



# HEDGING WITH LINEAR REGRESSIONS AND NEURAL NETWORKS

1. چکیده

2. شبیه‌سازی داده‌ها (Data Simulation)

3. شبکه‌های عصبی (Neural Networks)

4. رگرسیون خطی (Linear Regressions)

5. جمع‌بندی

✓ در یک مدل یک دوره ای، اپراتور در ابتدا یک گزینه رو به ارزش  $C_0$  میفروشد و دلتا واحد از دارایی مبنا را خریداری میکند ارزش نهایی سبد با در نظر گرفتن قیمت نهایی گزینه  $(C_1)$  و قیمت نهایی دارایی مبنا  $(S_1)$ :

$$V_1^\delta = \delta S_1 + (1 + r_{\text{onr}} \Delta t)(C_0 - \delta S_0) - C_1,$$

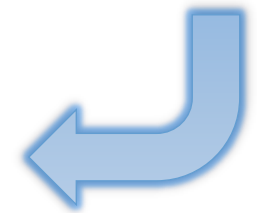
✓ هدف اصلی تعیین مقدار دلتاست ( $\delta$ ) نحوی که:

✓ واریانس خطای پوشش ریسک در معاملات روزانه گزینه ها کمینه شود

✓ واریانس پرتفوی پوشش داده شده با میانگین مربعات خطای پوششی (HMSE) تقریب زده می شود.

$$\mathbf{E} \left[ \left( V_1^\delta \right)^2 \right] = \mathbf{E} \left[ \left( \delta S_1 + (1 + r_{\text{onr}} \Delta t)(C_0 - \delta S_0) - C_1 \right)^2 \right].$$

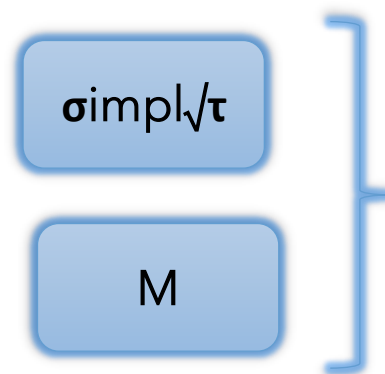
بدین ترتیب با کاهش واریانس، ریسک کمینه می شود و Hedging اتفاق می افتد.



✓ یکی از روش‌های سنتی برای تعیین مقدار دلتا: روش دلتای بلک شولز (Black-Scholes Delta)

$$\delta_{BS} = N(d_1),$$

$$d_1 = \frac{1}{\sigma_{impl}\sqrt{\tau}} \left[ \ln \left( \frac{S_0}{K} \right) + \left( r + \frac{1}{2}\sigma_{impl}^2 \right) \tau \right].$$

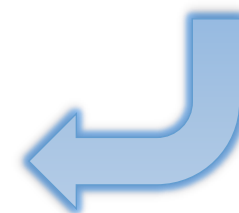


✓ دلتای بلک شولز تابعی از دو پارامتر

$$\delta_{BS} = f_{BS} (M, \sigma_{impl}\sqrt{\tau}) .$$

✓ در نتیجه:

و ایده‌ی اصلی این مقاله از این پرسش به ذهن می‌رسد: که چرا تابع دیگری به جای fBS در نظر نگیریم؟



ترکیب خطی از حساسیت‌های گزینه

Delta  
Vega  
Gamma  
Vanna

Linear Regressions  
(parametric)

سه حالت مختلف برای feature های ورودی

$M, \sigma \sqrt{\tau}$

Delta, Vega,  $\tau$

Delta, Vega, Vanna,  $\tau$

Neural Network  
(Non-parametric)

✓ دو روش معرفی شده در مقاله

یکی از چالش های اصلی این پروژه: عدم دسترسی به داده های مورد استفاده در مقاله  
✓ راهکار: شبیه سازی داده ها پیش از پیاده سازی رگرسیون خطی و مدل شبکه های عصبی

1. چکیده

2. شبیه‌سازی داده‌ها (Data Simulation)

3. شبکه‌های عصبی (Neural Networks)

4. رگرسیون خطی (Linear Regressions)

5. جمع‌بندی



بافرمول بلک-شولز  
 قیمت OPTION ها را  
 شبیه سازی میکنیم

$$d_1 = \frac{\ln\left(\frac{S(0)}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

