SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 976

MODEL DUBOKOG UČENJA ZA OPTIMIZACIJU PORTFELJA ZASNOVAN NA HETEROGENIM PODACIMA

David Supančić

Zagreb, travanj, 2025.

Zahvaljujem mentoru doc. dr. sc. Stjepanu Begušiću na pomoći pri izradi diplomskog rada.

Sadržaj

1. Uvod

Optimizacija portfelja predstavlja jedan od temeljnih izazova u financijskoj industriji. Još od pionirskog rada Harryja Markowitza iz 1952. godine i razvoja Moderne teorije portfelja (MPT), znanstvenici i praktičari neprestano traže učinkovitije i robusnije metode za optimalnu alokaciju imovine. Tradicionalni pristupi poput MPT-a i optimizacije temeljene na Sharpeovom omjeru oslanjaju se na pretpostavke koje često ne uspijevaju uhvatiti punu složenost i dinamiku stvarnih financijskih tržišta. Osobito se ističe osjetljivost tih metoda na točnu procjenu očekivanih prinosa (mean) i kovarijanci, što u praksi rezultira nestabilnim i neoptimalnim portfeljima.

S razvojem računalne snage, napretkom algoritama i sve većom dostupnošću financijskih podataka, metode strojnog učenja postaju sve privlačnije za rješavanje problema optimizacije portfelja. Posebno duboko učenje pokazuje značajan potencijal u modeliranju složenih, nelinearnih i vremenski zavisnih odnosa unutar financijskih serija podataka. U tom kontekstu, LSTM (Long Short-Term Memory) neuronske mreže, koje su posebno dizajnirane za obradu vremenskih nizova, ističu se kao moćan alat jer mogu učiti dugoročne zavisnosti među podacima — što je od ključnog značaja za precizno financijsko modeliranje.

Za razliku od klasičnih pristupa koji zahtijevaju eksplicitnu procjenu srednje vrijednosti i varijance povrata, cilj ovog rada je upravo zaobići taj korak. Umjesto toga, predloženi model koristi funkciju cilja koja omogućuje modelu da samostalno uči optimalnu strategiju alokacije imovine iz sirovih podataka. Time se eliminira potreba za potencijalno nepouzdanom statističkom procjenom i omogućuje izgradnja robusnijeg, adaptivnog sustava za upravljanje portfeljem.

U ovom radu implementiran je generalizirani model dubokog učenja za dinamičku alokaciju imovine s ciljem optimizacije različitih funkcija cilja. Model omogućuje uključivanje bilo koje količine podataka o raznim vrstama imovine i indikatorima. Učinkovitost predloženog pristupa analizirana je kroz vremenski period od 15 godina. Dobiveni rezultati uspoređuju se s tradicionalnim investicijskim strategijama kako bi se procijenila prednost pristupa temeljenog na dubokom učenju.

2. Optimizacija portfelja

Jedan od temeljnih izazova u financijama jest kako optimalno raspodijeliti raspoloživi kapital između različitih investicijskih instrumenata. Cilj je maksimizirati očekivani povrat, uz istovremeno upravljanje rizikom. U tom kontekstu, teorija optimizacije portfelja igra ključnu ulogu. Najpoznatiji i najutjecajniji model optimizacije portfelja razvio je Harry Markowitz 1952. godine, čime je postavio temelje moderne teorije portfelja (engl. *Modern Portfolio Theory*, MPT).

2.1. Markowitzov model

Markowitzov pristup temelji se na pretpostavci da investitori žele maksimizirati očekivani povrat portfelja za zadanu razinu rizika, odnosno minimizirati rizik za zadani očekivani povrat. Model uvodi kvantitativni okvir za procjenu odnosa između rizika i povrata te naglašava važnost diverzifikacije.

Pretpostavimo da je portfelj sastavljen od n različitih imovina, s pripadajućim udjelima $w_1, w_2, \dots, w_n$, gdje vrijedi:

$$\sum_{i=1}^{n} w_i = 1$$

Očekivani povrat portfelja računa se kao:

$$E(R_p) = \sum_{i=1}^{n} w_i E(R_i)$$

gdje je $E(R_i)$ očekivani povrat pojedine imovine. Rizik portfelja izražen je varijancom ukupnog povrata:

$$\sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \text{Cov}(R_i, R_j)$$

U matričnom obliku, varijanca portfelja može se zapisati kao:

$$\sigma_p^2 = \mathbf{w}^T \Sigma \mathbf{w}$$

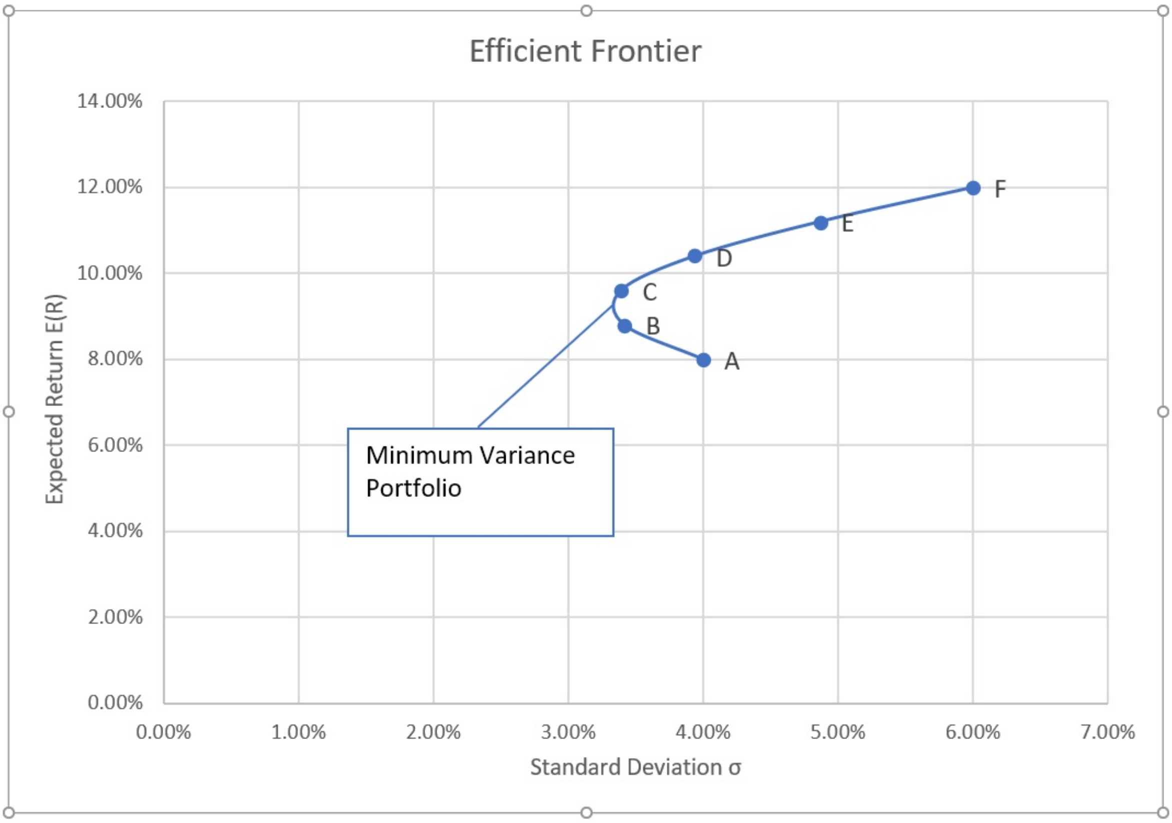
gdje je Σ matrica kovarijanci između svih parova imovina, a $\mathbf{w}$ je vektor udjela u portfelju.

Efikasna granica

Jedan od ključnih rezultata Markowitzove teorije je pojam *efikasne granice* (engl. *efficient frontier*). To je skup svih portfelja koji za zadanu razinu rizika nude maksimalni očekivani povrat, odnosno minimalni rizik za zadani očekivani povrat.

Portfelji koji leže ispod efikasne granice su suboptimalni jer nude manji povrat za istu razinu rizika. Portfelji iznad granice su teorijski nedostižni zbog ograničenja tržišta i realnih uvjeta trgovanja.

Geometrijski, efikasna granica u prostoru očekivanog povrata i rizika (standardne devijacije) ima oblik parabole otvorene prema gore. Svaki portfelj na toj granici predstavlja optimalnu alokaciju resursa za neku razinu prihvatljivog rizika.



Slika 2.1. Efikasna granica. [3]

Markowitz je naglasio važnost diverzifikacije — ideje da kombiniranjem imovina koje nisu savršeno pozitivno korelirane, investitor može smanjiti ukupni rizik portfelja. Varijanca portfelja ovisi ne samo o varijancama pojedinih imovina, već i o kovarijancama među njima.

Primjerice, ako su dvije imovine negativno korelirane, njihova kombinacija može rezultirati portfeljem s manjom ukupnom varijancom nego što bi se očekivalo na temelju pojedinačnih rizika.

Iako je Markowitzov model revolucionaran, ima i određene nedostatke:

- Pretpostavlja da su povrati normalno distribuirani, što u praksi često nije slučaj.
- Potrebna je procjena velikog broja parametara (sredine, varijance i kovarijance), što može biti izazovno i sklono pogreškama.
- Model ne uzima u obzir transakcijske troškove i ograničenja likvidnosti.

Unatoč tim ograničenjima, Markowitzova teorija predstavlja temelj gotovo svih modernih pristupa upravljanja portfeljem, uključujući i kasnije proširene modele poput Sharpeovog modela s tržišnim portfeljem (CAPM) i višefaktorskih modela.

Funkcija korisnosti investitora

Iako efikasna granica predstavlja skup optimalnih portfelja u smislu odnosa rizika i povrata, konačni izbor portfelja ovisi o sklonosti investitora prema riziku. Tu ulogu ima *funkcija korisnosti* (engl. *Utility function*), koja kvantificira individualne preferencije investitora.

Formula funkcije korisnosti može se zapisati kao:

$$U = (E(R_p) - R_f) - \frac{1}{2}\lambda\sigma_p^2$$

gdje je:

- $E(R_p)$ je očekivani povrat portfelja,
- R_f je bezrizična kamatna stopa,
- σ_p^2 je varijanca portfelja (kvadrat standardne devijacije),
- λ je koeficijent averzije prema riziku (što je veći, to je investitor manje sklon riziku).

Izraz $(E(R_p) - R_f)$ predstavlja premiju za rizik, odnosno dodatni povrat koji investitor očekuje u odnosu na sigurnu investiciju. S druge strane, izraz $\frac{1}{2}\lambda\sigma_p^2$ predstavlja „kaznu" za preuzeti rizik. Funkcija korisnosti tako ilustrira ključni kompromis: veći povrat povećava korisnost, ali veći rizik (osobito za riziku nesklone investitore) ju umanjuje.

Investitor tada bira portfelj koji maksimizira ovu funkciju korisnosti. Na taj način se biraju točke s efikasne granice koje najbolje odgovaraju osobnim preferencijama prema riziku. Dakle, iako postoji beskonačno mnogo efikasnih portfelja, funkcija korisnosti omogućuje personaliziran odabir „najboljeg" portfelja za pojedinog investitora.

Ova ideja naglašava ključnu poantu: **investitor ne maksimizira samo povrat, već optimizira omjer između povrata i rizika prema vlastitoj sklonosti riziku**.

3. Metodologija

3.1. Long Short-Term Memory

Duboko učenje predstavlja podskup strojnog učenja koji koristi višeslojne neuronske mreže za učenje kompleksnih reprezentacija podataka. Za razliku od tradicionalnih statističkih metoda, duboke neuronske mreže mogu automatski otkriti složene obrasce u podacima bez potrebe za eksplicitnim definiranjem značajki. Ključni aspekti dubokog učenja uključuju:

- Automatsku ekstrakciju značajki
- Skalabilnost s količinom dostupnih podataka
- Mogućnost modeliranja kompleksnih nelinearnih odnosa

LSTM (Long Short-Term Memory) mreže su posebna vrsta povratnih neuronskih mreža (RNN) dizajniranih za rješavanje problema nestajućeg i eksplodirajućeg gradijenta koji se javljaju kod standardnih RNN-ova pri obradi dugih sekvenci podataka. Ključna inovacija LSTM-a je memorijska ćelija koja može pamtiti informacije kroz dulje vremenske periode, zajedno s mehanizmima "vrata" koji kontroliraju protok informacija:

1. **Vrata zaboravljanja (Forget Gate)** – odlučuju koje informacije treba zadržati, a koje odbaciti:

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

2. **Ulazna vrata (Input Gate)** – određuju koliko novih informacija ulazi u ćeliju:

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

3. **Ažuriranje stanja ćelije (Cell State Update)** – kombinacija prethodnih vrijednosti i novih podataka:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

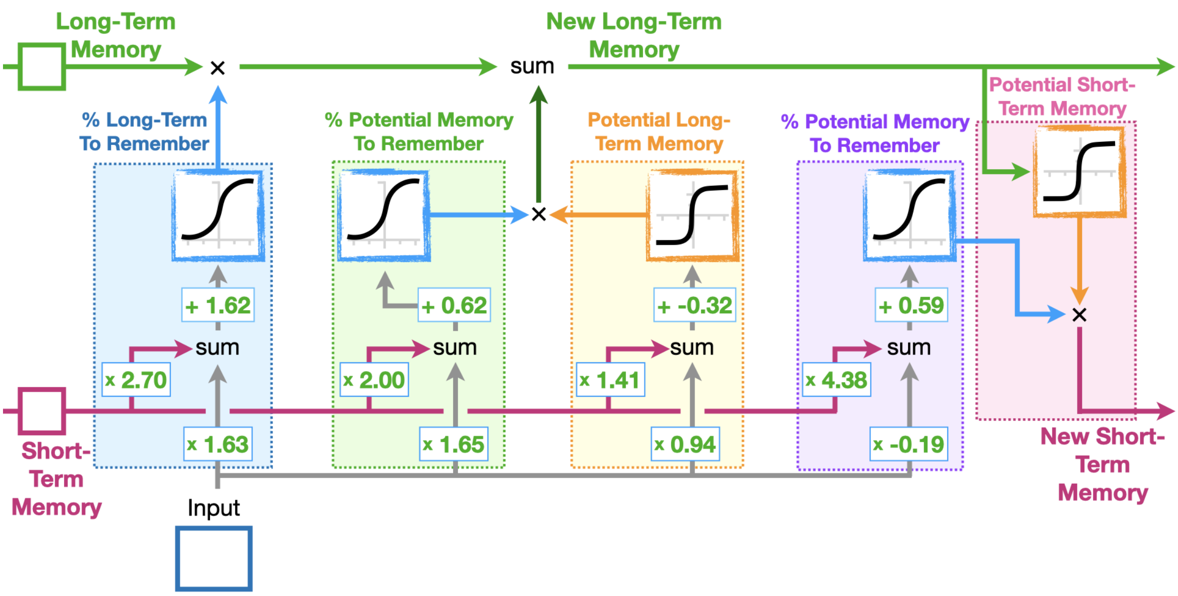
4. **Izlazna vrata (Output Gate)** – kontroliraju koliko informacija iz ćelije utječe na izlaz:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

5. **Ažuriranje izlaza (Hidden State Update)** – definiranje konačnog izlaza na temelju stanja ćelije:

$$h_t = o_t \odot \tanh(C_t)$$

Ova arhitektura čini LSTM posebno pogodnim za financijske vremenske serije gdje dugoročne ovisnosti mogu imati značajan utjecaj na buduće kretanje cijena.



Slika 3.1. LSTM arhitektura. [4]

3.2. Arhitektura modela

Predloženi model, nazvan *PortfolioLSTM*, koristi Long Short-Term Memory (LSTM) neuronsku mrežu za učenje optimalne alokacije imovine na temelju vremenskih nizova povrata. Za razliku od tradicionalnih metoda koje se oslanjaju na eksplicitnu procjenu statističkih mjera poput srednje vrijednosti i varijance, ovaj model koristi funkciju cilja koja izravno optimizira portfelj, omogućujući modelu da uči pravilne težine kroz podatke i zadani cilj (npr. maksimizaciju Sharpeovog omjera).

Model se sastoji od sljedećih ključnih komponenti:

1. **LSTM sloj**

   Prvi sloj modela je LSTM sloj koji prima ulazne vremenske nizove (npr. povrate imovine) dimenzije `(batch_size, sequence_length, input_size)`. Ovaj sloj omogućava učenje sekvencijalnih obrazaca i dugoročnih ovisnosti u podacima. Model podržava višeslojne LSTM konfiguracije kroz parametar `num_layers`, a regularizacija putem `dropout` opcionalno se primjenjuje između LSTM slojeva kada je broj slojeva veći od jedan.

2. **Mehanizam pozornosti (attention)**

   Za dodatno naglašavanje važnih vremenskih trenutaka u sekvenci, model uključuje jednostavan mehanizam pozornosti. Ova komponenta koristi dvije potpuno povezane (linearne) transformacije s Tanh aktivacijom između njih kako bi se izračunala važnost svakog vremenskog koraka u skrivenom stanju. Time se omogućuje modelu da fokusira pažnju na značajne dijelove vremenske serije.

3. **Dropout sloj**

   Dropout se primjenjuje nakon LSTM sloja kako bi se smanjio rizik od preučenosti modela. Budući da LSTM mreže sadrže velik broj parametara, postoji opasnost da model nauči i zapamti šumove ili slučajne obrasce prisutne u podacima, umjesto da prepozna općenite zakonitosti. Tijekom treniranja dropout nasumično isključuje određeni postotak neuronskih jedinica, čime se potiče razvoj robusnijih i stabilnijih reprezentacija. Ovakav pristup posebno je važan kod financijskih podataka, jer oni često sadrže mnogo šuma, nestabilnosti i kratkoročnih fluktuacija. Primjena dropouta doprinosi tome da model ne interpretira te slučajne promjene kao značajne obrasce, čime se poboljšava njegova sposobnost generalizacije na nove, neviđene podatke.

4. **Potpuno povezani sloj (fully connected layer)**

   Izlaz iz attention sloja (ili posljednjeg LSTM koraka, ovisno o implementaciji) zatim se prosljeđuje kroz potpuno povezani sloj koji transformira skriveni vektor u vektor težina portfelja. Veličina izlaznog vektora odgovara broju imovina u portfelju (`output_size`).

5. **Softmax aktivacijska funkcija / Skaliranje**

   Kako bi se osiguralo da je zbroj težina portfelja jednak jedan (što je standardni uvjet upravljanja portfeljima), koristi se Softmax funkcija ili skaliranje nad izlaznim vektorom.

   Softmax funkcija omogućava samo duge pozicije. Duga pozicija označava kupnju financijske imovine s ciljem ostvarivanja dobiti kroz rast njezine cijene. Ulagatelj zarađuje ako cijena poraste.

$$w_i = \text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

   Skaliranje omogućava i duge i kratke pozicije. Kratka pozicija podrazumijeva prodaju posuđene imovine uz očekivanje pada cijene, kako bi je kasnije otkupio jeftinije i ostvario dobit.

$$w_i = \frac{z_i}{\sum_j z_j}$$

   gdje su: w_i normalizirane težine, z_i sirove težine.

$$s_i = \min(w_i, 0)$$

   gdje je: s_i kratke (negativne) težine.

$$S = \sum_j |s_j| + \varepsilon$$

gdje je: S zbroj apsolutnih vrijednosti kratkih pozicija

$$\alpha = \min\left(1, \frac{\text{max\_shorting}}{S}\right)$$

gdje su: α skalar za ograničenje kratkih pozicija,

max_shorting maksimalno dopušteno kratko izlaganje.

$$w_i' = \begin{cases} \alpha \cdot w_i & w_i < 0 \\ w_i & \text{inače} \end{cases}$$

gdje je: w_i' skalirane težine.

$$w_i'' = \frac{w_i'}{\sum_j w_j'}$$

gdje je: w_i'' konačno normalizirane težine.

Time se dobiva distribuirana alokacija kapitala među svim raspoloživim imovinama.

Ova arhitektura kombinira snagu vremenskog modeliranja LSTM mreže s fokusiranjem pozornosti i end-to-end optimizacijom portfelja, bez potrebe za tradicionalnim statističkim pretpostavkama. Time se otvara prostor za robustan i prilagodljiv sustav upravljanja portfeljem, sposoban za učenje iz kompleksnih tržišnih obrazaca.

3.3. Inicijalizacija težina

Kako bi se poboljšala konvergencija tijekom treniranja neuronske mreže i izbjegli problemi poput eksplodirajućih ili iščezavajućih gradijenata, u modelu se koristi **Xavier-Glorot inicijalizacija** težina. Ova metoda posebno je prikladna za slojeve s linearnim ili Tanh aktivacijama, a osmišljena je kako bi očuvala varijancu signala kroz slojeve mreže.

Za svaki linearni sloj (uključujući LSTM parametre i slojeve pozornosti), težine se inicijaliziraju nasumično iz uniforme distribucije:

$$W \sim \mathcal{U}\left(-\frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}, \frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}\right)$$

gdje su n_{in} i n_{out} broj ulaznih i izlaznih jedinica sloja.

