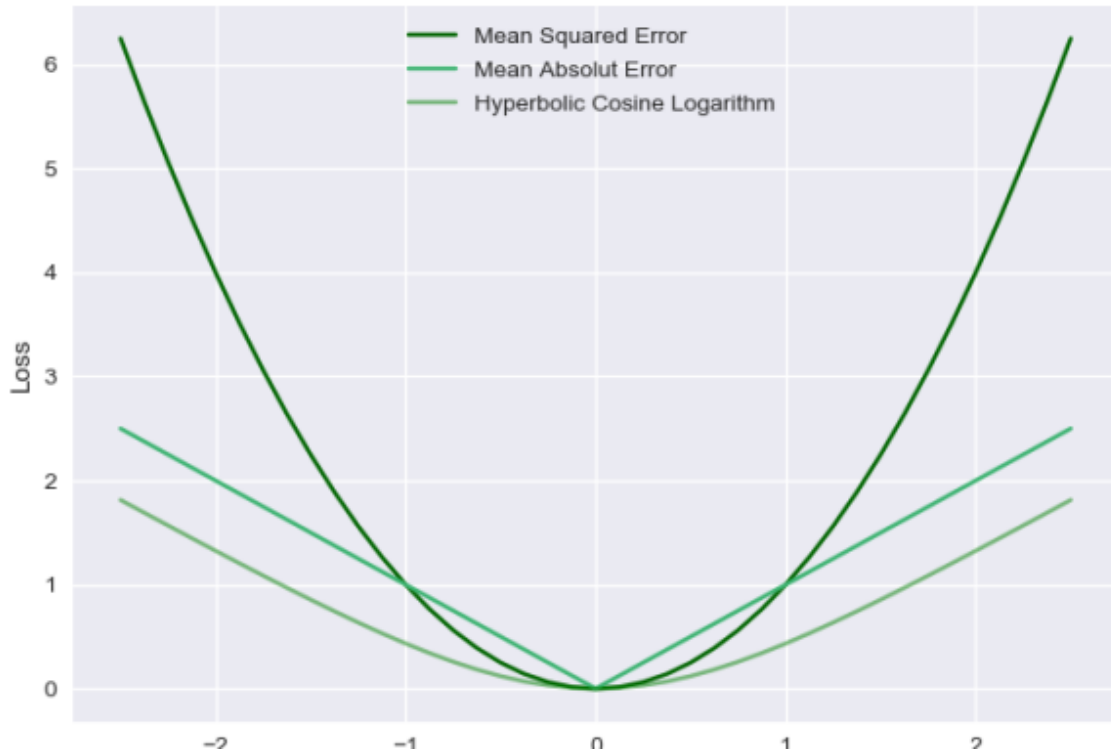
ELU-ovi su pokazali da povećavaju brzinu učenja, kao i značajno bolje performanse generalizacije u poređenju sa svim drugim funkcijama aktivacije. Štaviše, kako funkcija prima negativne vrednosti, ona dozvoljava prosek aktivacije bliži nuli. Ovo zauzvrat pomaže u ublažavanju problema nestajanja gradijenta. Slika 2.1 lepo ilustruje prednosti eksponencijalne linearne jedinice. Na osnovu ovih razloga, ovaj rad koristi ELU kao aktivacionu funkciju za neuronsku mrežu.

Funkcija greske : Funkcija gubitka (ponekad i troškova) kvantifikuje performanse predviđanja izlaza iz mreže (Nielsen, 2015) [11]. Uporedivo sa linearnom regresijom gde je funkcija gubitka definisana kao srednja kvadratna greška greške predviđanja (Geron, 2019)[8]. U okviru ove teze, analizirane su tri različite funkcije gubitaka kako bi se izabrala pogodna opcija. Funkcije gubitaka o kojima se govori u aplikaciji su srednja kvadratna greška (MSE), srednja apsolutna greška (MAE) i hiperbolički kosinusni logaritam (LogCosh). Pogled na oblik različitih funkcija gubitka odmah će dati uvid u prednosti i nedostatke svake funkcije.

MAE : Kao što se može primetiti na slici 2.2, MAE ima konstantan gradijent za sve nivoe greške predviđanja. Ovo može biti štetno jer velika greška predviđanja ima isti efekat na optimizator kao i vrlo mala greška predviđanja. Ovo je očigledno nedostatak za korišćenje MAE kao funkcije gubitka. Bez obzira na to, MAE ima prednost u tome što ne kažnjava nesrazmerno one koji su van granica.