$N()$  تابع توزیع تجمعی یک  
 توزیع نرمال استاندارد

$$d_2 = d_1 - \sigma\sqrt{T} = \frac{\ln\left(\frac{S(0)}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

$N()$  تابع توزیع تجمعی یک  
 توزیع نرمال استاندارد

$$\text{CALL OPTION PRICE} = S_T * N(D1) - K * \text{EXP}(-R * T) * N(D2)$$

شبیه سازی مسیر دارایی پایه با استفاده از  
 GBM (حرکت براونی هندسی)

بر اساس قوانین بورس گزینه‌های هیئت  
 مدیره شیکاگو (CBOE) بر اساس  
 قیمت سهام برای هر سررسید

-  $R$  نرخ بهره بدون ریسک  
 -  $T$  زمان انقضا

$$S_{t+\Delta t} = S_t \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}\right]$$

$$S_t \sim \hat{\text{GBM}}(\mu, \sigma)$$

$$d_2 = d_1 - \sigma\sqrt{T} = \frac{\ln\left(\frac{S(0)}{K}\right) + (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$$

$$d_1 = \frac{\ln\left(\frac{S(0)}{K}\right) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$$

بافرمول بلک-شولز  
قیمت OPTION ها را  
شبیه سازی میکنیم

N () تابع توزیع تجمعی یک  
توزیع نرمال استاندارد

N () تابع توزیع تجمعی یک  
توزیع نرمال استاندارد

$$\text{PUT OPTION PRICE} = K * \text{EXP}(-R * T) * N(-D2) - ST * N(-D1)$$

بر اساس قوانین بورس گزینه‌های هیئت  
مدیره شیکاگو (CBOE) بر اساس  
قیمت سهام برای هر سررسید

- R نرخ بهره بدون ریسک  
- T زمان انقضا

شبیه سازی مسیر دارایی پایه با استفاده از  
GBM (حرکت براونی هندسی)

$$S_t \sim \text{GBM}(\mu, \sigma)$$

$$S_{t+\Delta t} = S_t \exp \left[ \left( \mu - \frac{\sigma^2}{2} \right) \Delta t + \sigma \varepsilon \sqrt{\Delta t} \right]$$

• توابعی را برای شبیه سازی حساسیت های آپشن با فرمول های زیر ساختیم:

$$\Delta_{Call} = N(d_1) \quad \text{دلتا} \quad N'(d_1) = \frac{e^{\frac{-d_1^2}{2}}}{\sqrt{2\pi}} \quad \theta_{call} = -\frac{S_0 N'(d_1) \sigma}{2\sqrt{T}} - r \cdot K e^{-rT} N(d_2) \quad \text{تتا}$$

$$\Gamma = \frac{N'(d_1)}{S \sigma \sqrt{T}} \quad \text{گاما} \quad v = S_0 \sqrt{T} N'(d_1) \quad \text{وگا}$$

$$vanna = \sqrt{T-t} N'(d_1) \left( \frac{d_2}{\sigma} \right) \quad \text{وانا}$$

در انتها با نرمال کردن تمام مقادیر محاسبه شده در ستون ها برای OPTION های مختلف و محاسبه ی دوباره مقادیر شبیه سازی شده ی نرمال شده با فرمول هایی که در توابع حین شبیه سازی تعریف کردیم ، داده های تمیز شده تولید و ذخیره می کنیم . در این بخش کد ابتدا داده های درون نمونه را با پاک کردن آن و ترکیب داده های گزینه CALL و PUT پردازش می کند. سپس، از روش مونت کارلو برای شبیه سازی مجموعه داده های آزمایشی خارج از نمونه بر اساس سناریوهای آتی قیمت سهام استفاده می کنیم. این مجموعه داده های آزمایشی برای ارزیابی عملکرد مدل بر روی داده های دیده نشده و تحلیل رفتار آن در شرایط مختلف بازار شبیه سازی شده استفاده می شوند.