Svi bias parametri postavljeni su na nulu:

$$b = 0$$

Ova inicijalizacija osigurava stabilan protok informacija kroz mrežu i doprinosi učinkovitijem učenju, posebno u složenim modelima.

3.4. Funkcije cilja

Model je treniran s ciljem maksimizacije različitih funkcija cilja kako bi se omogućilo ulaganje raznim vrstama investitora.

3.4.1. Sharpeov omjer

Formula za Sharpeov omjer je:

$$Sharpe = \frac{R_p - R_f}{\sigma_p}$$

gdje je:

- R_p — povrat portfelja,
- R_f — bezrizična kamatna stopa,
- σ_p — standardna devijacija (volatilnost) povrata portfelja.

3.4.2. Sortinov omjer

Formula za Sortinov omjer je:

$$\text{Sortino} = \frac{R_p - R_f}{\sigma_{\text{downside}}}$$

gdje je:

- R_p — povrat portfelja,
- R_f — bezrizična kamatna stopa,
- σ_{downside}— standardna devijacija negativnih povrata.

3.4.3. Funkcija korisnosti (Utility function)

Funkcija korisnosti može se zapisati kao:

$$U = (R_p - R_f) - \frac{1}{2}\lambda\,\sigma_p^2$$

gdje je:

- U — korisnost (utility) portfelja,
- R_p — povrat portfelja,
- R_f — bezrizična kamatna stopa,
- λ — koeficijent averzije prema riziku,
- σ_p — standardna devijacija povrata portfelja.

3.4.4. Jednostavna funkcija korisnosti

Jednostavna funkcija korisnosti može se zapisati kao:

$$U = (R_p - R_f) - \lambda \sigma_p$$

3.4.5. Povrat portfelja

$$R_p = \sum_{i=1}^{n} w_i R_i$$

gdje je:

- R_p — povrat portfelja,
- w_i — udio (težina) imovine i u portfelju,
- R_i — povrat imovine i,
- n — ukupan broj imovina u portfelju.

3.4.6. Eksponencijalno ponderirana funkcija povrata (EWR)

Eksponencijalno ponderirana funkcija povrata je state-of-the-art mjera koja koristi eksponencijalno opadajuće težine za naglašavanje lošijih prinosa u distribuciji, čime se modelira povećana osjetljivost na negativne ishode. Težine ovise o relativnoj poziciji u distribuciji, a funkcija je glatka i diferencijabilna, što je čini pogodnom za optimizaciju gradijentnim metodama.

Formula za eksponencijalno ponderiranu funkciju povrata je:

$$
\begin{aligned}
&R_{(1)} \le R_{(2)} \le \cdots \le R_{(n)} \quad \text{(sortirani prinosi)} \\
&p_i = \frac{i}{n} \quad \text{(pozicija u distribuciji)} \\
&w_i = \exp\Big(-\frac{\max(0,\ p_i - \alpha)}{s}\Big) \quad \text{(eksponencijalne težine)} \\
&\tilde{w}_i = \frac{w_i}{\sum_{j=1}^{n} w_j} \quad \text{(normalizacija težina)} \\
&\mathrm{EWR} = 252 \sum_{i=1}^{n} \tilde{w}_i \, R_{(i)}.
\end{aligned}
$$

Parametri su:

$$\alpha = 0.05, \quad s = 0.01.$$

Faktor 252 koristi se za anualizaciju (252 je prosječni broj radnih dana tržišta).

Diferencijabilnost funkcije sortiranja

Sortiranje tehnički nije diferencijabilno u strogo matematičkom smislu jer se redoslijed elemenata može naglo promijeniti kada se njihove vrijednosti jako približe ili izjednače. Međutim, u praktičnim implementacijama (PyTorch) sortiranje samo reorganizira elemente bez promjene njihovih stvarnih vrijednosti pa se gradijenti mogu kontinuirano računati i vraćati natrag prema svakom ulazu. Drugim riječima, vrijednosti ostaju glatko povezane s originalnim podacima, dok se samo njihov položaj mijenja, što rijetko uzrokuje problem pri optimizaciji. Dok god se uzimaju glatke funkcije tih vrijednosti (npr. eksponencijalne težine), sve se može optimizirati metodama gradijenta.

Intuitivno, to se može zamisliti kao da slaganje elemenata po veličini samo određuje red u kojem ih gledate, ali ne "lomi" samu liniju podataka: gradijent zna odakle je svaka vrijednost došla. Tek kad se pokuša napraviti nagli rez (npr. uzeti samo određeni prag i sve ostalo ignorirati), funkcija postaje nediferencijabilna. Ako se umjesto toga primijene glatko opadajuće težine, sve ostaje kontinuirano i pogodno za učenje.

U ovom radu, bezrizična kamatna stopa R_f definirana je na temelju prinosa američkog trezorskog zapisa s rokom dospijeća od 3 mjeseca (*3-month Treasury bill* ili skraćeno **T-bill**). T-bill je kratkoročni dužnički vrijednosni papir koji izdaje američka vlada i smatra se gotovo potpuno bezrizičnim instrumentom zbog vrlo male vjerojatnosti neispunjenja obveza. Njegova stopa prinosa često se koristi kao procjena R_f u financijskim modelima.

U svaku od ovih funkcija dodani su trošak transakcije (kako bi evaluacija bila realna) i L2 regularizacija (kazna za visoke vrijednosti težina portfelja kako bi se model natjeralo na diverzifikaciju).

$$\text{L2\_penalty} = 0.001 \sum_{i=1}^{n} w_i^2$$

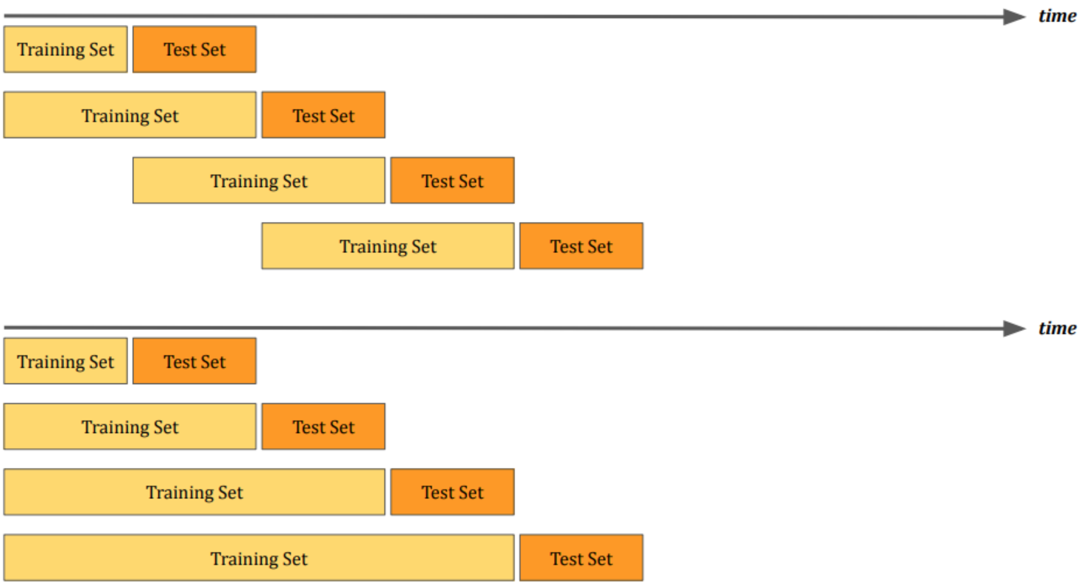
3.5. Treniranje modela

Treniranje modela i postupak backtestinga provedeno je s posebnim naglaskom na izbjegavanje curenja podataka i osiguravanje što realnijeg okvira evaluacije. Za svaku godinu koju evaluiramo, model je treniran ispočetka koristeći samo podatke koji bi u tom trenutku povijesti bili dostupni. Na taj način, prilikom testiranja simuliramo situaciju kao da smo zaista u toj godini i ne poznajemo buduće podatke, čime se osigurava da procjena prediktivnih sposobnosti modela ne bude pristrana.

Za odabir podataka za treniranje implementirana su dva pristupa: *klizeći prozor* (engl. sliding window) i *proširujući prozor* (engl. expanding window):

- **Klizeći prozor** koristi samo unaprijed definirani broj zadnjih godina podataka za treniranje, bez obzira na to koliko podataka je ukupno dostupno. Ovaj pristup ograničava memoriju modela na recentnija kretanja i sprječava potencijalni utjecaj zastarjelih obrazaca.

- **Proširujući prozor** podrazumijeva da za svaku novu godinu proširujemo skup podataka za treniranje tako da uključuje sve dostupne povijesne podatke od početka do godine prije trenutne evaluacijske godine.



Slika 3.2. Vrste backtestinga. [5]

U ovom radu korišten je proširujući prozor zbog relativno ograničene povijesti podataka (manje od 20 godina).

Model je treniran optimizatorom Adam, čiji je postupak ažuriranja parametara definiran sljedećom formulom:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t$$

pri čemu su:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t},$$

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t,$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2,$$

gdje je g_t gradijent funkcije cilja u trenutnom koraku, η je početna stopa učenja, a tipične vrijednosti parametara su $\beta_1 = 0.9$, $\beta_2 = 0.999$ i $\epsilon = 10^{-8}$.

Broj epoha treniranja bio je 50, uz početnu stopu učenja $\eta = 0.001$. Dodatno je implementiran sustav ranog zaustavljanja i prilagodbe stope učenja kako bi se smanjila mogućnost prenaučenosti:

- Ako se vrijednost funkcije cilja na skupu za treniranje ne poboljša tijekom 5 uzastopnih epoha, stopa učenja se dijeli s 10, te postaje $\eta = 0.0001$.
- Ako se i nakon ove promjene još jednom dogodi ista situacija (još 5 epoha bez poboljšanja), treniranje za tu godinu se zaustavlja.

Ovim pristupom osigurano je da model ne bude pretjerano prilagođen trenutačnim podacima i da ujedno zadrži sposobnost generalizacije na nepoznate podatke.

3.6. Validacija modela

Validacija modela provedena je isključivo radi određivanja optimalnog broja epoha treniranja. Postupak je bio sljedeći: za svaku godinu evaluacije (na primjer, za 2010. godinu), prvo se trenirao posebni model koristeći sve dostupne podatke iz godina koje prethode godini validacije (u ovom primjeru do zaključno 2008. godine). Taj model treniran je kroz maksimalno 50 epoha. Nakon svake epohe mjerena je vrijednost funkcije cilja na podacima iz godine neposredno prije evaluacijske godine (u ovom slučaju 2009.) koja je služila isključivo kao validacijski skup.

Nakon završenih 50 epoha, kao optimalan broj epoha odabran je onaj za koji je funkcija cilja na validacijskom skupu bila najveća. Potom se trenirao novi model s istim parametrima, ali ovaj put nad cijelim skupom podataka koji uključuje i godinu koja je prije bila korištena samo za validaciju (dakle, u primjeru s 2010. godinom treniranje je provedeno nad podacima do zaključno 2009.). Ovaj model treniran je upravo onoliko epoha koliko je prethodno određeno kao optimalno, čime je osigurano da izbor broja epoha ne ovisi o podacima budućih godina i da se izbjegne pristranost u procjeni performansi.

Međutim, ovakav tip validacije u praksi se nije pokazao posebno korisnim jer se validacija provodi na podacima iz godine koja često ima drugačiju prirodu i strukturu u odnosu na promatranu godinu evaluacije što je ograničavalo primjenjivost rezultata. Uz to, cijeli postupak značajno je produljio vrijeme treniranja modela pa se u većini rezultata (osim u onima kojima je cilj pokazati da validacija ne čini razliku) nije ni koristio.

3.7. Evaluacija

Evaluacija modela provodi se na godišnjoj razini, pri čemu se za svaku godinu trenira novi model na povijesnim podacima i testira na tekućoj godini. Ključne metrike evaluacije uključuju:

- **Sharpeov omjer**: Mjeri godišnju uspješnost ulaganja prilagođenu riziku kao omjer viška povrata i ukupne volatilnosti.

$$\text{Sharpe} = \frac{R_p - R_f}{\sigma_p}$$

- **Sortinov omjer**: Mjeri godišnju uspješnost ulaganja uzimajući u obzir samo štetnu volatilnost.

$$\text{Sortino} = \frac{R_p - R_f}{\sigma_{\text{downside}}}$$

- **Godišnji povrat**: Predstavlja ukupan postotak zarade ili gubitka od ulaganja kroz jednogodišnje razdoblje. Godišnji povrat može se izračunati kao:

$$R_{\text{godišnji}} = \frac{V_{\text{kraj}} - V_{\text{početak}}}{V_{\text{početak}}} \times 100\%,$$

  gdje su $V_{\text{početak}}$ i V_{kraj} početna i završna vrijednost portfelja u godini.

- **Standardna devijacija**: Mjeri godišnju volatilnost povrata, indicirajući razinu rizika povezanog s ulaganjima tijekom godine. Računa se prema formuli:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(r_i - \bar{r})^2},$$

  gdje je r_i povrat u periodu i, $\bar{r}$ srednji povrat, a N broj perioda unutar godine.

- **Maksimalni pad vrijednosti (Maximum Drawdown)**: Predstavlja najveći zabilježeni postotak pada vrijednosti portfelja od vrhunca do najniže točke.

- **Maksimalni dnevni pad (Maximum Daily Drawdown)**: Označava najveći jednodnevni gubitak vrijednosti portfelja unutar promatranog razdoblja.

4. Implementacija

4.1. Priprema podataka

Model koristi povijesne povrate indeksa iz datoteke '.csv'. Podaci za treniranje i testiranje modela preuzeti su pomoću yfinance biblioteke u Pythonu te su manualno procesirani za potrebe modela. Za analizu su korišteni podaci počevši od datuma kada je najkasnije uvedena imovina u portfelju postala dostupna (21.6.2006.), čime je osigurana konzistentnost skupa podataka.

4.1.1. Vrste imovina

U ovom radu koriste se isključivo dnevni povrati temeljeni na ETF-ovima za sve promatrane klase imovine.

Analiza je provedena na temelju sljedećih imovina:

- američke dionice (SPY),
- kratkoročne državne obveznice (SHY),
- srednjoročne državne obveznice (IEF),
- dugoročne državne obveznice (TLT),
- korporativne obveznice (LQD),
- zlato (GLD),
- nekretnine (VNQ),
- nafta (USO),

- tržišta u razvoju (VWO),
- dionice eurozone Euro Stoxx 50 (FEZ).

4.1.2. Indikatori

Korišteni su sljedeći indikatori:

- VIX (indeks volatilnosti), koji mjeri očekivanu volatilnost tržišta na temelju cijena opcija.
- Prinos američkog trezorskog zapisa (*T-bill*), koji predstavlja referentnu bezrizičnu kamatnu stopu.
- Odnos dvaju pomičnih prosjeka za svaku imovinu:

$$\mathrm{MA}_k(t) = \frac{1}{k}\sum_{i=0}^{k-1} P_{t-i},$$

  gdje je P_t cijena na dan t, a indikator je definiran kao:

$$\frac{\mathrm{MA}_{21}(t) - \mathrm{MA}_{50}(t)}{\mathrm{MA}_{50}(t)}.$$

- Povijesni Sharpeov omjer u razdoblju od 50 dana za svaku imovinu:

$$\mathrm{Sharpe}_{50}(t) = \frac{\overline{R}_{50}(t) - r_f}{\sigma_{50}(t)},$$

  gdje je:

$$\overline{R}_{50}(t) = \frac{1}{50}\sum_{i=0}^{49} R_{t-i},$$

  $\sigma_{50}(t)$ standardna devijacija prinosa u istom razdoblju, a r_f prinos bez rizika.

- 21-dnevni indeks relativne snage (RSI):

$$\mathrm{RSI}_{21}(t) = 100 - \frac{100}{1 + RS(t)},$$

gdje je:

$$RS(t) = \frac{\text{Srednji dobitak u 21 danu}}{\text{Srednji gubitak u 21 danu}}.$$

4.2. Hiperparametri

Implementacija koristi sljedeće hiperparametre:

Veličina ulaza (`input_size`): jednaka broju imovina u datasetu, tj. 10

Razdoblje promatranja (`lookback`): 20 ili 40 dana

Razdoblje predviđanja (`lookahead`): 3 ili 5 dana

Veličina skrivenog sloja (`hidden_size`): 128

Stopa učenja (`learning_rate`): 0.001

Broj epoha (`num_epochs`): 50

Dropout rate: 0.5

Broj godina koje ne sudjeluju u testiranju, tj. koje služe za inicijalno treniranje: 3

Batch size: 4

Averzija prema riziku (samo za neke funkcije cilja): -5, 0.5, 1 ili 5

Maksimalna kratka pozicija (`max_shorting`): 0 ili 0.5

Funkcija cilja: sharpe, sortino, funkcija korisnosti, jednostavna funkcija korisnosti, povrati, eksponencijalno ponderirani povrati (EWR), povrati + EWR ili sharpe + EWR

Validacija: 0 ili 1

Korištenje softmax funkcije: 0 (skaliranje) ili 1 (softmax)

Trošak transakcije: 0.0005

5. Rezultati

Model je evaluiran usporedbom sa svim klasama imovina, jednako ponderiranim portfeljem (koji predstavlja jednostavnu benchmark strategiju) i povijesno-optimiziranim portfeljem (metoda koja maksimizira zadanu funkciju cilja u prošlom lookback razdoblju s obzirom na povijesne podatke i takve težine portfelja prenosi u budućnost).

5.1. Usporedba neuronske mreže sa softmax funkcijom i neuronske mreže sa skaliranjem kao posljednjim slojem

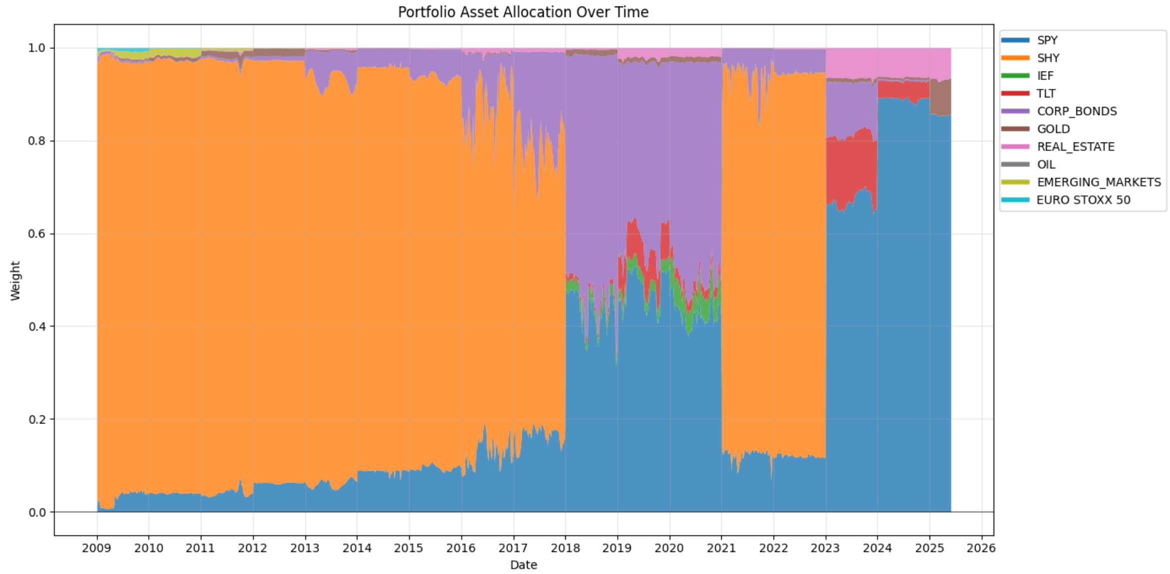
Usporedba modela koji kao zadnji sloj koristi softmax funkciju i modela koji kao zadnji sloj koristi skaliranje bez mogućnosti kratke pozicije:

Tablica 5.1. Usporedba softmax i skaliranje.