MSE : Za razliku od MAE, MSE ne pokazuje konstantan gradijent u celom prostoru greške. Ovo je korisno jer manje greške dovode do malih gradijenata, dok velike greške predviđanja dovode do većih gradijenata. Problem sa MSE-om je što on nesrazmerno kažnjava veću grešku predviđanja.

LogCosh : Kao što je ilustrovano u 2.2, logaritamska hiperbolična kosinusna funkcija je donekle



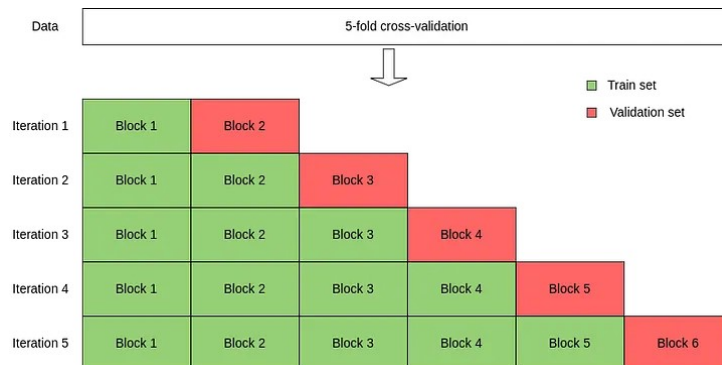
Slika 2.2 : Funkcije greške

mešavina između MAE i MSE. $\log(\cosh \epsilon)$ je približno jednak $(\epsilon^2)/2$ za male ϵ i $|\epsilon| - \log(2)$ za velike ϵ . To s jedne strane implicira da se za male greške predviđanja oblikuje slično MSE, što implicira da manje greške predviđanja imaju manji gradijente nego veće greške predviđanja. S druge strane, za velike greške u predviđanju, oblik je ekvivalentan MAE, što sugerise konstantne gradijente i stoga na funkciju gubitka neće snažno uticati povremena izuzetno netačna predviđanja.

Kao funkciju gubitka smo odlučili da koristimo hiperbolički kosinus logaritma (LogCosh) za modeliranje volatilnosti, pošto su podaci o finansijskoj volatilnosti veoma iskrivljeni i stoga podložni povremenim odstupnicima.

3.5.2 Arhitektura mreže

Za izbor određene arhitekture za modele koristili smo **Time Series split cross-validation**. Podelili smo validacioni skup na 5 jednakih delova. Trenirali smo modele sa različitim paramterima i birali onaj



Slika 3.1 : Cross validation

čiji je zbir ocena grešaka na trening R2 i test R2 najveći. Na ovaj način sprečavamo model da se prilagodi određenom trendu koji se dešava na tržištu.

Nakon isprobavanja raznih kombinacija za broj neurona u prvom i drugom sloju, kao i prisustvo semi-varijanse, došli smo do sledećih arhitektura:

Table 2: Arhitektura mreža

	Dnevni			Nedeljni			Mesecni		
	L1	L2	LR	L1	L2	LR	L1	L2	LR
$LSTM_{20}^{RV}$	20	20	0.01	20	40	0.01	40	20	0.01
$LSTM_{30}^{RV}$	20	20	0.01	20	40	0.01	20	40	0.01
$LSTM_{20}^{SV}$	20	20	0.01	40	20	0.01	40	20	0.01
$LSTM_{30}^{SV}$	20	20	0.01	20	20	0.01	20	40	0.01