1. چکیده

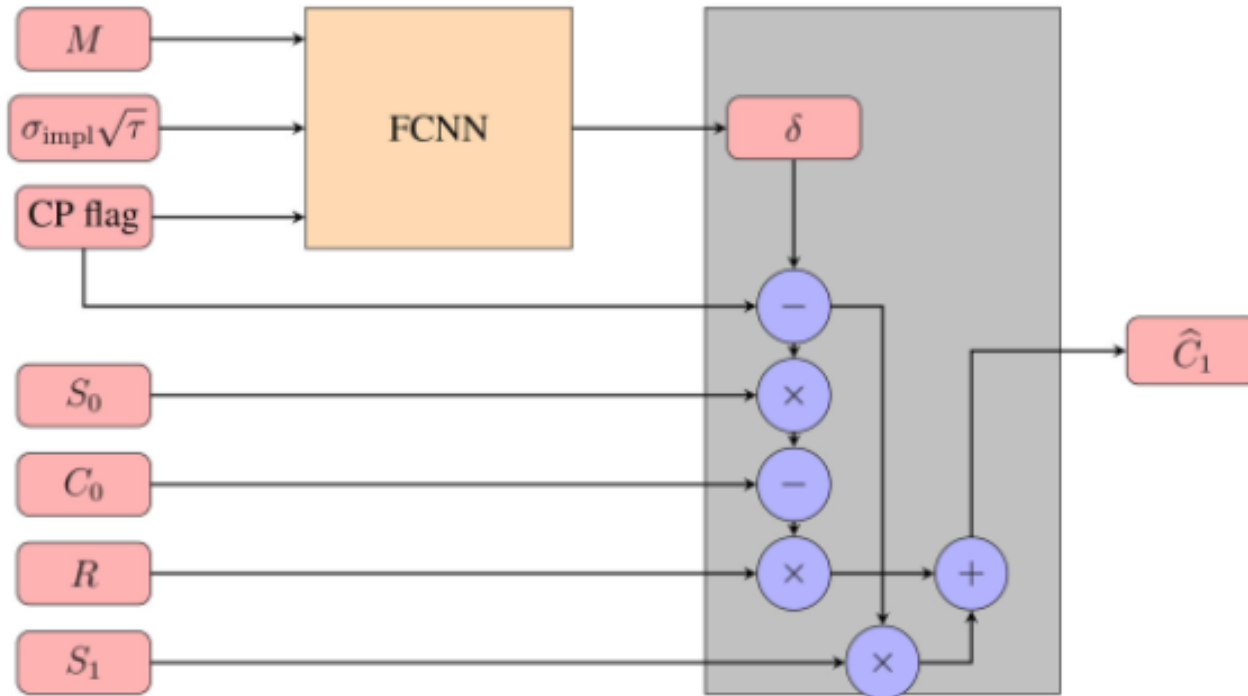
2. شبیه‌سازی داده‌ها (Data Simulation)

3. شبکه‌های عصبی (Neural Networks)

4. رگرسیون خطی (Linear Regressions)