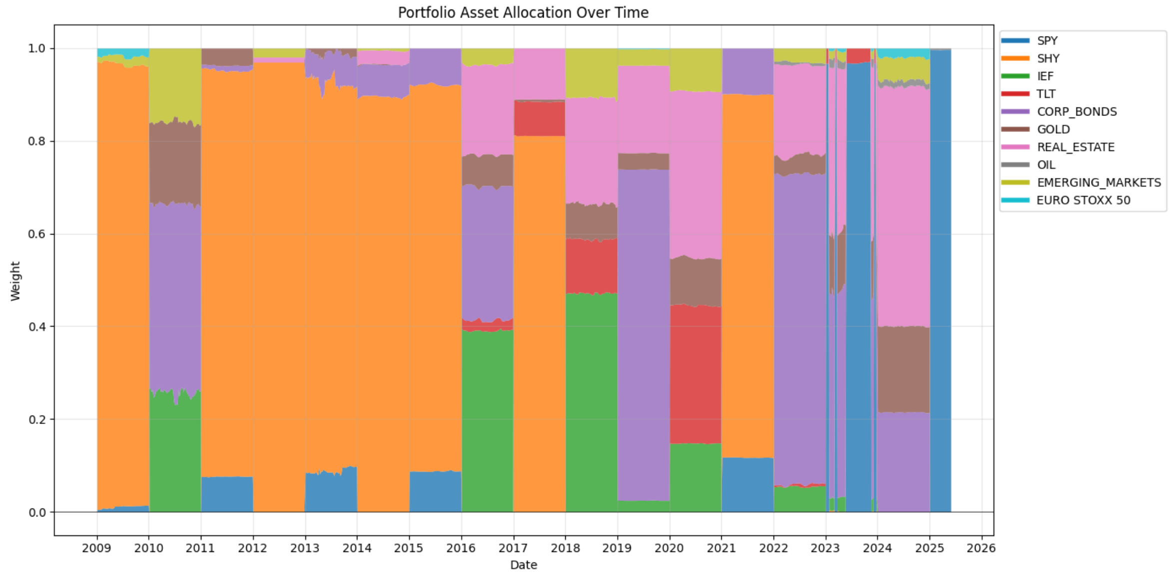
	Annual return	**Annual std dev**	**Sharpe**	**Sortino**	**Max daily DD**	**Max DD**
Sharpe softmax	0.0564	0.0589	1.1133	1.6383	0.0657	0.2287
Sharpe skaliranje	0.0413	0.0676	0.8491	1.1553	0.0624	0.2403
EWR softmax	0.0874	0.2117	0.4193	0.6251	0.1773	0.4240
EWR skaliranje	0.1015	0.1836	0.7640	1.0383	0.1135	0.3585

Rezultati pokazuju da je primjena softmax sloja kao završnog sloja učinkovitija za optimizaciju Sharpeovog omjera, dok se pristup skaliranja pokazao pogodnijim za optimizaciju EWR-a. Usporedba ova dva pristupa zahtijeva dodatna istraživanja, osobito u kontekstu različitih funkcija cilja, kako bi se dobio sveobuhvatniji uvid u njihove performanse i primjenjivost.

Sharpeov omjer

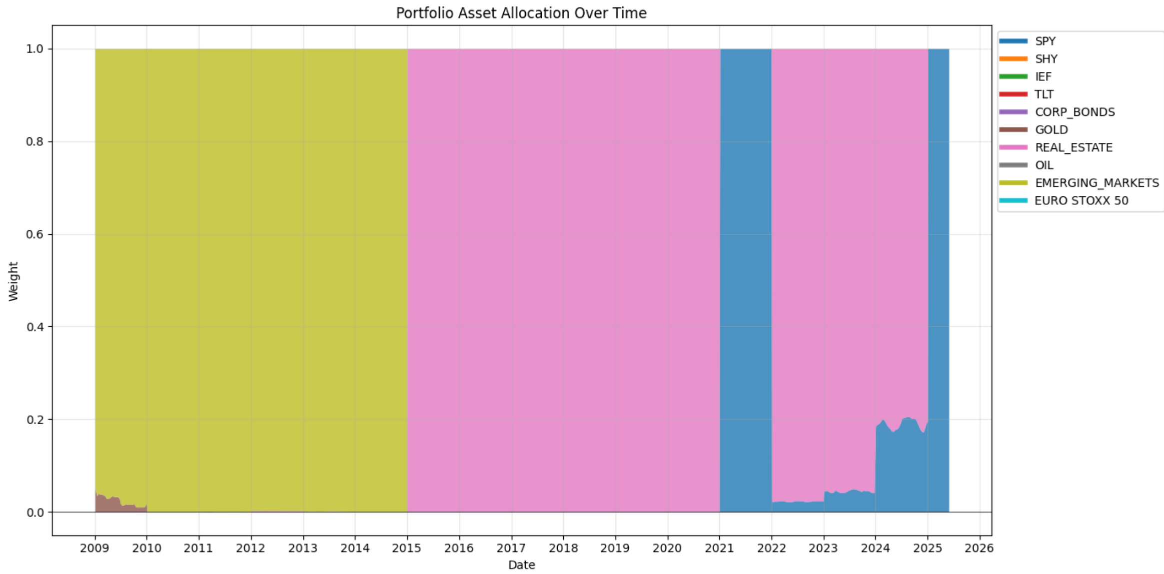


Slika 5.1. Sharpe – optimiziran pomoću softmax sloja.

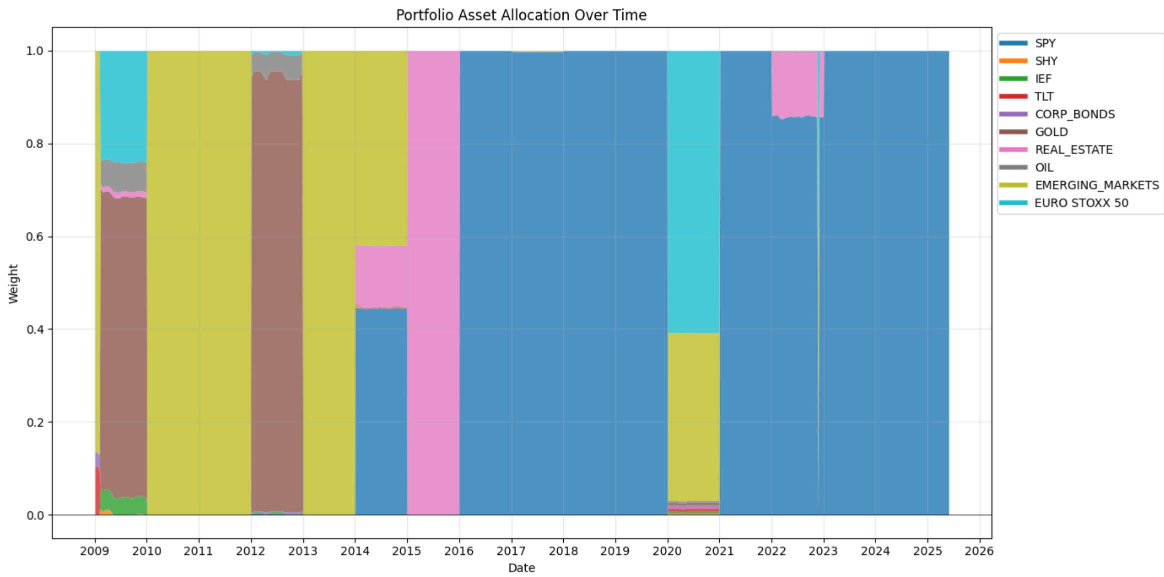


Slika 5.2. Sharpe – optimiziran pomoću skaliranja.

EWR



Slika 5.3. EWR – optimiziran pomoću softmax sloja.



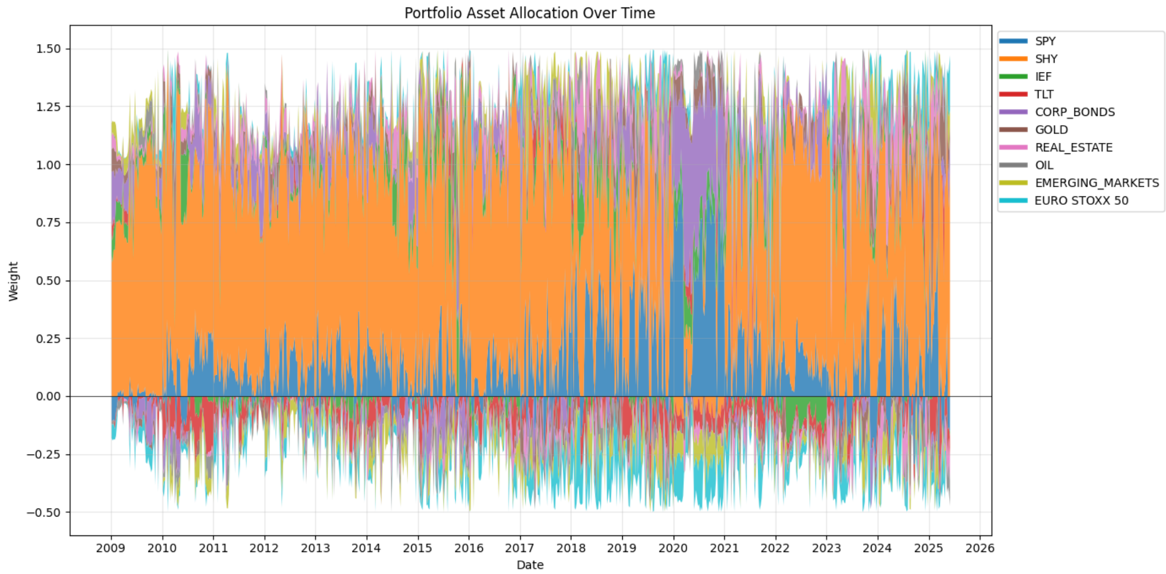
Slika 5.4. EWR – optimiziran pomoću skaliranja.

5.2. Učinak indikatora na treniranje neuronske mreže

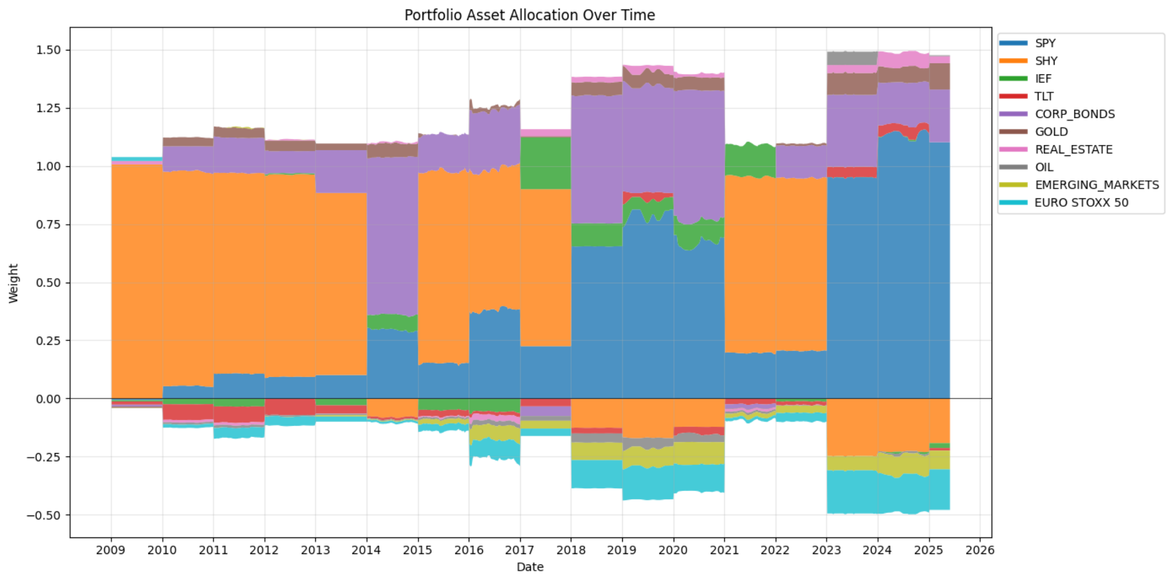
Sharpeov omjer

Tablica 5.2. Učinak indikatora – Sharpe.

	Annual return	Annual std dev	Sharpe	Sortino	Max daily DD	Max DD
Bez indikatora	0.0265	0.0580	0.2568	0.4170	0.0668	0.2124
S indikatorima	0.0742	0.0682	1.1857	1.7211	0.0665	0.2262



Slika 5.5. Težine portfelja – treniranje Sharpeovog omjera bez indikatora.

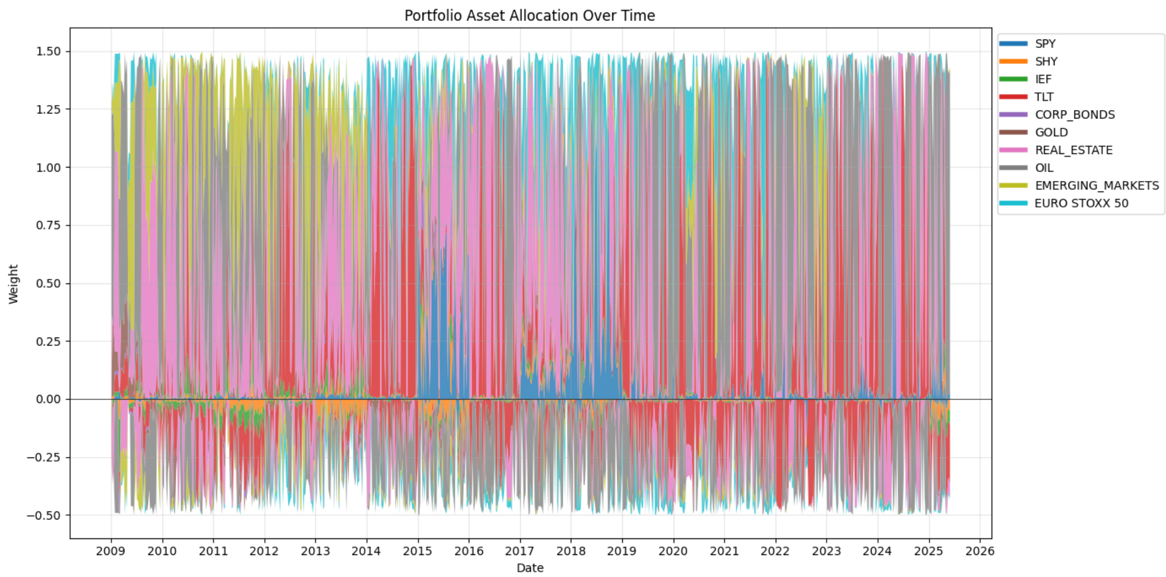


Slika 5.6. Težine portfelja – treniranje Sharpeovog omjera s indikatorima.

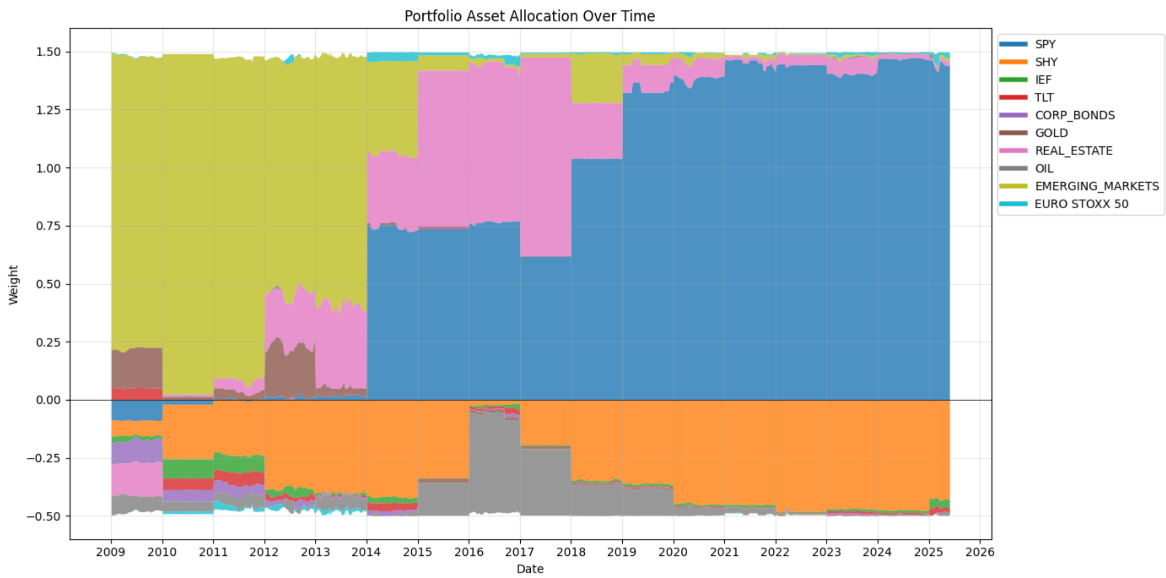
EWR

Tablica 5.3. Učinak indikatora – EWR

	Annual return	Annual std dev	Sharpe	Sortino	Max daily DD	Max DD
Bez indikatora	0.0773	0.3250	0.3146	0.5642	0.2244	0.6689
S indikatorima	0.1782	0.2707	0.7729	1.1244	0.1663	0.4661



Slika 5.7. Težine portfelja – treniranje EWR omjera bez indikatora.



Slika 5.8. Težine portfelja – treniranje EWR omjera s indikatorima.

Teorijski, neuronska mreža bi trebala biti sposobna samostalno prepoznati korisne obrasce i indikatore, poput pomičnog prosjeka, povijesnog Sharpeovog omjera ili indeksa relativne snage, kao i eventualno otkriti dodatne relevantne značajke. Međutim, u praksi se pokazuje da to nije slučaj – vidljive su značajne razlike u rezultatima kada se mreži eksplicitno dodaju poznati korisni indikatori u usporedbi s treniranjem isključivo na temelju povrata.

Činjenica da model postiže znatno bolje i stabilnije rezultate kada se indikatori uključe u skup podataka dodatno potvrđuje da ne dolazi do curenja podataka (engl. *data leakage*).

5.3. Učinak validacije na treniranje neuronske mreže

Tablica 5.4. Učinak validacije na treniranje neuronske mreže.

	Annual return	**Annual std dev**	**Sharpe**	**Sortino**	**Max daily DD**	**Max DD**
Sharpe bez validacije	0.0742	0.0682	1.1857	1.7211	0.0665	0.2262
Sharpe s validacijom	0.0753	0.0891	1.0686	1.5231	0.0706	0.2416
EWR bez validacije	0.1782	0.2707	0.7729	1.1244	0.1663	0.4661
EWR s validacijom	0.1841	0.2222	0.8467	1.2112	0.1439	0.3748
Povrati bez validacije	0.1406	0.2642	0.6017	0.8935	0.1524	0.4185
Povrati s validacijom	0.1436	0.2290	0.7265	1.0737	0.1600	0.4457

Validacija broja epoha primijenjena u ovom radu pokazala je nestabilne rezultate te se nije pokazala kao pouzdan način za postizanje dosljednog poboljšanja performansi modela. Uzrok tome leži u činjenici da se validacija provodi na godini koja prethodi testnom razdoblju, a koja se, zbog specifičnih tržišnih uvjeta i kompleksnosti, razlikuje od testne godine. Posljedično, optimalan broj epoha potreban za treniranje modela varira iz godine u godinu, što ukazuje na ograničenja ovakvog pristupa validaciji. Stoga je potrebno provesti dodatna istraživanja usmjerena na alternativne metode validacije koje bi bile robusnije u kontekstu promjenjivih tržišnih uvjeta.

5.4. Učinak korištenja troška transakcije u funkciji gubitka na treniranje neuronske mreže

Tablica 5.5. Učinak troška transakcije pri treniranju.

	Annual return	Annual std dev	Sharpe	Sortino	Max daily DD	Max DD
Sharpe bez TT	0.0861	0.0851	1.0644	1.5644	0.0664	0.2265
Sharpe s TT	0.0726	0.0814	1.0867	1.5949	0.0653	0.2213
EWR bez TT	0.1583	0.2635	0.7595	1.1211	0.1654	0.4687
EWR s TT	0.1782	0.2707	0.7729	1.1244	0.1663	0.4661

Rezultati upućuju na to da uključivanje transakcijskih troškova u funkciju cilja tijekom treniranja modela donosi blago poboljšanje performansi.