4 Referentni modeli

HAR modeli (Heterogeneous Autoregressive Models) su parsimoniozne restriktivne verzije visoko-rednih autoregresija. Standardni HAR model u literaturi o realizovanoj varijansi regresira realizovanu varijansu na 3 terma: prošlu 1, 5 i 22 prosečne realizovane varijanse. Corsi [6] navodi da je standardna obična metoda najmanjih kvadrata (OLS) konzistentna i normalno distribuirana, međutim, pominje se da se javlja problem kada se razmatra višekoraktno predviđanje. Tačnije, pri višekoraktnom predviđanju, prisustvo preklapajućih regresora čini uobičajenu inferenciju neadekvatnom. Iz tog razloga, kako Corsi tako i Patton & Sheppard [7] primenjuju korekciju kovarijanse prema Newey-West-u kako bi izvršili inferenciju o procenjenim parametrima. Stoga, u primeni, autor primenjuje Newey-West HAC (Heteroskedastično i autokorelirano korekciono) sa brojem zaostajanja ekvivalentnim broju $2(h-1)$, gde je h dužina vođene varijable, kao što je predloženo od strane Patton & Sheppard. Patton & Sheppard takođe ističu da procena putem OLS ima nezgodnu manu fokusiranja na prilagođavanje podataka tokom perioda visoke volatilnosti, dok su mirniji periodi sa nižom volatilnošću zanemareni. Kako bi prevazišli ovaj problem, Patton & Sheppard koriste težinsku metodu najmanjih kvadrata (WLS) za procenu. Alternativno, umesto primene WLS procene, moglo bi se primeniti log-transformacija podataka o realizovanoj volatilnosti, a zatim primeniti redovna OLS. Naravno, potrebno je ponovo transformisati podatke na početni nivo kako bi se predvideli budući nivoi volatilnosti.

5 Rezultati

5.1 Kriterijum evaluacije

Za kriterijum evaluacije koristićemo standardne RMSE (Root Mean Squared Error) i MAE (Mean Absolute Error), kao i MAPE (Mean Absolute Percentage Error). Takođe ćemo koristiti Andersen i Bollerslev-ov pristup (1998) pa ćemo računati i R-squared u odnosu na osnovnu regresiju.

5.2 Dnevni forecast

Ovo su rezultati za dnevni forecast:

Table 3: Rezultati dnevnog forecasta

	Train			Test		
	RMSE	MAE	R2	RMSE	MAE	R2
HAR^{RV}	0.9655	3.2178	0.7243	0.3135	1.7443	0.6334
HAR_{log}^{RV}	1.0133	3.1667	0.6963	0.3217	1.6141	0.6142
HAR^{SV}	0.9596	3.1411	0.7277	0.3084	1.7099	0.6453
HAR_{log}^{SV}	1.0175	3.1470	0.6938	0.3187	1.5989	0.6213
$LSTM_{20}^{RV}$	0.9771	3.2359	0.7176	0.3311	1.6509	0.5912
$LSTM_{30}^{RV}$	1.2983	5.1502	0.5015	0.4195	2.0558	0.3438
$LSTM_{20}^{SV}$	1.1456	3.4010	0.6119	0.3293	1.7497	0.5955
$LSTM_{30}^{SV}$	1.1163	3.7944	0.6315	0.3319	1.9228	0.5893

RMSE je pomnožen sa $1e+4$ radi bolje reprezentacije dok je MAE pomnožen sa $1e+5$

Kao što smo očekivali HAR modeli zasnovani na semi-varijansi su bolj od modela zasnovanih na realizovanom volatilitiju, što se poklapa sa rezultatima [7], dok to nije uvek tačno za LSTM modele. Kako RMSE kažnjava velika odstupanja, dok MAE sve greške posmatra isto, možemo zaključiti da $LSTM^{RV}$ obično daje veća predviđanja nego $LSTM^{SV}$. Takođe, možemo primetiti da modeli zasnovani na većem lagu daju bolje rezultate, odakle možemo pretpostaviti da postoji dodatna informacija koja je poželjna u daljoj istoriji.

Sa druge strane vidimo da LSTM ne daje bolje rezultate od HAR modela. Jedna od pretpostavki je ta da nismo trenirali model na dovoljno velikom skupu, kao i da je skup za određivanje hiper parametara bio jako mali, pa nismo uspeali da pronađemo dobru arhitekturu mreže.

5.3 Nedeljni forecast

Ovo su rezultati za nedeljni forecast:

Table 4: Rezultati nedelnog forecasta

Model	Train			Test		
	RMSE	MAE	R2	RMSE	MAE	R2
HAR^{RV}	0.8599	3.0847	0.7330	0.2831	1.8292	0.6129
HAR_{log}^{RV}	0.9215	3.0687	0.6934	0.2814	1.5765	0.6176
HAR^{SV}	0.8537	3.0448	0.7368	0.2784	1.7945	0.6257
HAR_{log}^{SV}	0.9201	3.0620	0.6944	0.2802	1.5780	0.6208
$LSTM_{20}^{RV}$	1.1551	3.6549	0.5183	0.3351	1.8505	0.4577
$LSTM_{30}^{RV}$	0.8190	3.0413	0.7578	0.2923	1.9365	0.5873
$LSTM_{20}^{SV}$	0.8504	2.9026	0.7389	0.2860	1.6566	0.6050
$LSTM_{30}^{SV}$	0.8378	3.0829	0.7465	0.3224	1.7299	0.4982

Note: This is a footnote for the dnevni forecast table.