5. جمع‌بندی

## ساختار شبکه عصبی (HedgeNet)

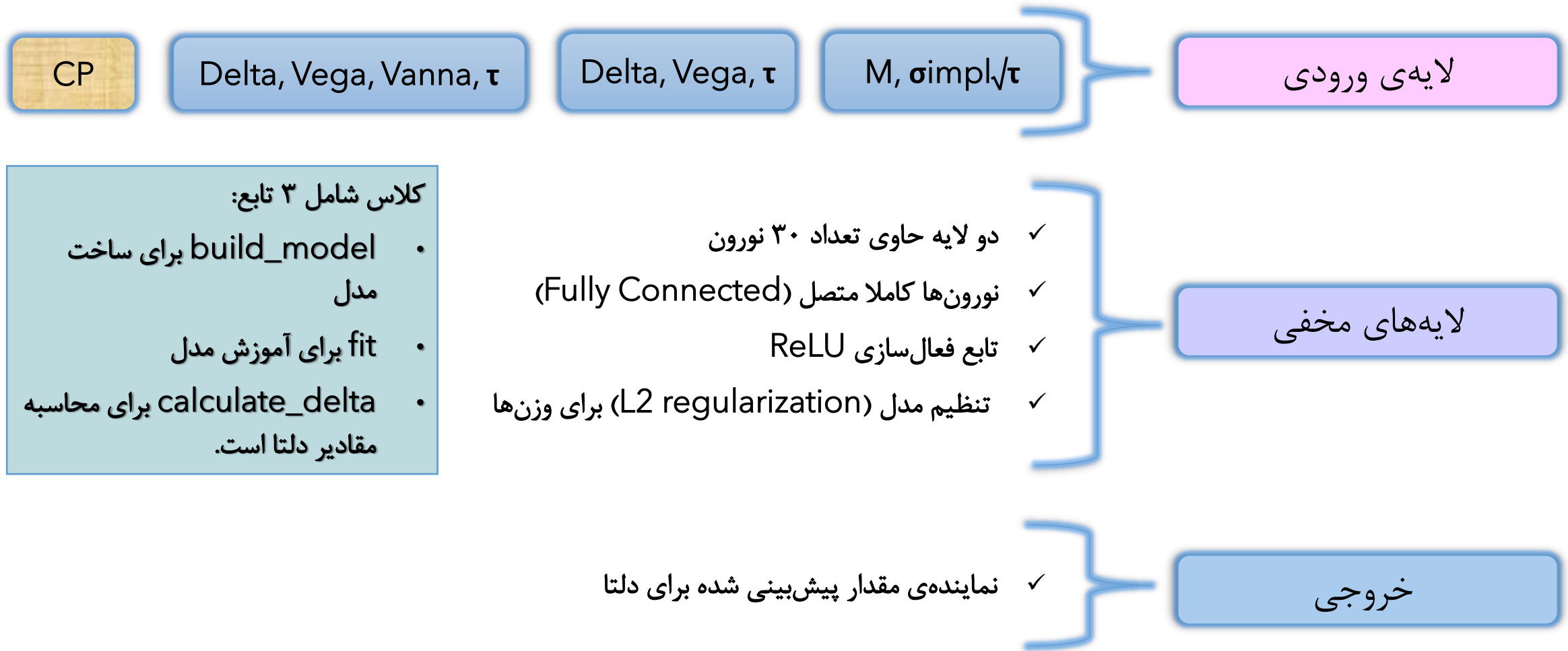


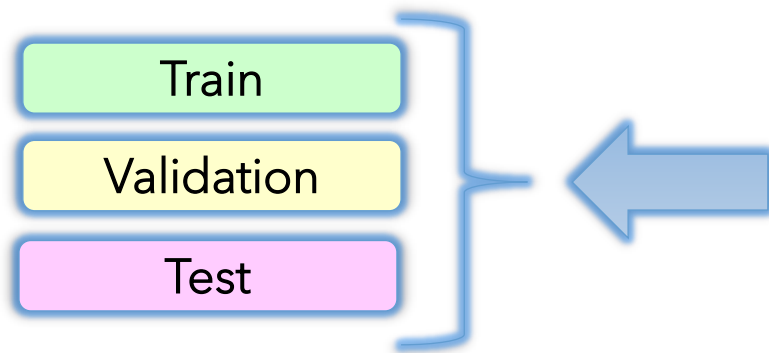
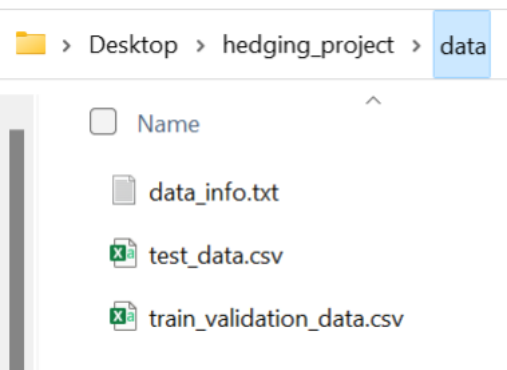
$$\delta = \text{Fcnn}(M, \sigma_{impl}\sqrt{\tau})$$

$$C1^{\wedge} = \delta S1 + (1 + \text{ronrt})(C0 - \delta S0), V\delta1 = \delta S1 + (1 + \text{ronrt})(C0 - \delta S0) - C1$$