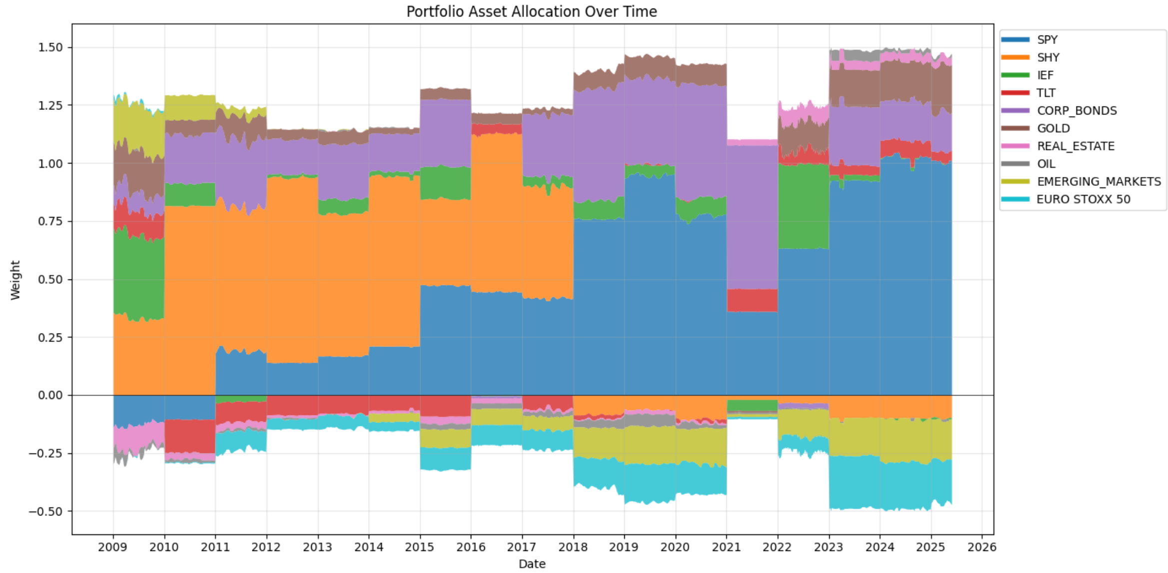
5.5. Analiza stabilnosti modela

Neuronske mreže često se smatraju svojevrsnim black box modelima jer je njihovo ponašanje ponekad teško interpretirati. Zbog inherentne nestabilnosti i osjetljivosti na ulazne podatke, može se dogoditi da model generira neočekivane ili neadekvatne predikcije, što u kontekstu financija može rezultirati značajnim gubicima i takozvanim crnim događajima. Upravo zato posebnu pažnju treba posvetiti procesu dizajniranja modela i izbora podataka, kako bi se smanjio rizik od neželjenih posljedica. U ovom radu uspješno je razvijen stabilan LSTM model za optimizaciju portfelja, koji pokazuje konzistentne performanse i povećava pouzdanost predikcija u dinamičnim tržišnim uvjetima. U nastavku prikazujem rezultate modela kroz grafove koji ilustriraju njegovu stabilnost i prediktivnu točnost.

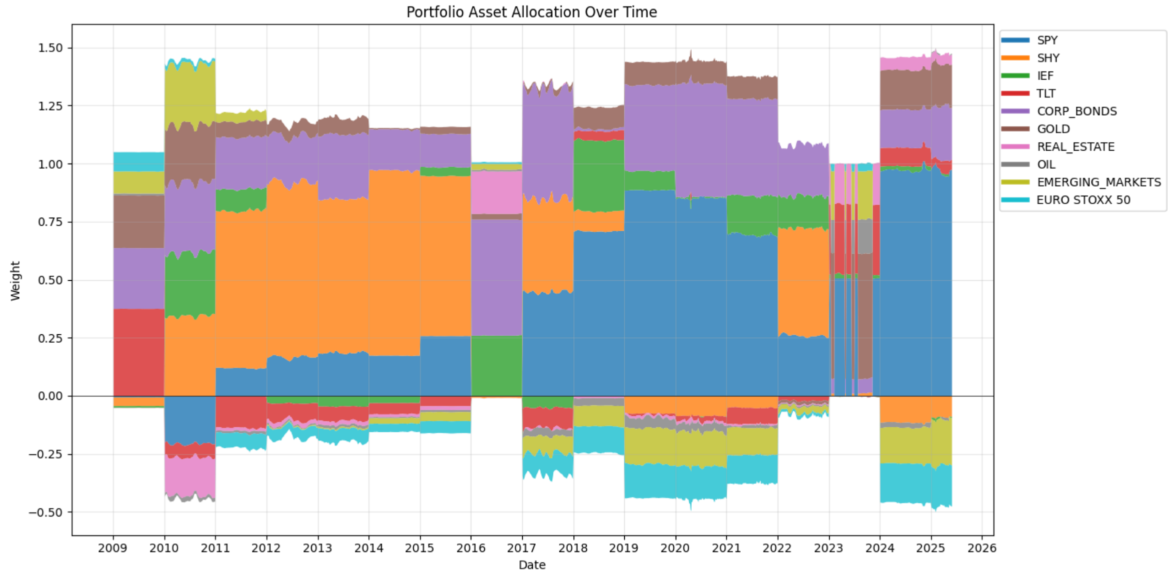
Istrenirana su i evaluirana dva različita modela koji se razlikuju samo po hiperparametrima lookback i lookahead:

- Model_S (lookback=20, lookahead=3)
- Model_L (lookback=40, lookahead=5)

Model_S – Sharpe



Slika 5.9. Težine portfelja Model_S – Sharpe iteracija 1.

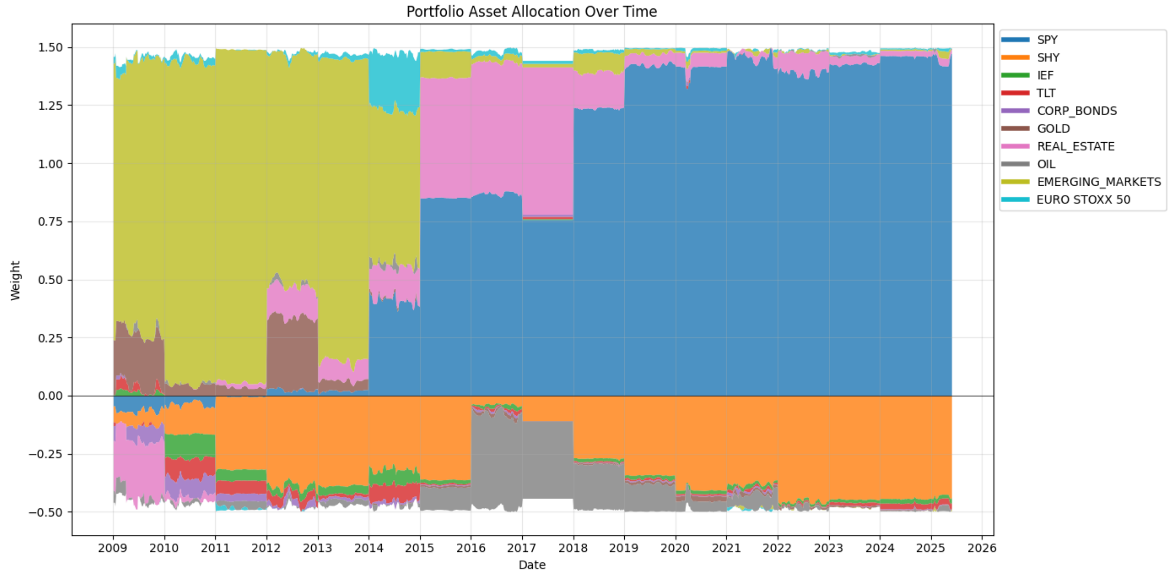


Slika 5.10. Težine portfelja Model_S – Sharpe iteracija 2.

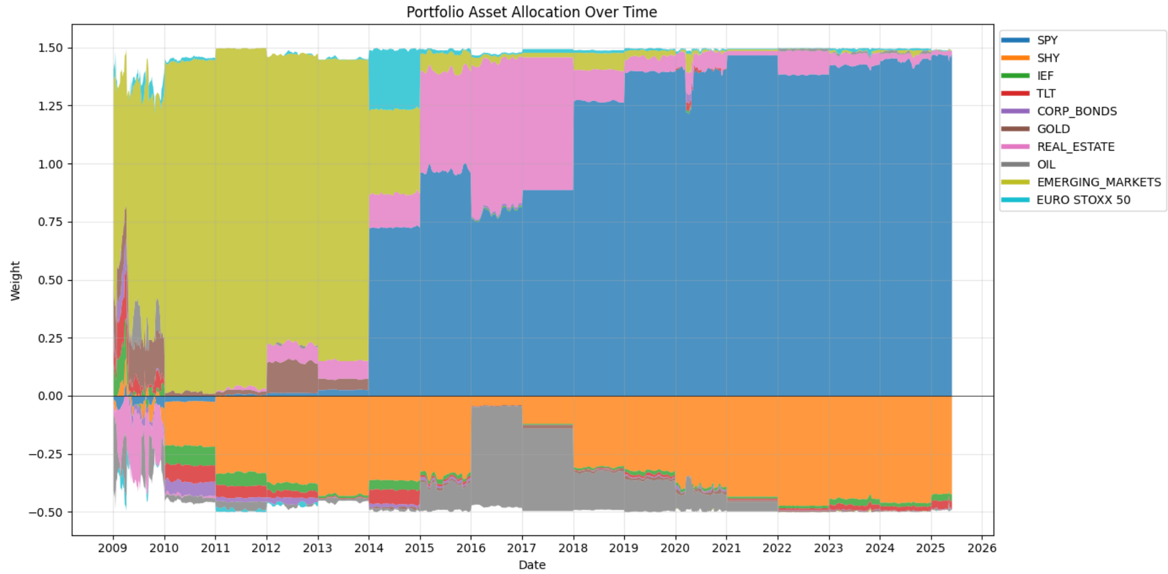
Tablica 5.6. Iteracije treniranja Sharpeovog omjera.

Annual return	Annual std dev	Sharpe	Sortino	Max daily DD	Max DD
0.0726	0.0814	1.0867	1.5949	0.0653	0.2213
0.0824	0.0815	1.0474	1.5523	0.0736	0.2468

Model_S – EWR



Slika 5.11. Težine portfelja Model_S – EWR iteracija 1.



Slika 5.12. Težine portfelja Model_S – EWR iteracija 2.

Tablica 5.7. Iteracije treniranja EWR.

Annual return	Annual std dev	Sharpe	Sortino	Max daily DD	Max DD
0.1590	0.2673	0.7190	1.0498	0.1573	0.4495
0.1548	0.2657	0.7297	1.0619	0.1493	0.4328

5.6. Analiza različitih funkcija cilja

Za svaku funkciju cilja istrenirana su i evaluirana dva različita modela koji se razlikuju samo po hiperparametrima lookback i lookahead:

- Model_S (lookback=20, lookahead=3)
- Model_L (lookback=40, lookahead=5)

5.6.1. Tablica svih rezultata

Model S

Tablica 5.8. Tablica svih rezultata Model_S.

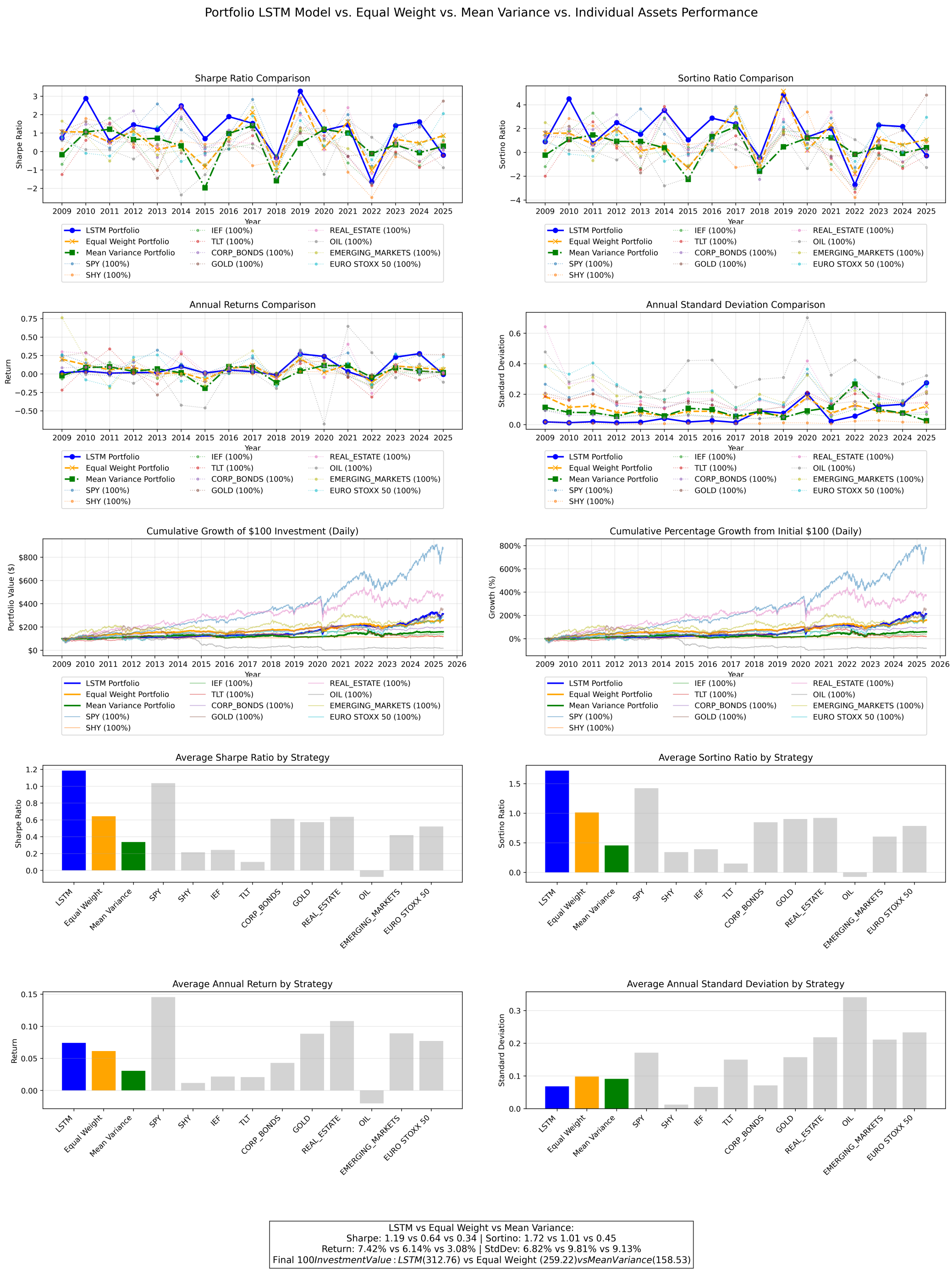
Loss function	Annual return	Annual std dev	Sharpe	Sortino	Max daily DD	Max DD
Sharpe	0.0726	0.0814	1.0867	1.5949	0.0653	0.2213
Sharpe	0.0824	0.0815	1.0474	1.5523	0.0736	0.2468
EWR	0.1590	0.2673	0.7190	1.0498	0.1573	0.4495
EWR	0.1548	0.2657	0.7297	1.0619	0.1493	0.4328
Sortino	0.1120	0.3170	0.3934	0.5565	0.2004	0.5935
Simple utility $\lambda = -5$	0.1083	0.3350	0.3307	0.4778	0.2119	0.6456
Simple utility $\lambda = 0.5$	0.1369	0.2373	0.6698	0.9794	0.1273	0.3780
Simple utility $\lambda = 1$	0.1193	0.1866	0.7971	1.1494	0.1224	0.3375
Simple utility $\lambda = 5$	0.0150	0.0106	0.5162	0.8212	0.0060	0.0509
Utility $\lambda = -5$	0.1156	0.3161	0.3993	0.5836	0.2643	0.5839
Utility $\lambda = 0.5$	0.1600	0.2571	0.6941	1.0313	0.1279	0.3778
Utility $\lambda = 1$	0.1559	0.2417	0.7445	1.0933	0.1225	0.3636
Utility $\lambda = 5$	0.1272	0.1695	0.7957	1.1453	0.0794	0.3303
Returns + EWR	0.1513	0.2678	0.7189	1.0498	0.1643	0.4569
Sharpe + EWR	0.0854	0.0816	1.1042	1.6558	0.0684	0.2206
Returns	0.1491	0.2743	0.5955	0.8674	0.1496	0.4309

Model L

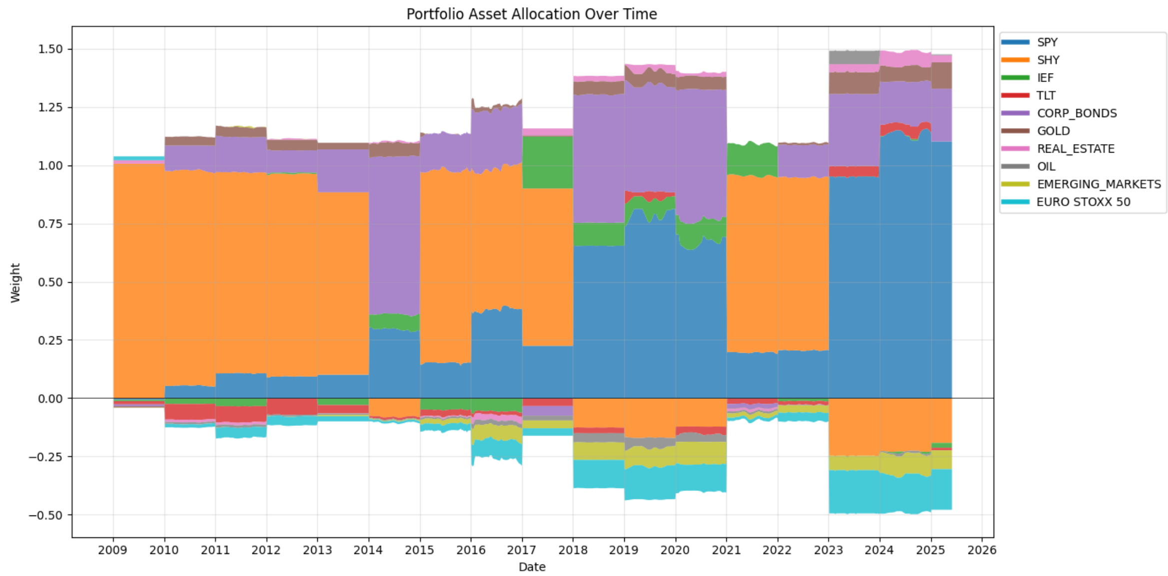
Tablica 5.9. Tablica svih rezultata Model_L.

Loss function	Annual return	Annual std dev	Sharpe	Sortino	Max daily DD	Max DD
Sharpe	0.0768	0.0729	1.1612	1.6991	0.0671	0.2292
Sharpe	0.0742	0.0682	1.1857	1.7211	0.0665	0.2262
EWR	0.1494	0.2607	0.7400	1.0800	0.1636	0.4650
EWR	0.1782	0.2707	0.7729	1.1244	0.1663	0.4661
Sortino	0.1222	0.1926	0.8359	1.1378	0.1254	0.3679
Simple utility $\lambda = -5$	0.1043	0.3301	0.3297	0.4760	0.2131	0.6380
Simple utility $\lambda = 0.5$	0.1499	0.2264	0.7570	1.1150	0.1210	0.3922
Simple utility $\lambda = 1$	0.0922	0.1491	0.8070	1.1848	0.0953	0.2857
Simple utility $\lambda = 5$	0.0111	0.0122	0.2532	0.4506	0.0120	0.0517
Utility $\lambda = -5$	0.1425	0.3105	0.5690	0.8412	0.2868	0.6022
Utility $\lambda = 0.5$	0.1567	0.2519	0.7184	1.0613	0.1336	0.3762
Utility $\lambda = 1$	0.1447	0.2428	0.6713	0.9866	0.1278	0.3778
Utility $\lambda = 5$	0.1275	0.1660	0.8764	1.2886	0.1165	0.3306
Returns + EWR	0.1645	0.2644	0.7857	1.1536	0.1648	0.4671
Sharpe + EWR	0.0950	0.1122	1.1329	1.6445	0.0767	0.2521
Returns	0.1406	0.2642	0.6017	0.8935	0.1524	0.4185

5.6.2. Sharpeov omjer



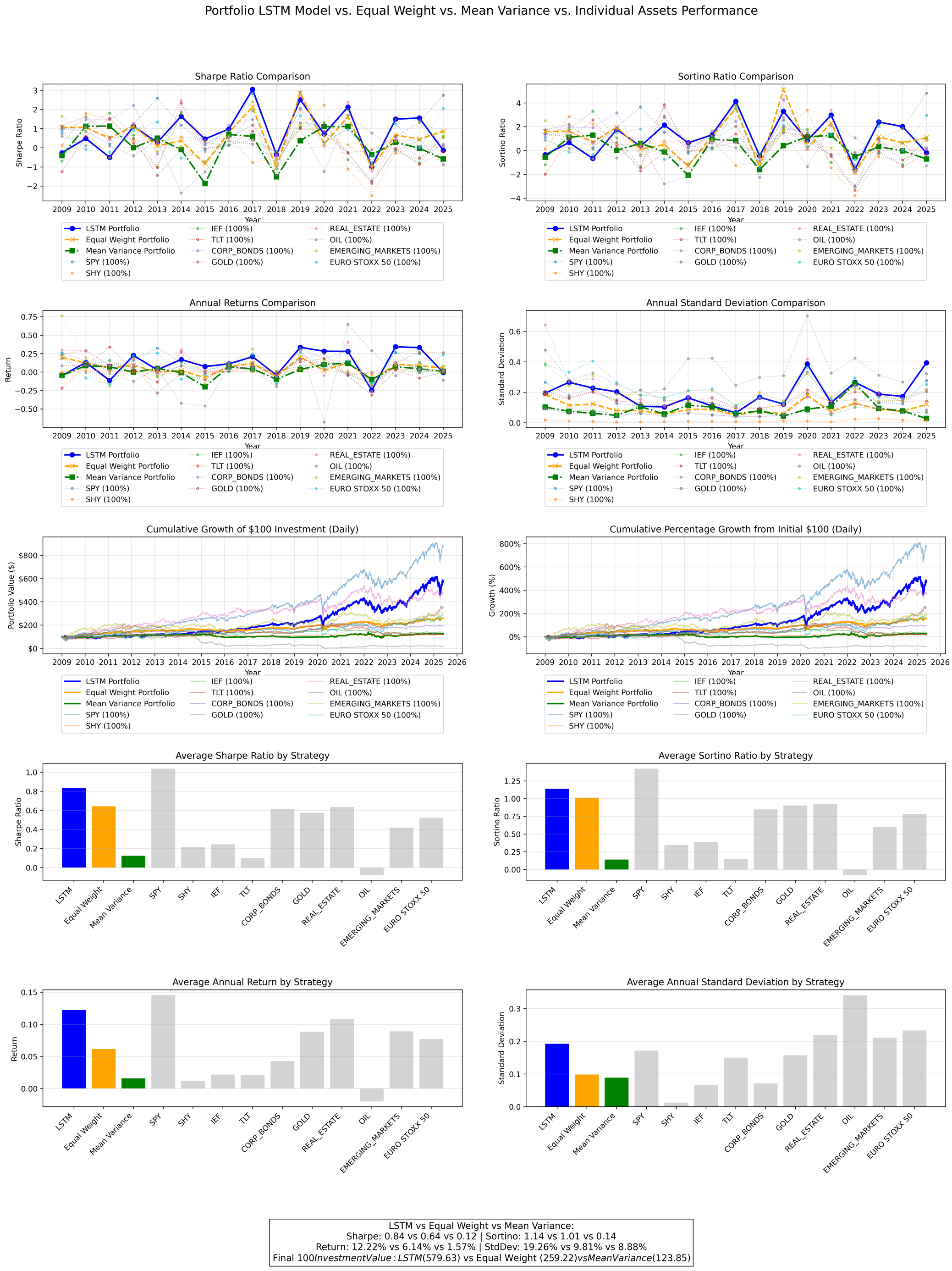
Slika 5.13. Sharpeov omjer Model_L.