Pored stvari koje smo već diskutovali, možemo primetiti da u ovom slučaju modeli sa SV su bolji od RV modela kada posmatramo MAE u okviru LSTM modela.

5.4 Mesečni forecast

Ovo su rezultati za mesečni forecast:

Table 5: Rezultati mesečnog forecasta

Model	Train			Test		
	RMSE	MAE	R2	RMSE	MAE	R2
HAR^{RV}	0.9030	3.6703	0.6447	0.2983	2.3624	0.3722
HAR_{log}^{RV}	0.9840	3.4763	0.5780	0.2649	1.6223	0.5049
HAR^{SV}	0.8974	3.6554	0.6490	0.2963	2.3365	0.3806
HAR_{log}^{SV}	0.9836	3.4740	0.5784	0.2652	1.6246	0.5037
$LSTM_{20}^{RV}$	0.9352	3.3539	0.6188	0.2767	1.7476	0.4599
$LSTM_{30}^{RV}$	1.1763	3.7923	0.3970	0.2980	1.9844	0.3735
$LSTM_{20}^{SV}$	0.9283	3.2693	0.6245	0.2705	1.8752	0.4839
$LSTM_{30}^{SV}$	0.9541	3.2824	0.6033	0.2928	1.8847	0.3954

Note: This is a footnote for the dnevni forecast table.

U ovom slučaju, primećujemo da je LSTM značajno bolji u odnosu na HAR modele kada je u pitanju mesečno predviđanje.

6 Zaključak

Ovaj rad je imao za zadatak da ilustruje kako se neuronska mreža može koristiti umesto standardnih statističkih metoda. Iako su rezultati bili malo lošiji u odnosu na klasične metode za jednostavne mreže i malo podataka, to je u neku ruku bilo i očekivano. Kao što smo već rekli, jasno je da neuronske mreže dolaze do izražaja tek kada se koriste sa puno podataka, kao i sa relativno normalnom distribucijom. Već smo rekli da deformisanje distribucije dovodi do raznih problema koji utiču na teže predviđanje. Takođe, prilikom implementacije je korišćena **min-max** transformacija koja normalizuje podatke u odnosu na minimum i maksimum u podacima koji moraju biti premereni. Kako smo za skaliranje koristili samo podatke iz trening skupa, to može dovesti do problema, jer je nemoguće predvideti ekstremne vrednosti koje donosi budućnost. Takođe, jasno je da su modeli poput HAR-a lako interpretirani, pa su prilagodniji za razumevanje, dok mreža predstavlja **black box** koji nije lako intepretirati, stoga je jako teško istražiti kako se model ponaša.

References

- [1] T.G. Andersen. Realized volatility and its use in estimating stochastic volatility models. 2001.
- [2] Kinnebrock S. Shephard N. Barndorff-Nielsen, O.E. Measuring downside risk-realised semivariance. 2008.
- [3] Unterthiner T. Hochreiter S. Clevert, D.-A. Fast and accurate deep network learning by exponential linear units. 2015.
- [4] R. Cont. Empirical properties of asset returns: Stylized facts and statistical issues. 2001.
- [5] Davide Pirino Corsi, Fulvio and Roberto Reno. Threshold bipower variation and the impact of jumps on volatility forecasting. 2010.
- [6] F. Corsi. A smiple approximate long-memory model of realized volatility. 2009.
- [7] Michel Goossens, Frank Mittlebach, and Alexander Samarin. The latex companion. 1993.
- [8] A Géron. Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems. 2019.
- [9] Schmidhuber J. Hochreiter, S. Long short-term memory. neural computation. 1997.
- [10] Dacorogna M. M. Dave R. D Pictet O. V. Olsen R. B. Ward J.R. Muller, U.A. Fractals and intrinsic time: A challenge to econometricians. 1993.
- [11] M.A Nielsen. Neural networks and deep learning. 2015.
- [12] Christopher Olah. Understanding lstm networks. 2015.