✓ استفاده از کتابخانه Keras در Python

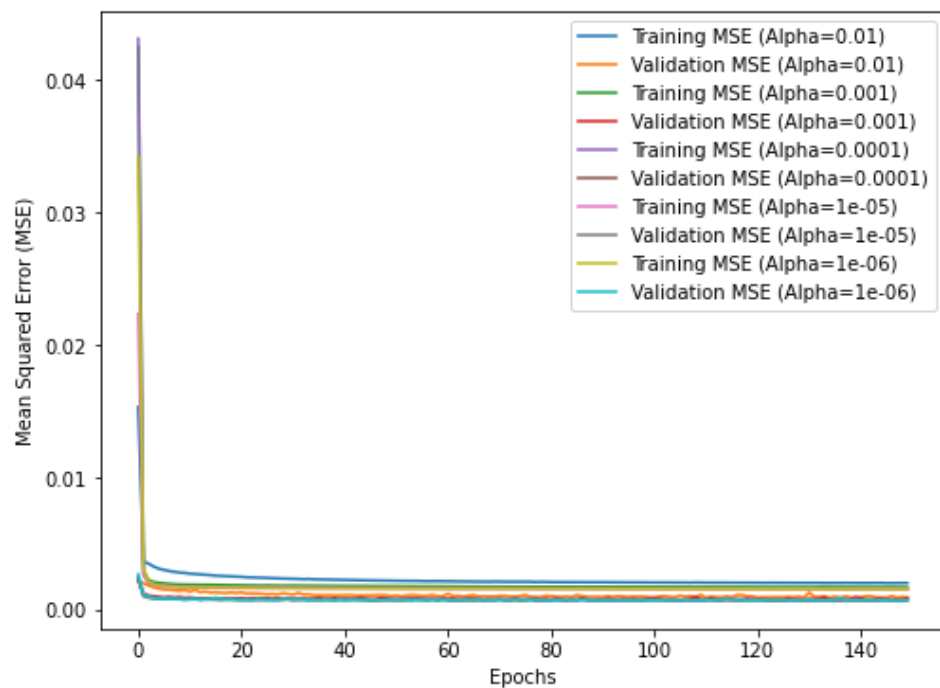
✓ ساختار HedgeNet:





گام ۱. آماده‌سازی داده‌ها (حذف داده‌های پرت، استانداردسازی ویژگی‌ها جداسازی داده‌ها)

گام ۲. Hyperparameter Tuning



```
best_hyperparameter_value
```

```
['reg_alpha': 1e-06]
```

گام ۳. آموزش با استفاده از بهترین مدل

Best Model :

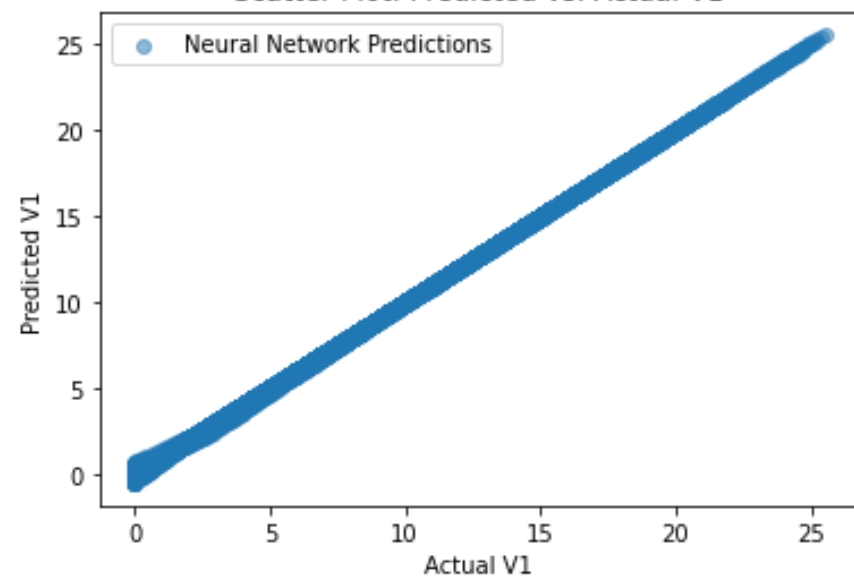
```
'nodes_per_layer': (30, 30),  
'alpha': 1e-6,  
'lr': 1e-4,  
'epochs': 150,  
'activation_function': 'relu'  
'optimizer' : 'adam'  
'loss_function' : 'HMSE'
```