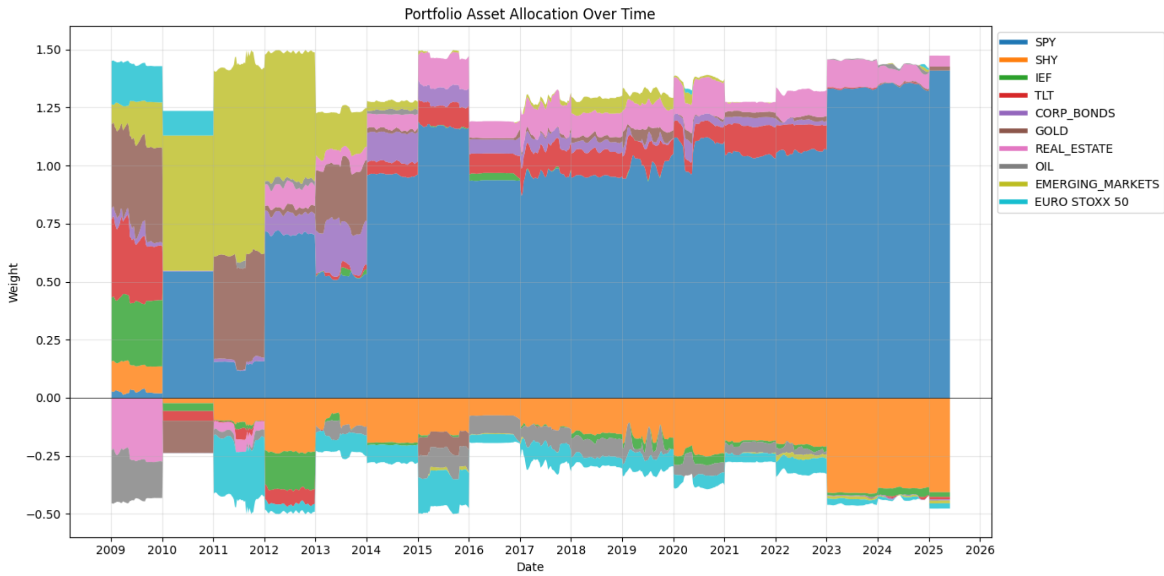
Slika 5.14. Težine portfelja Model_L – Sharpe.

Treniranjem modela s ciljem maksimizacije Sharpeovog omjera postignut je Sharpeov omjer od 1.19 i Sortinov omjer od 1.72, čime je tijekom razdoblja dužeg od 15 godina model nadmašio sve ostale klase imovine i referentne (benchmark) modele prema tim kriterijima.

5.6.3. Sortinov omjer



Slika 5.15. Sortinov omjer Model_L.



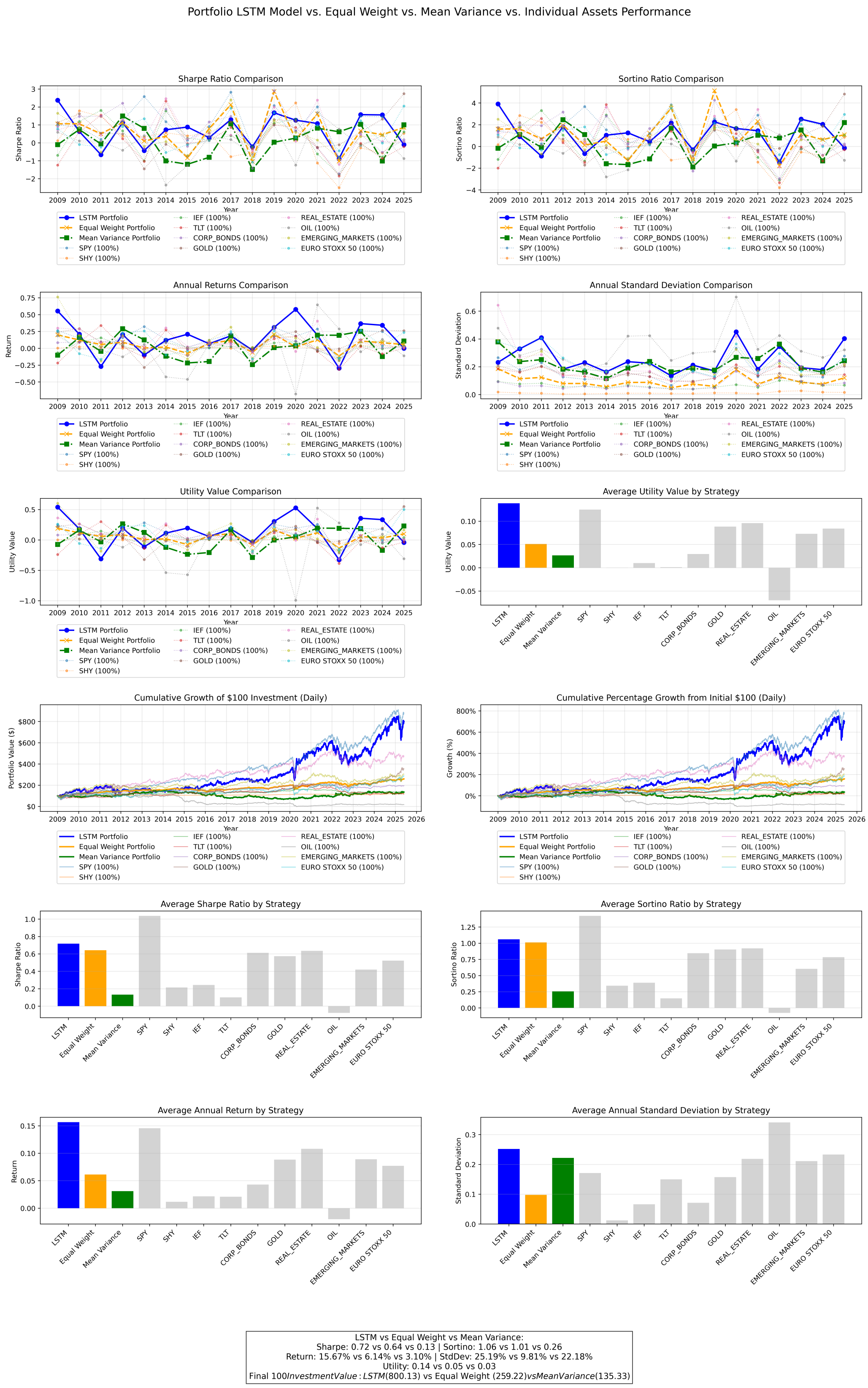
Slika 5.16. Težine portfelja Model_L – Sortino.

Treniranjem modela s ciljem maksimizacije Sortinovog omjera postignut je Sortinov omjer od 1.14, čime je tijekom razdoblja dužeg od 15 godina model nadmašio sve ostale klase imovine i referentne (benchmark) modele osim SPY prema tom kriteriju.

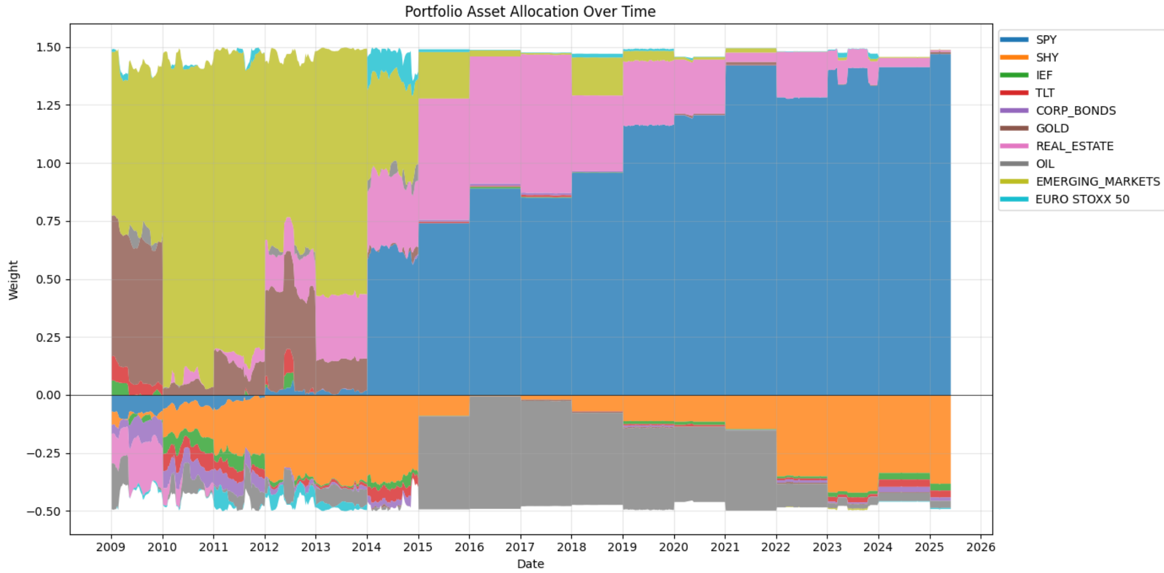
Optimizacija Sortinovog omjera pokazala se manje učinkovitom u odnosu na optimizaciju Sharpeovog omjera, ponajprije zbog tehničkih ograničenja prilikom izračuna downside standardne devijacije. Naime, u slučajevima kada u lookahead periodu nema negativnih povrata, downside devijacija postaje nedefinirana, što dovodi do nestabilnosti modela.

5.6.4. Utility Function

Averzija prema riziku = 0.5

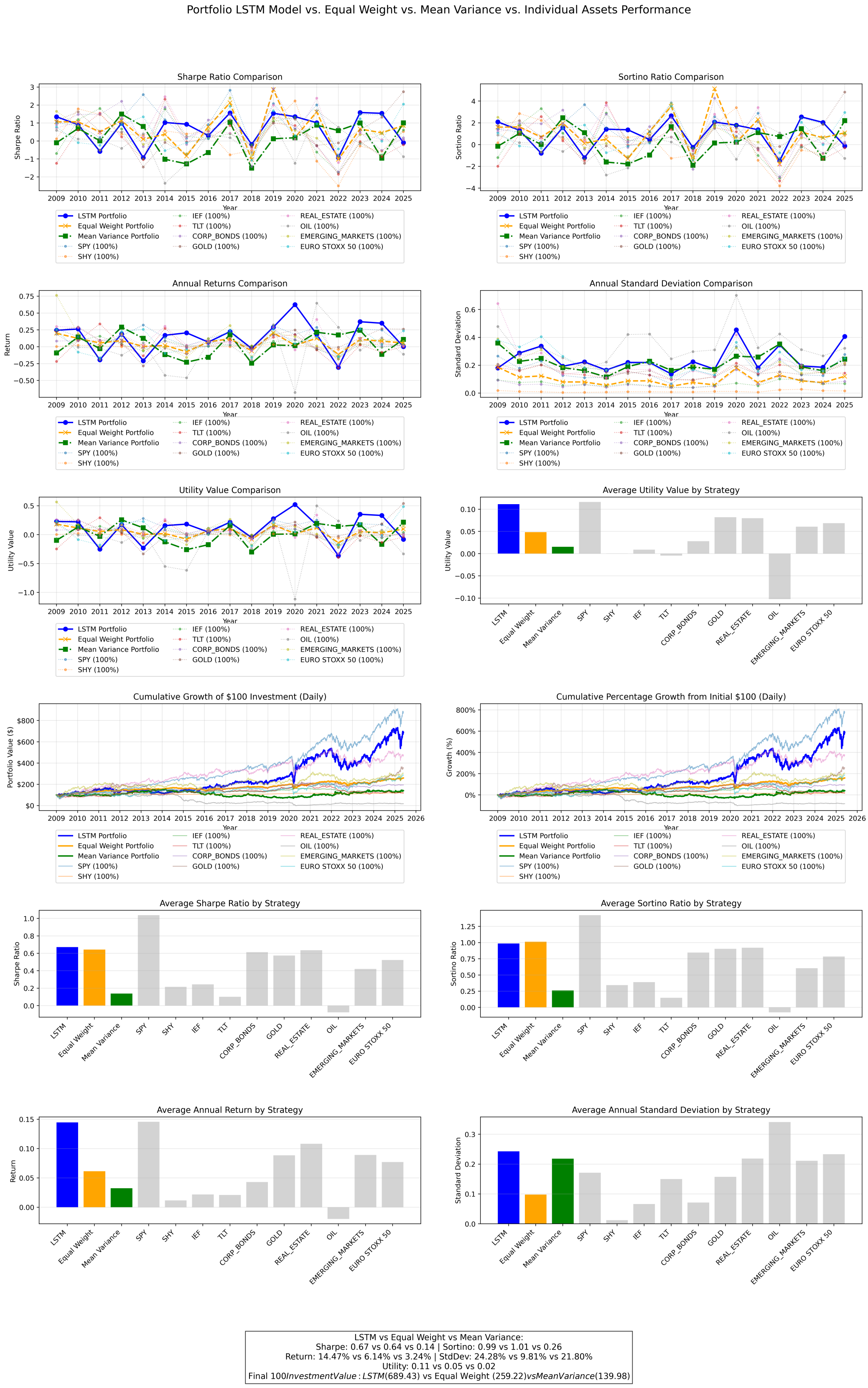


Slika 5.17. Utility Function 0.5 Model_L.

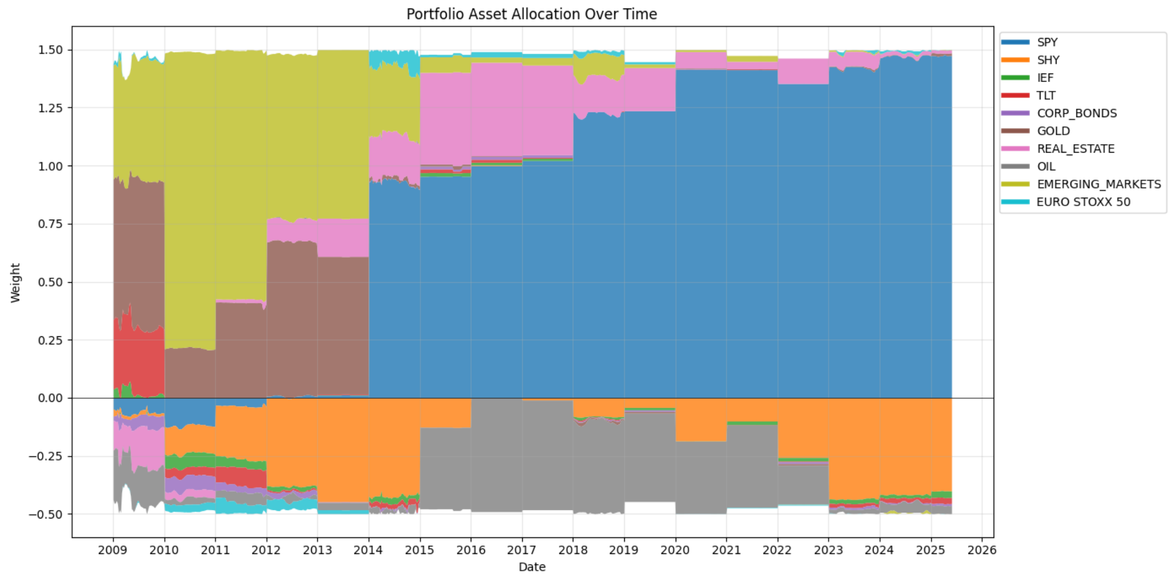


Slika 5.18. Težine portfelja Model_L – Utility Function 0.5.

Averzija prema riziku = 1

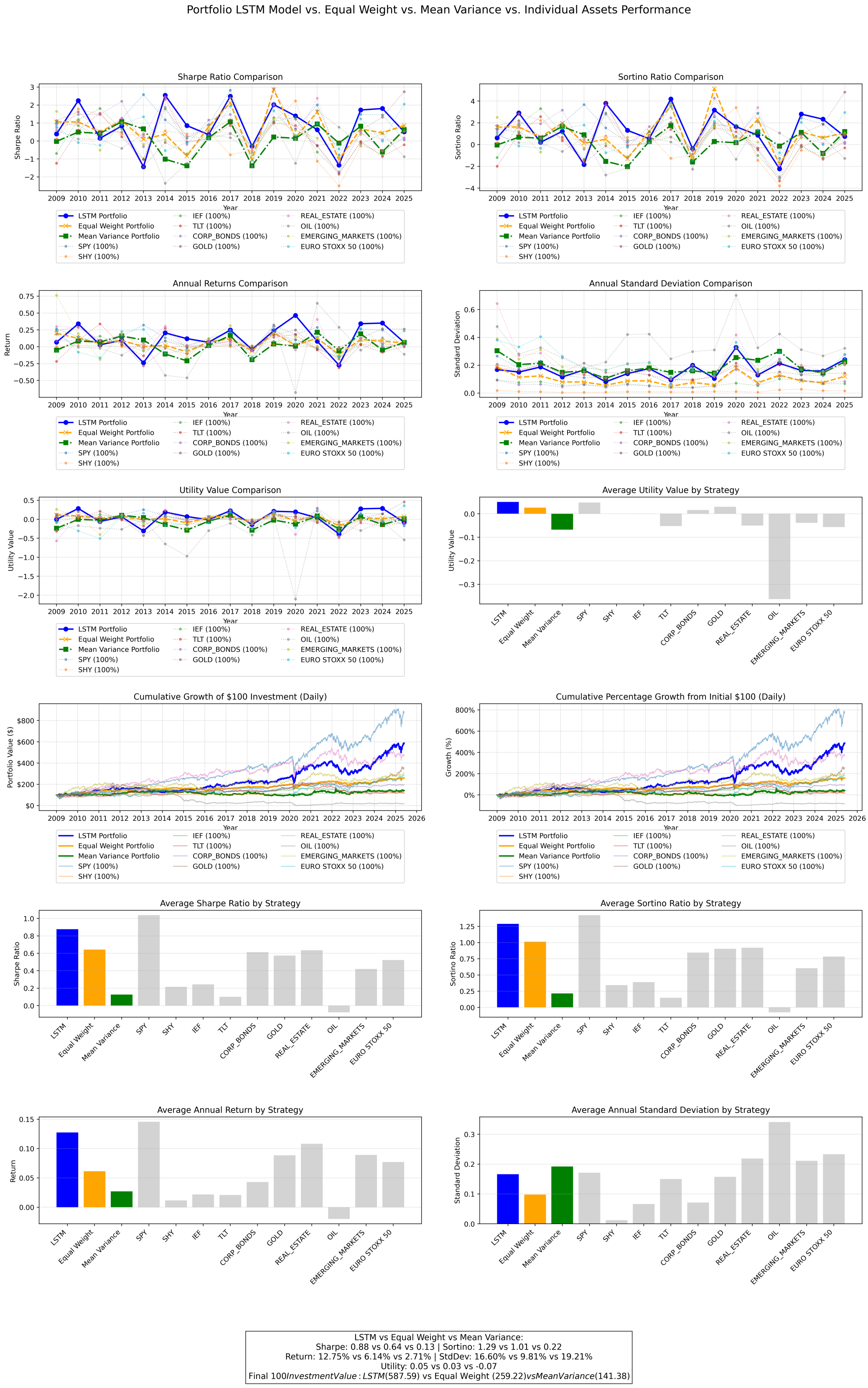


Slika 5.19. Utility Function 1 Model_L.

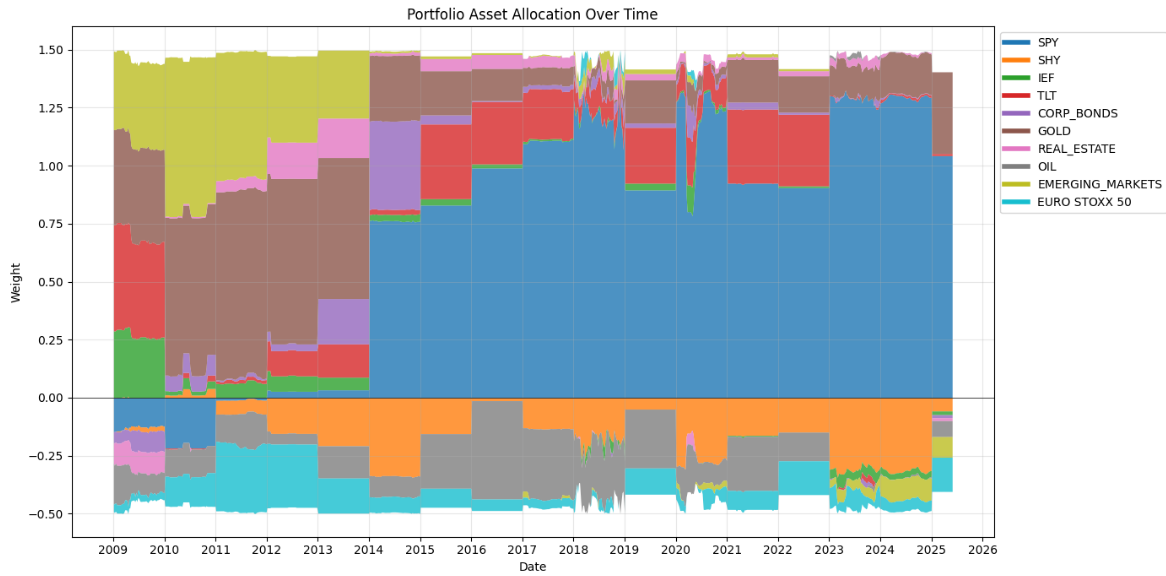


Slika 5.20. Težine portfelja Model_L – Utility Function 1.

Averzija prema riziku = 5

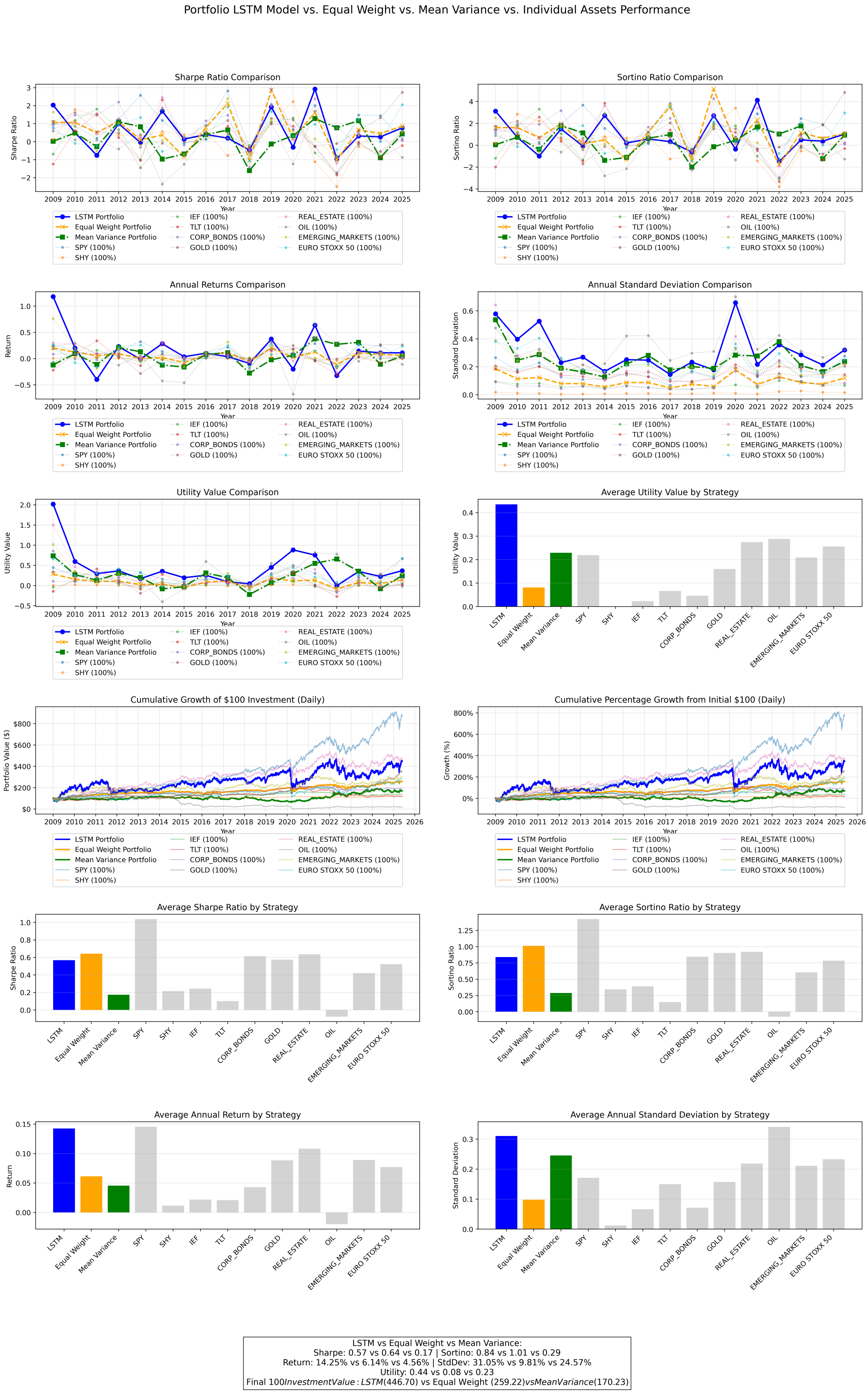


Slika 5.21. Utility Function 5 Model_L.

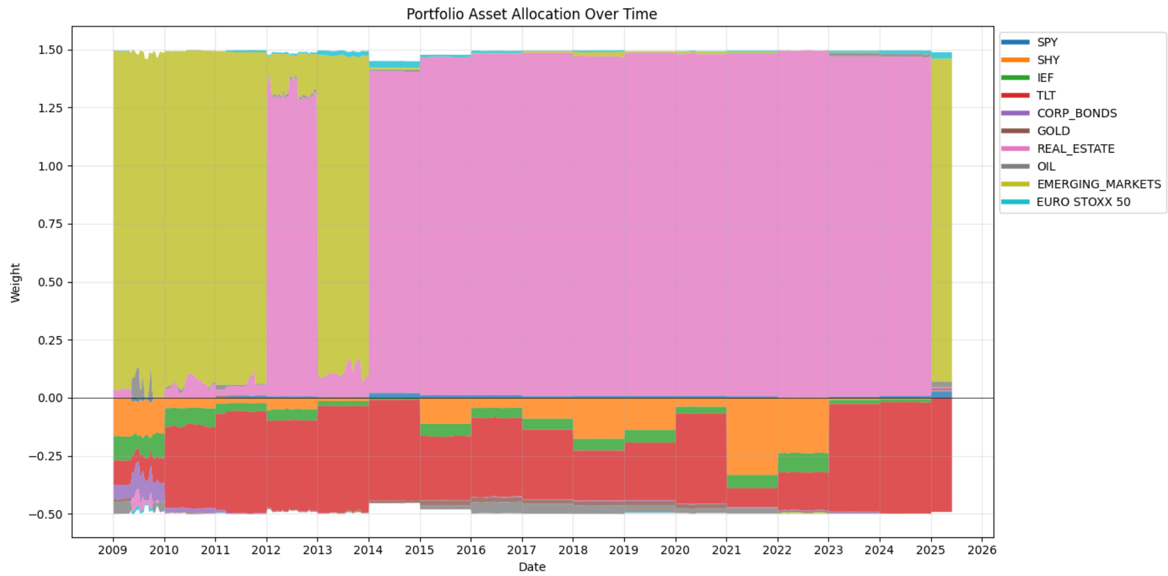


Slika 5.22. Težine portfelja Model_L – Utility Function 5.

Averzija prema riziku = -5



Slika 5.23. Utility Function -5 Model_L.



Slika 5.24. Težine portfelja Model_L – Utility Function -5.

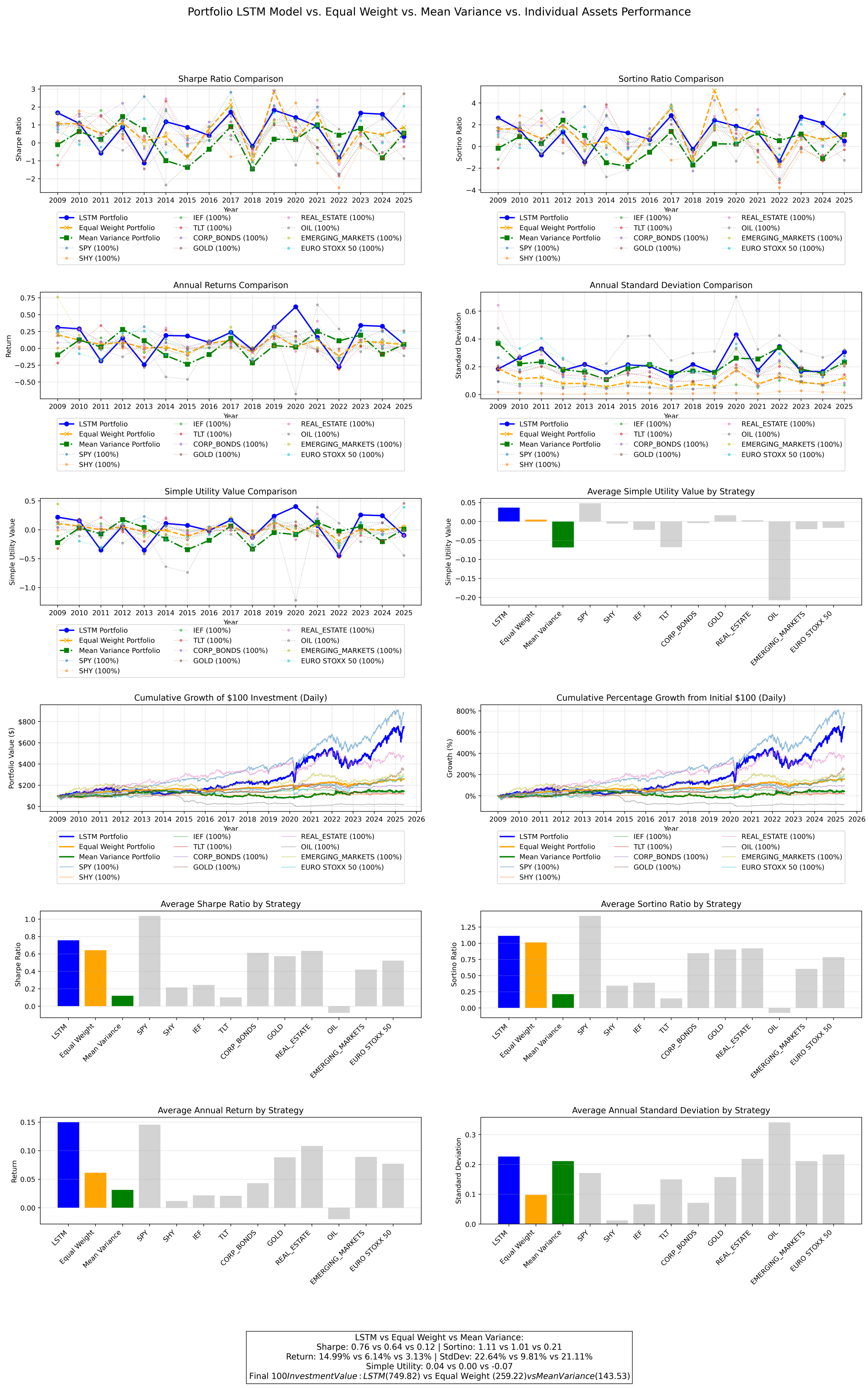
Rezultati pokazuju da maksimizacija funkcije korisnosti (Utility) daje konkurentne rezultate – prema samoj funkciji korisnosti, tijekom razdoblja od 15 godina, model je nadmašio sve ostale klase imovine i referentne portfelje.

Model treniran na ovaj način pokazuje sposobnost prilagodbe razini rizika koju investitor može tolerirati. Na primjer, za koeficijent averzije prema riziku jednak 0,5, ostvareni su prosječni godišnji povrat od 16%, standardna devijacija (rizik) od 25% te maksimalni pad vrijednosti portfelja (Maximum Drawdown) od 38%. Kako averzija prema riziku raste, smanjuju se i rizik i očekivani prinosi: pri averziji 1, godišnji povrat iznosi 14%, standardna devijacija 24%, a maksimalni pad vrijednosti 38%, dok za averziju 5 povrat iznosi 13%, uz standardnu devijaciju od 17% i maksimalni pad od 33%.

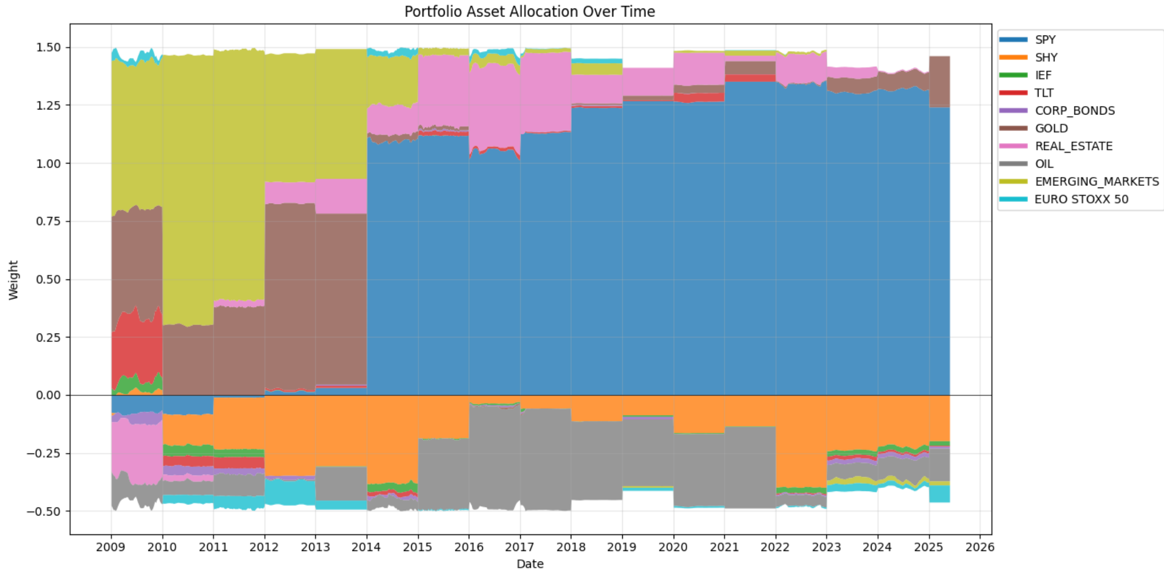
Dodatno, model je treniran i s negativnim koeficijentom averzije prema riziku jednakom -5, kako bi se istražila mogućnost maksimizacije rizika za investitore sklone preuzimanju visokog rizika. U tom slučaju, ostvareni su godišnji povrat od 14%, standardna devijacija od 31% i maksimalni pad vrijednosti portfelja od 60%.

5.6.5. Simple Utility Function

Averzija prema riziku = 0.5



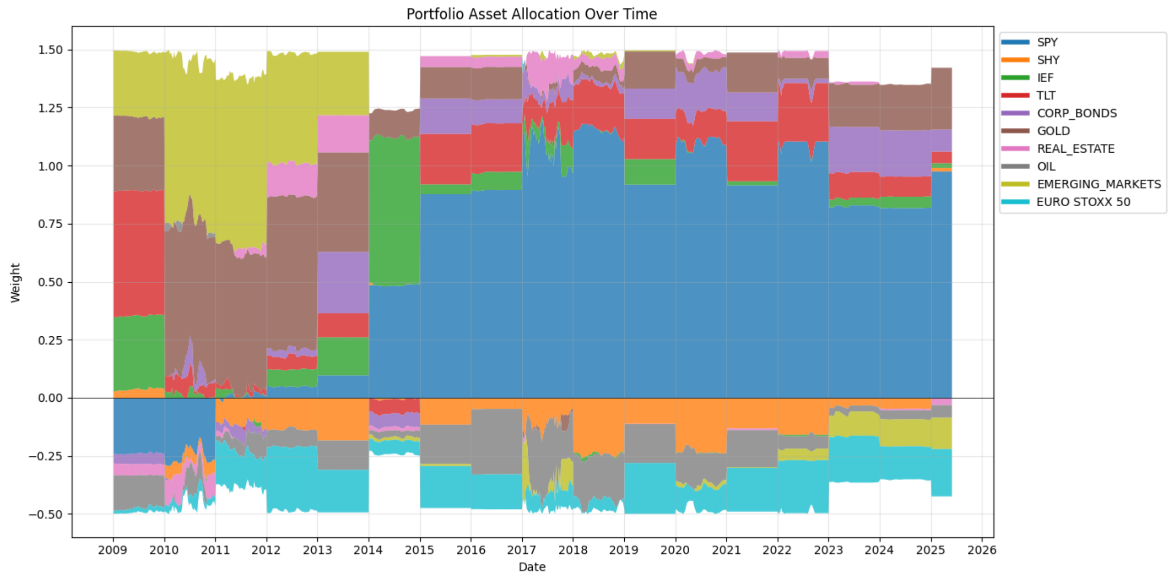
Slika 5.25. Simple Utility Function 0.5 Model_L.



Slika 5.26. Težine portfelja Model_L – Simple Utility Function 0.5.

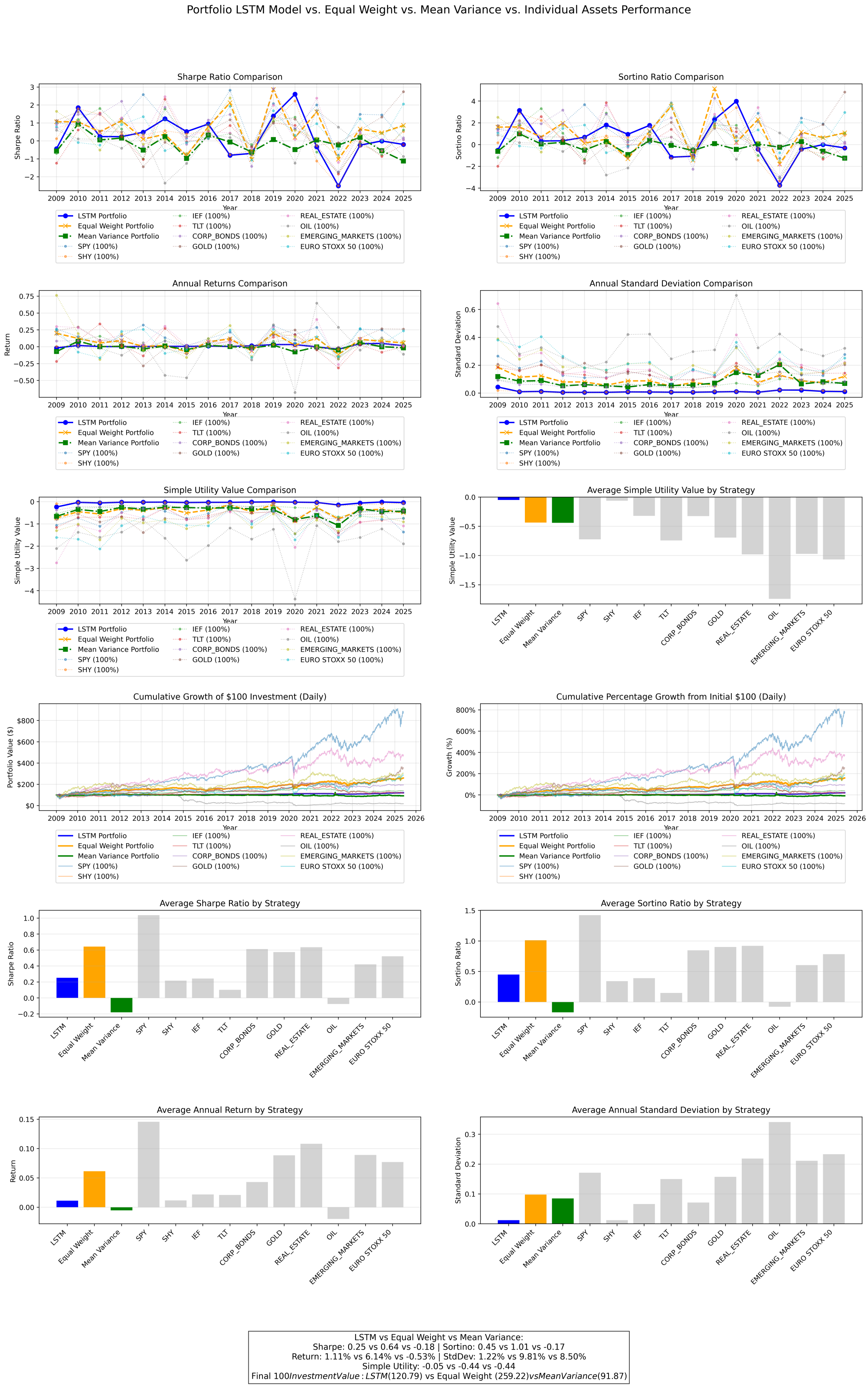
Averzija prema riziku = 1

Slika 5.27. Simple Utility Function 1 Model_L.