گام ۴. ارزیابی روی داده‌های تست

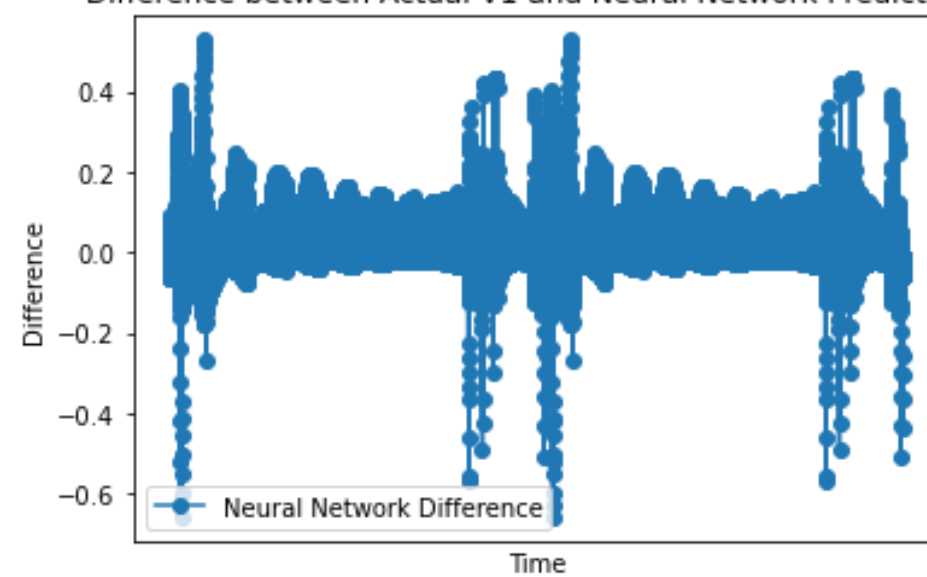
```
▶ # Print the overal HMSE  
print(np.mean(hmse_nn))  
  
0.0015342187035403355
```