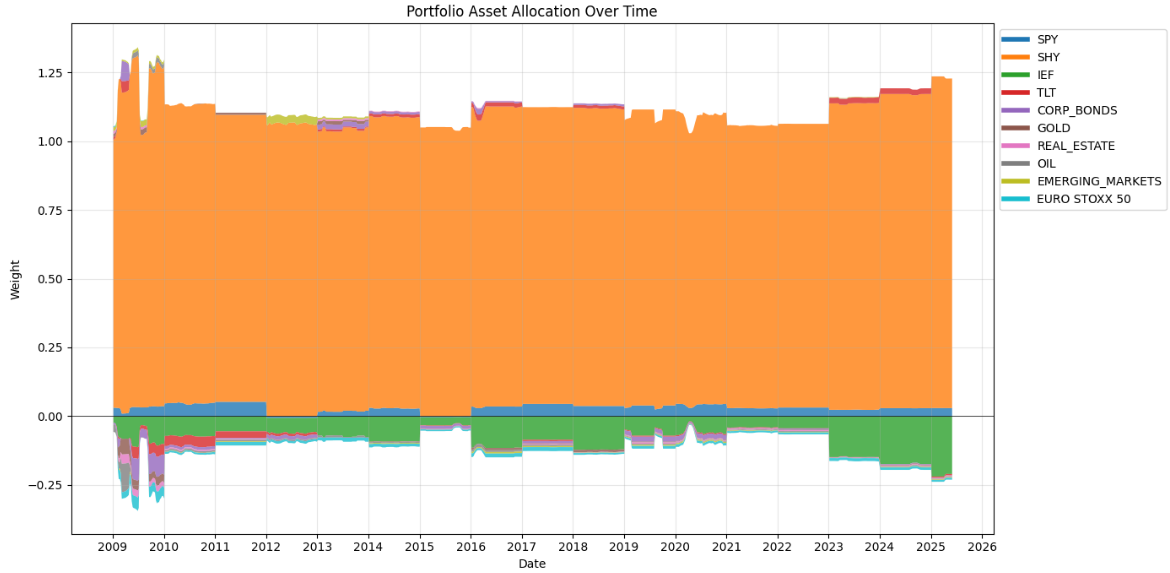


Slika 5.28. Težine portfelja Model_L – Simple Utility Function 1.

Averzija prema riziku = 5

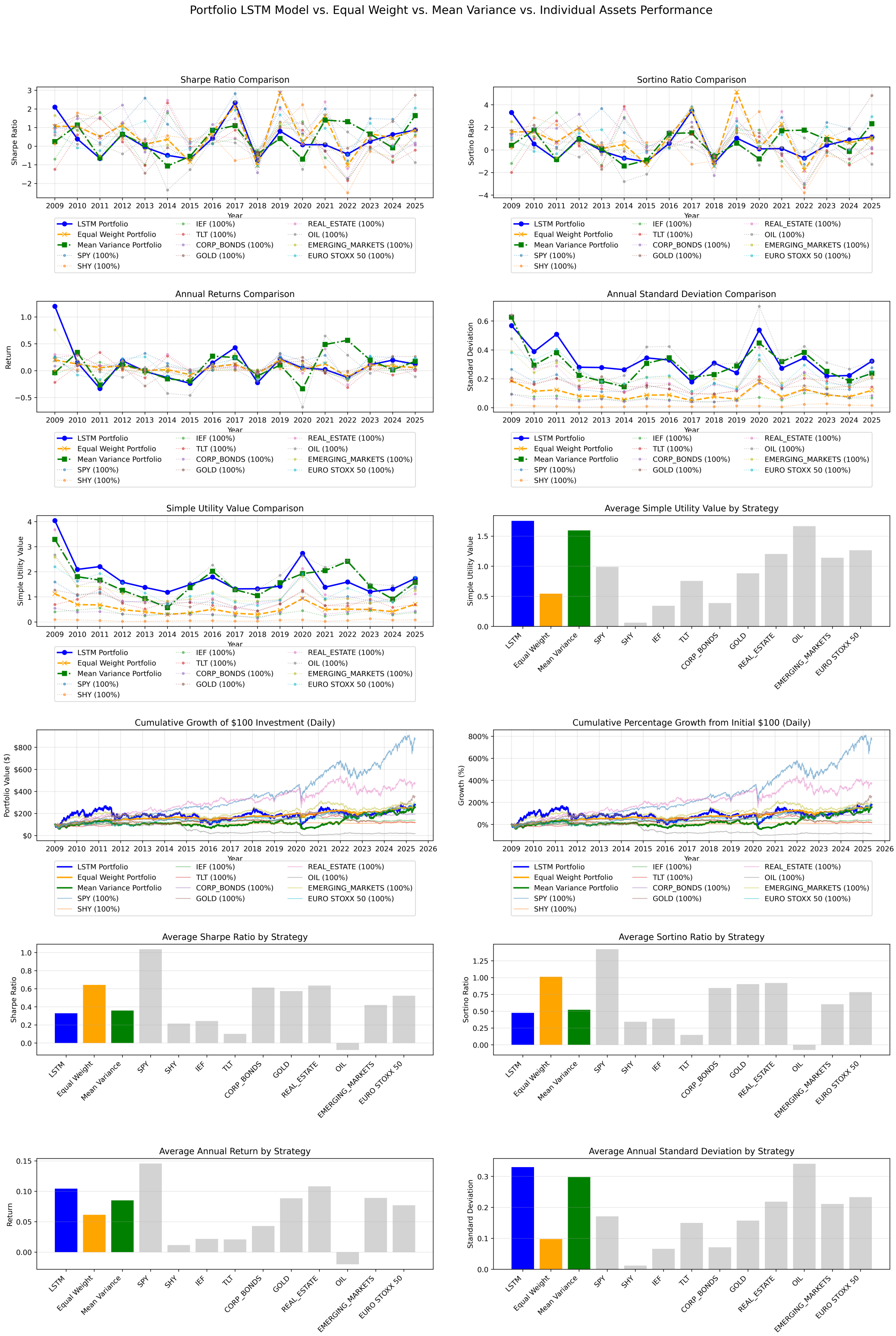


Slika 5.29. Simple Utility Function 5 Model_L.



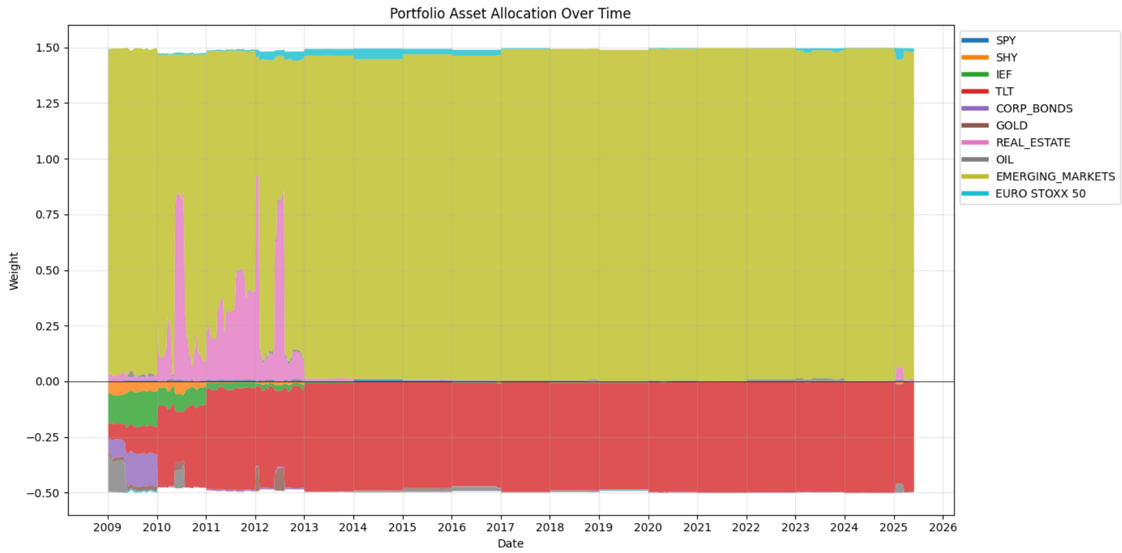
Slika 5.30. Težine portfelja Model_L – Simple Utility Function 5.

Averzija prema riziku = -5



LSTM vs Equal Weight vs Mean Variance:
Sharpe: 0.33 vs 0.64 vs 0.36 | Sortino: 0.48 vs 1.01 vs 0.52
Return: 10.43% vs 6.14% vs 8.51% | StdDev: 33.01% vs 9.81% vs 29.78%
Simple Utility: 1.75 vs 0.54 vs 1.60
Final 100*InvestmentValue* : *LSTM*(278.99) vs Equal Weight (259.22)*vsMeanVariance*(257.58)

Slika 5.31. Simple Utility Function -5 Model_L.



Slika 5.32. Težine portfelja Model_L – Simple Utility Function -5.

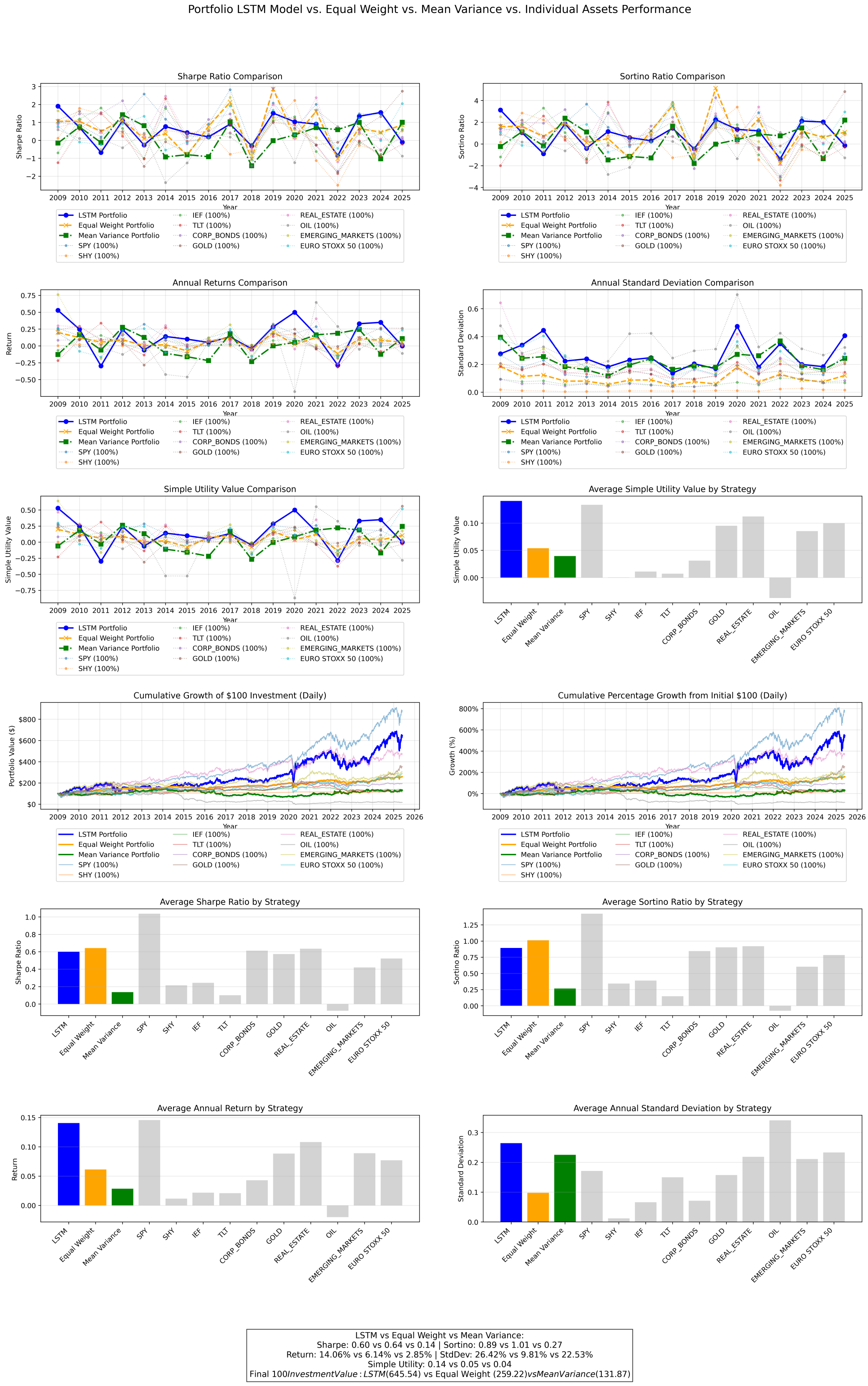
Rezultati pokazuju da maksimizacija jednostavne funkcije korisnosti (Simple Utility) daje konkurentne rezultate – prema samoj funkciji korisnosti, tijekom razdoblja od 15 godina, model je nadmašio sve ostale klase imovine i referentne portfelje.

Model treniran na ovaj način pokazuje sposobnost prilagodbe razini rizika koju investitor može tolerirati. Na primjer, za koeficijent averzije prema riziku jednak 0,5, ostvareni su prosječni godišnji povrat od 15%, standardna devijacija (rizik) od 23% te maksimalni pad vrijednosti portfelja (Maximum Drawdown) od 39%. Kako averzija prema riziku raste, smanjuju se i rizik i očekivani prinosi: pri averziji 1, godišnji povrat iznosi 9%, standardna devijacija 15%, a maksimalni pad vrijednosti 29%, dok za averziju 5 povrat pada na 1%, uz standardnu devijaciju od svega 1% i maksimalni pad od 5%.

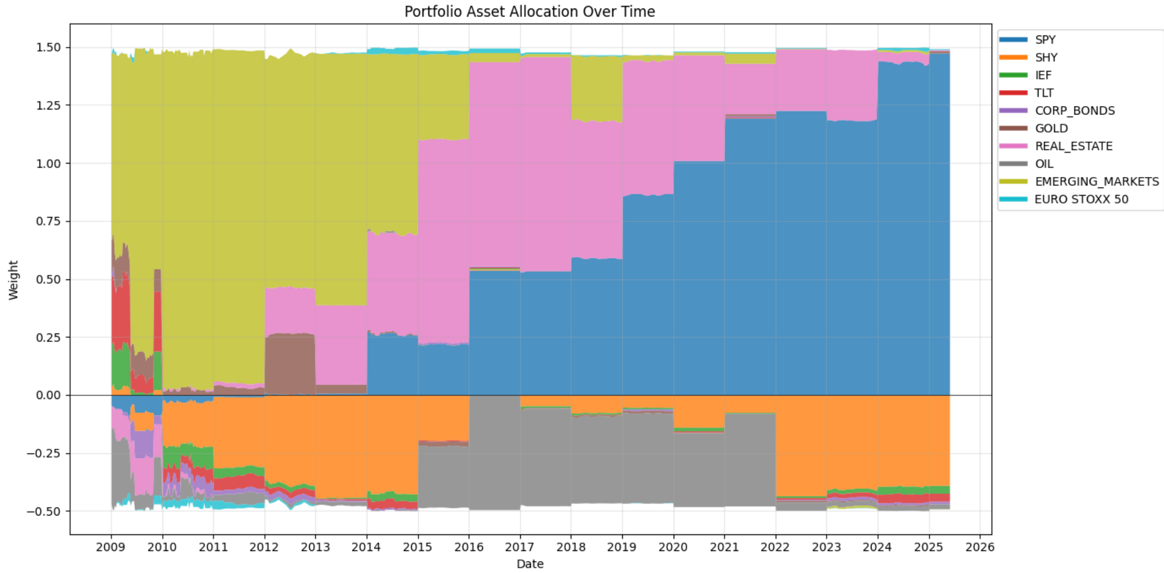
Dodatno, model je treniran i s negativnim koeficijentom averzije prema riziku jednakom -5, kako bi se istražila mogućnost maksimizacije rizika za investitore sklone preuzimanju visokog rizika. U tom slučaju, ostvareni su godišnji povrat od 10%, standardna devijacija od 33% i maksimalni pad vrijednosti portfelja od 64%.

Razlika između jednostavne funkcije korisnosti i klasične funkcije korisnosti leži u tome što je jednostavna funkcija korisnosti znatno osjetljivija na averziju prema riziku. Stoga su kod jednostavne funkcije korisnosti razlike u rezultatima između različitih vrijednosti averzije prema riziku daleko izraženije i značajnije nego kod klasične funkcije korisnosti.

5.6.6. Povrati



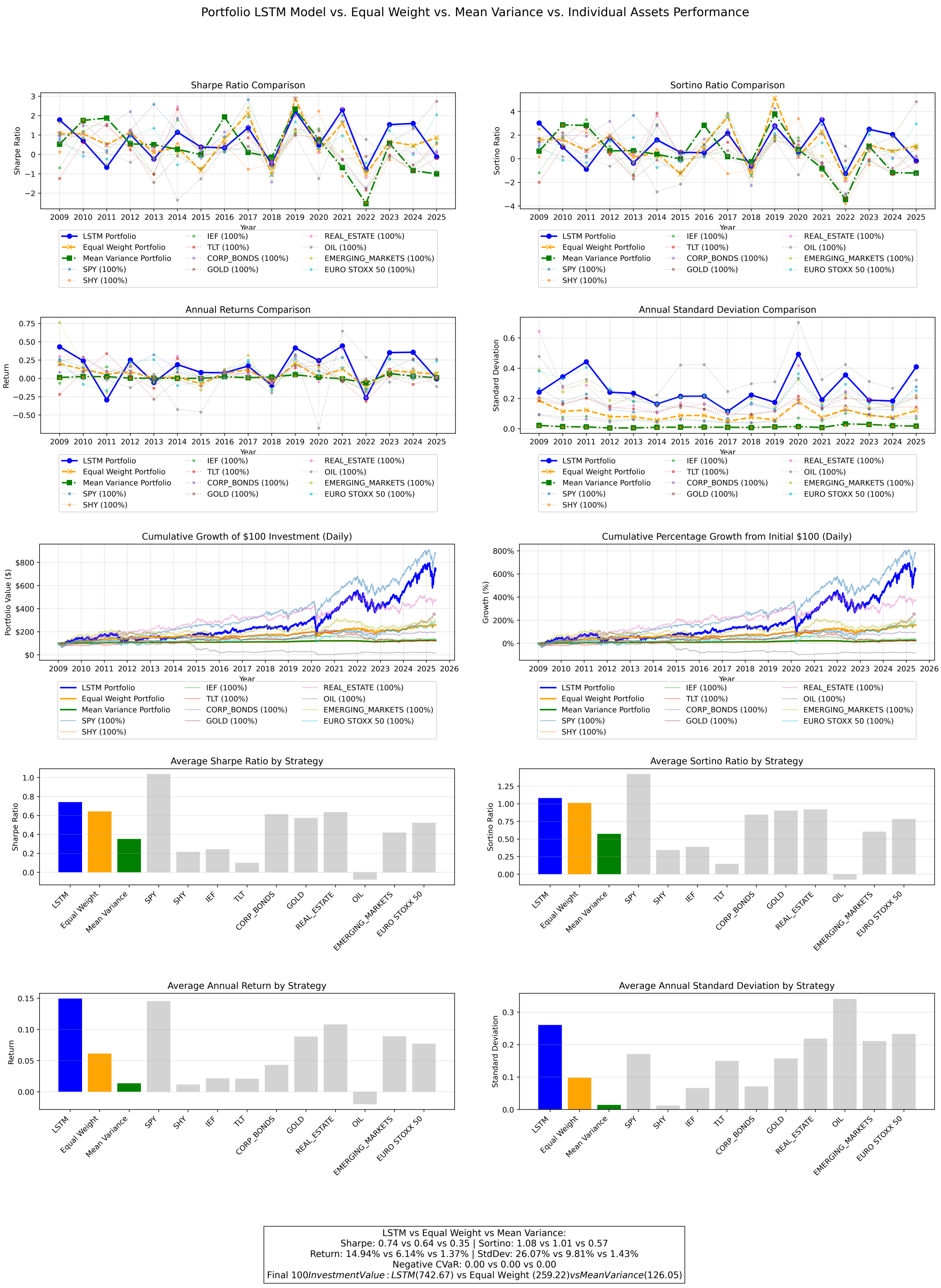
Slika 5.33. Povrati Model_L.