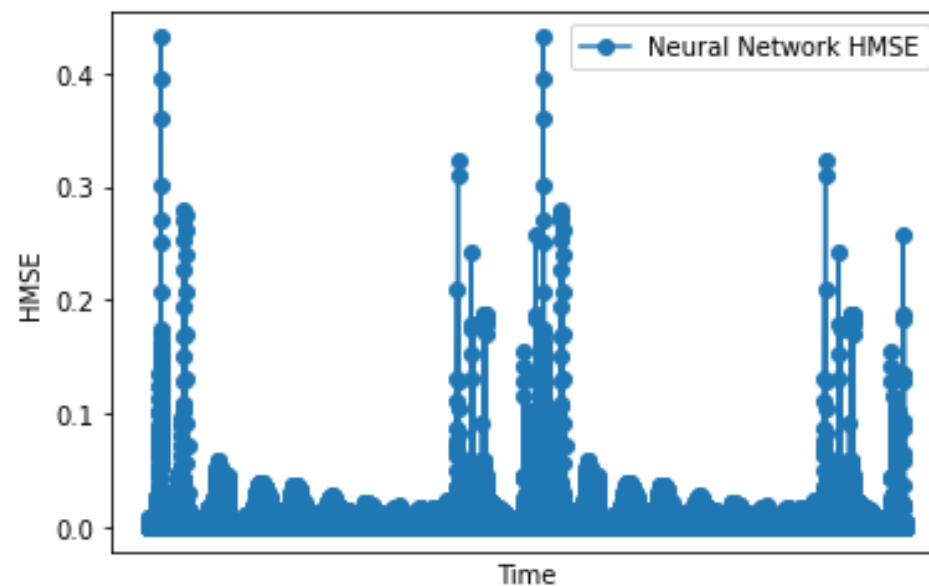
Scatter Plot: Predicted vs. Actual V1



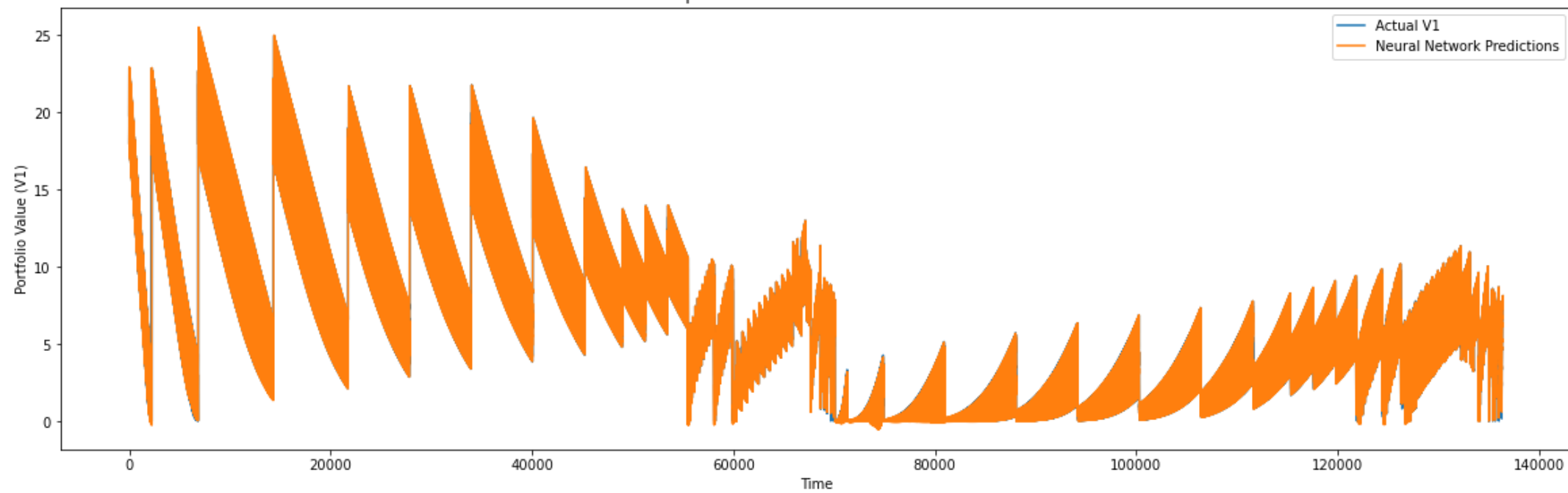
Difference between Actual V1 and Neural Network Predictions



HMSE between Actual V1 and Predictions



Comparison of Portfolio Value Predictions



1. چکیده

2. شبیه‌سازی داده‌ها (Data Simulation)

3. شبکه‌های عصبی (Neural Networks)

4. رگرسیون خطی (Linear Regressions)

5. جمع‌بندی

گام ۱. پاکسازی و آماده سازی داده ها مثل بخش قبل

1. No-Hedge

2. ۱۳ پیکربندی با استفاده از ترکیب:

Delta

Vega

Gamma

Vanna

گام ۲. استفاده از پیکربندی های مختلف برای رگرسیون خطی

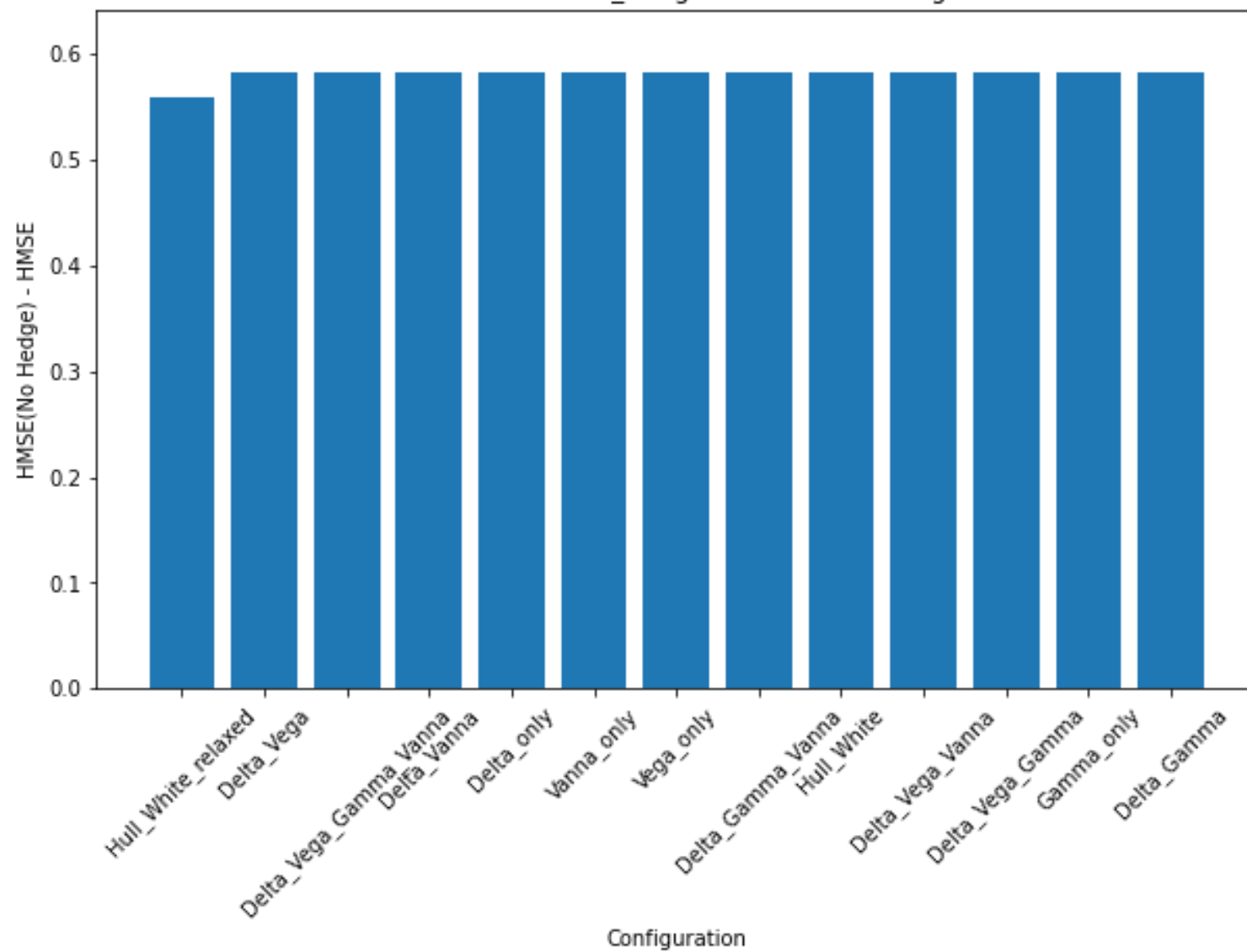
e.g. Delta-Vega ( $\delta = a * \delta_{BS} + b * V_{BS}$ )

گام ۳. اجرای رگرسیون خطی روی داده های آموزش و ذخیره نتایج (ضرایب، مقادیر دلتا)

## گام ۴. محاسبه‌ی HMSE روی داده‌های تست

```
No_Hedge HMSE: 0.5841938456661125
BS HMSE: 0.0012855653441444655
Delta_only HMSE: 0.0013307338673194662
Delta_Vega HMSE: 0.0015386190552253362
Delta_Gamma HMSE: 0.0011941685820317206
Delta_Vanna HMSE: 0.001332793949772294
Delta_Gamma_Vanna HMSE: 0.0012500598246228765
Delta_Vega_Gamma HMSE: 0.0012065560389979333
Delta_Vega_Vanna HMSE: 0.0012410582133362386
Hull_White_relaxed HMSE: 0.024310208830454722
Hull_White HMSE: 0.0012479273029527339
Gamma_only HMSE: 0.0011941844000625062
Vega_only HMSE: 0.0012866928369154305
Vanna_only HMSE: 0.0012936925576464604
Delta_Vega_Gamma_Vanna HMSE: 0.0014297587646802143
```

HMSE difference with No\_Hedge for different configurations



1. چکیده

2. شبیه‌سازی داده‌ها (Data Simulation)

3. شبکه‌های عصبی (Neural Networks)

4. رگرسیون خطی (Linear Regressions)

5. جمع‌بندی

## مقایسه میزان کاهش ریسک در Linear Regression و Neural Networks

```
▶ # Print the overall HMSE  
print(np.mean(hmse_nn))
```

```
0.0015342187035403355
```

```
Delta_only HMSE: 0.0013307338673194662  
Delta_Vega HMSE: 0.0015386190552253362  
Delta_Gamma HMSE: 0.0011941685820317206  
Delta_Vanna HMSE: 0.001332793949772294  
Delta_Gamma_Vanna HMSE: 0.0012500598246228765  
Delta_Vega_Gamma HMSE: 0.0012065560389979333  
Delta_Vega_Vanna HMSE: 0.0012410582133362386  
Hull_White_relaxed HMSE: 0.024310208830454722  
Hull_White HMSE: 0.0012479273029527339  
Gamma_only HMSE: 0.0011941844000625062  
Vega_only HMSE: 0.0012866928369154305  
Vanna_only HMSE: 0.0012936925576464604  
Delta_Vega_Gamma_Vanna HMSE: 0.0014297587646802143
```



از توجه شما سپاسگزارم