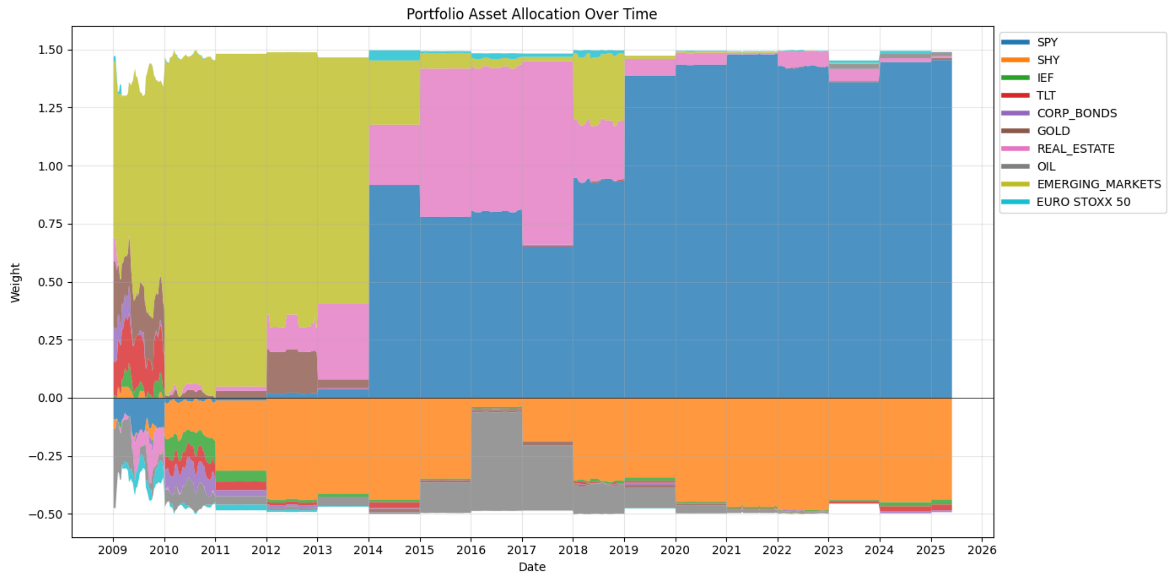
Slika 5.34. Težine portfelja Model_L – Povrati.

Treniranjem modela s ciljem maksimizacije povrata postignut je prosječni godišnji povrat od 14%, čime je tijekom razdoblja dužeg od 15 godina model nadmašio sve ostale klase imovine i referentne (benchmark) modele, osim SPY s kojim je izjednačen (Model_S blago bolji, Model_L blago lošiji povrati), prema tom kriteriju.

5.6.7. EWR



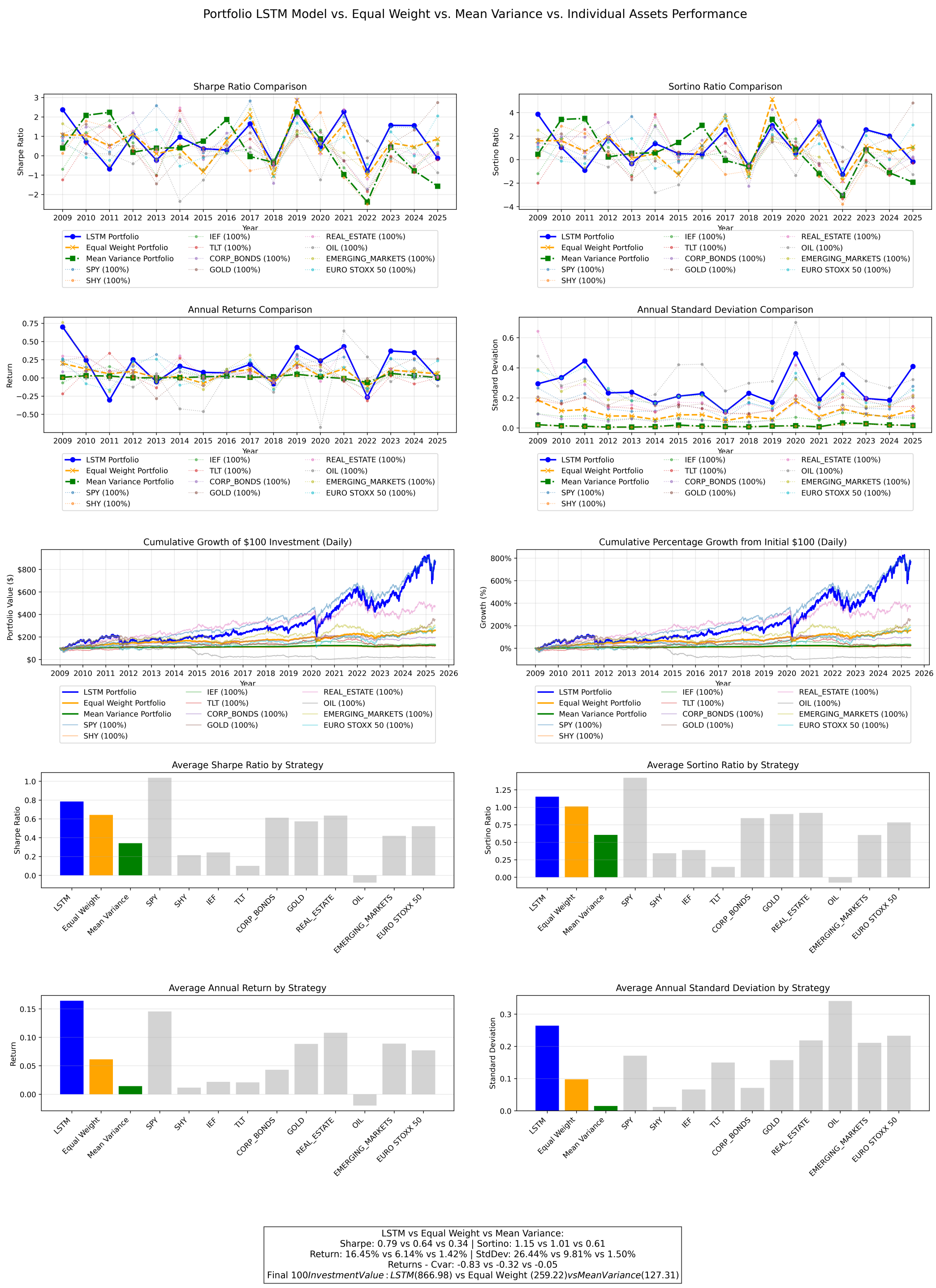
Slika 5.35. EWR Model_L.



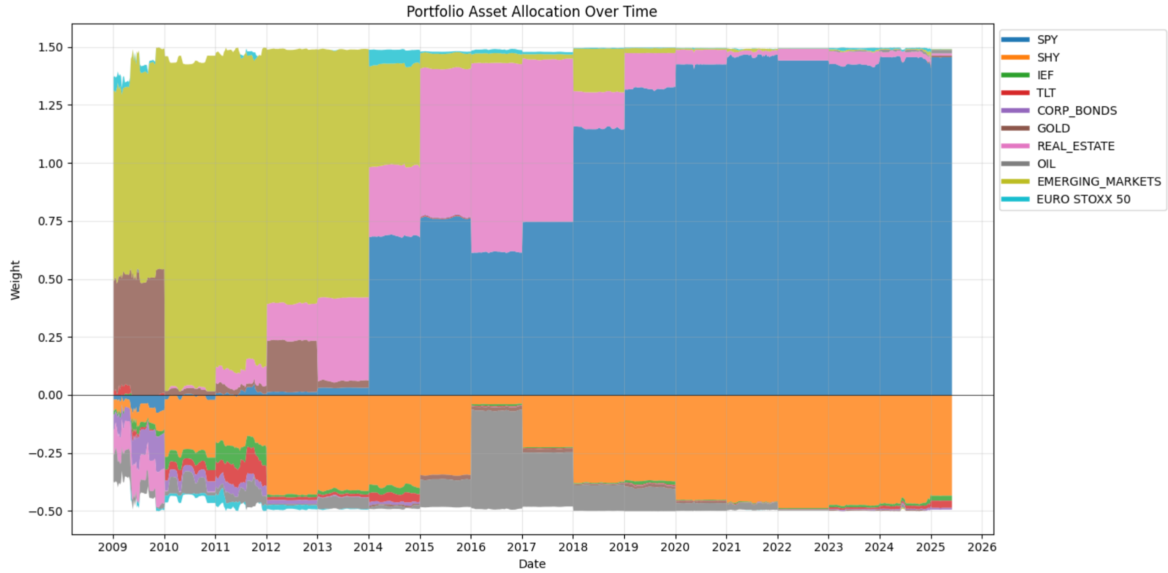
Slika 5.36. Težine portfelja Model_L – EWR.

Treniranjem modela s ciljem maksimizacije EWR-a postignut je prosječni godišnji povrat od 18%, čime je tijekom razdoblja dužeg od 15 godina model nadmašio sve ostale klase imovine i referentne (benchmark) modele prema tom kriteriju.

5.6.8. Povrati + EWR



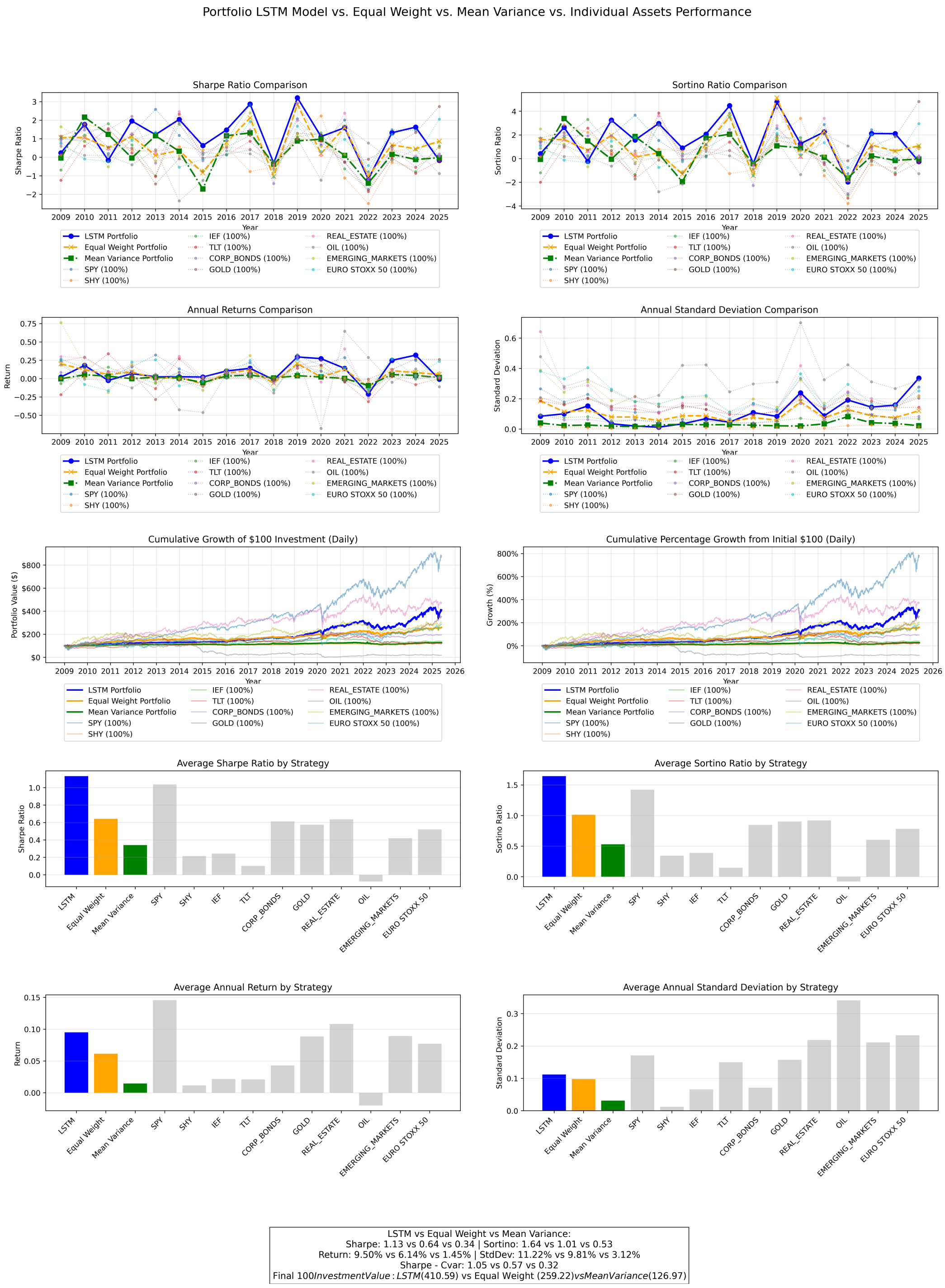
Slika 5.37. Povrati + EWR Model_L.



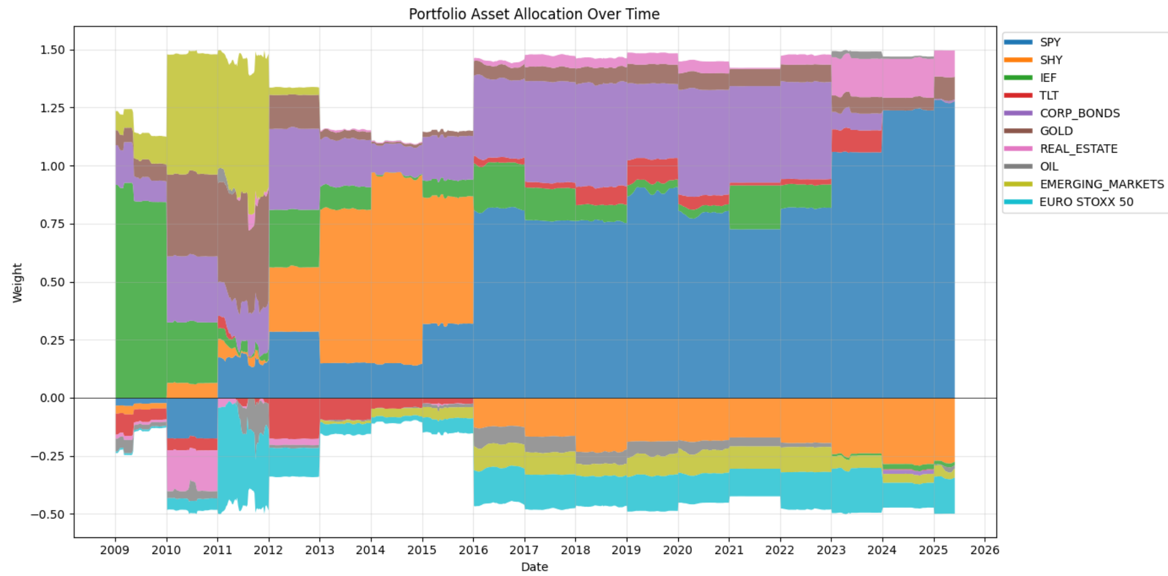
Slika 5.38. Težine portfelja Model_L – Povrati + EWR.

Treniranjem modela s ciljem maksimizacije razlike povrata i EWR-a postignut je prosječni godišnji povrat od 16%, čime je tijekom razdoblja dužeg od 15 godina model nadmašio sve ostale klase imovine i referentne (benchmark) modele prema tom kriteriju.

5.6.9. Sharpe + EWR



Slika 5.39. Sharpe + EWR Model_L.



Slika 5.40. Težine portfelja Model_L – Sharpe + EWR.

Treniranjem modela s ciljem maksimizacije razlike Sharpeovog omjera i EWR-a postignut je prosječni godišnji povrat od 10% i Sharpeov omjer 1.13, čime je tijekom razdoblja dužeg od 15 godina model nadmašio sve ostale klase imovine i referentne (benchmark) modele prema Sharpeovom omjeru. U odnosu na optimizaciju samo Sharpeovog omjera, ova metoda je zadržala Sharpeov omjer približno jednakim (1.13 u odnosu na 1.19) uz značajno poboljšanje povrata (10% u odnosu na 7%).

6. Prednosti i ograničenja pristupa

6.1. Prednosti

- Sposobnost učenja kompleksnih obrazaca u vremenskim serijama
- Dinamička prilagodba težina portfelja na temelju tržišnih uvjeta
- Mogućnost optimizacije različitih ciljnih funkcija (Sharpe, Sortino, itd.)
- Automatsko otkrivanje odnosa između imovina bez potrebe za ručnim definiranjem

6.2. Ograničenja

- Osjetljivost na izbor hiperparametara (veličina modela, stopa učenja, itd.)
- Rizik preučenosti, posebno s ograničenim skupovima podataka
- Računska složenost treniranja dubokih modela
- Interpretabilnost rezultata - neuronske mreže kao "crna kutija"
- Potreba za značajnom količinom povijesnih podataka

7. Zaključak

Ovaj rad pokazuje kako duboko učenje, konkretno primjena LSTM neuronskih mreža, može predstavljati učinkovit pristup u optimizaciji portfelja temeljenoj na povijesnim financijskim podacima. Implementirani model uspješno je optimizirao različite funkcije cilja te je, osim SPY-a u nekim slučajevima, nadmašio tradicionalne strategije upravljanja imovinom, uključujući i samostalne investicije u pojedine instrumente. Iako je SPY indeks zabilježio iznimne povrate u posljednjih 15 godina, predloženi model je u svim slučajevima uspio generirati konkurentne, pa čak i bolje rezultate na funkcijama cilja za koje je bio treniran, dok je ostale vrste imovine i benchmark portfelje potpuno nadmašio.

Dobiveni rezultati potvrđuju potencijal neuronskih mreža u učenju složenih odnosa iz sirovih podataka, bez potrebe za eksplicitnom procjenom očekivanih prinosa i kovarijanci. Time se otvaraju vrata daljnjoj primjeni dubokog učenja u području financija, s posebnim naglaskom na izgradnju fleksibilnih i prilagodljivih sustava za upravljanje portfeljem. U budućnosti, nastavak istraživanja u ovom smjeru mogao bi dovesti do razvoja još učinkovitijih i robusnijih modela koji bolje reflektiraju kompleksnost stvarnih financijskih tržišta.

Buduća istraživanja mogla bi se usmjeriti na:

- Integraciju dodatnih podataka (fundamentalni pokazatelji, sentimenti, makroekonomski indikatori)
- Eksperimentiranje s različitim arhitekturama dubokog učenja (transformeri, konvolucijske mreže …)
- Eksperimentiranje s drugim vrstama strojnog učenja (podržano učenje, nenadzirano učenje)

Literatura

[1] Zhang, Z., Zohren, S., & Roberts, S. (2020). *Deep Learning for Portfolio Optimization*. Oxford-Man Institute of Quantitative Finance, University of Oxford.

[2] Roncalli, T. (2020). *Handbook of Financial Risk Management: Simulation and Machine Learning*. Chapman and Hall/CRC.

[3] Efficient Frontier, `https://xplaind.com/971864/efficient-frontier`, 28.6.2025.

[4] StatQuest: Long, Short-Term Memory (LSTM) with PyTorch + Lightning, `https://lightning.ai/lightning-ai/studios/statquest-long-short-term-memory-lstm-with-pytorch-lightning?section=featured`, 28.6.2025.

[5] Slika: The Elements of Quantitative Investing, G. A. Paleologo, 2025, Wiley Finance

Sažetak

Model dubokog učenja za optimizaciju portfelja zasnovan na heterogenim podacima

David Supančić

Nova istraživanja u području optimizacije portfelja se fokusiraju na primjenu modela dubokog učenja na povijesnim povratima imovina kako bi se zaobišle greške u procjenama parametara očekivanja i kovarijance. Ovaj rad istražuje primjenu dubokog učenja, posebice LSTM (Long Short-Term Memory) neuronskih mreža, u optimizaciji investicijskih portfelja. Predstavljena je implementacija modela koji koristi povijesne podatke o povratima imovine za predviđanje optimalnih težina portfelja s ciljem maksimiziranja Sharpeovog omjera. Model je evaluiran kroz više godina te uspoređen s jednako ponderiranim i povijesno-optimiziranim portfeljem. Rezultati pokazuju da LSTM pristup može generirati vrijednost kroz dinamičku prilagodbu težina portfelja na temelju tržišnih uvjeta, nadmašujući tradicionalne strategije.

Ključne riječi: optimizacija portfelja; duboko učenje; Long Short-Term Memory; backtesting;