

به نام خدا

گزارش پیشرفت پروژه پیش‌بینی با استفاده از شبکه عصبی

چکیده گزارش

در این پروژه، هدف اصلی توسعه مدل‌هایی برای پیش‌بینی قیمت جفت‌ارز EUR/USD با استفاده از شبکه‌های عصبی بازگشتی (LSTM) و ترنسفورمرها بود. در طی مراحل مختلف، چندین مدل با رویکردها و ویژگی‌های متفاوت طراحی و پیاده‌سازی شدند تا چالش‌های اصلی مانند تاخیر در پیش‌بینی (Lagging) و سادگی بیش از حد مدل برطرف شوند. در مراحل اولیه، سادگی ورودی‌ها و محدود بودن ویژگی‌ها باعث ضعف مدل در یادگیری رفتار واقعی بازار شد. با گسترش ورودی‌ها و تغییر معماری‌ها از LSTM به ترنسفورمر، کیفیت نتایج به‌صورت ملموس افزایش یافت. با این حال، مسئله تاخیر در پیش‌بینی به‌عنوان چالشی مشترک در تمامی مدل‌ها باقی ماند. شبکه‌ها در بسیاری از موارد به‌جای یادگیری واقعی از داده‌ها، مقادیر گذشته را تقلید می‌کردند.

پیشنهادهای بهبود

۱. تغییر خروجی مدل (از Close به تغییرات قیمت)

یکی از مشکلات اصلی مشاهده‌شده در تمامی مدل‌های توسعه‌یافته تا این مرحله، تاخیر (lag) در پیش‌بینی‌هاست. تحلیل رفتار مدل‌ها نشان داد که بسیاری از آن‌ها به جای درک روند و پیش‌بینی واقعی آینده، صرفاً به تکرار مقادیر قبلی تمایل دارند. یکی از راهکارهای موثر برای مقابله با این رفتار، تغییر فرم خروجی مدل، از مقدار قیمت بسته‌شدن (Close) به تغییر نسبی آن است:

$$\text{Target} = \text{Close}(t+1) - \text{Close}(t)$$

این تغییر ساده، مدل را مجبور می‌کند تا بر روی حرکت و نوسانات بازار تمرکز کند، نه مقدار خام قیمت. در واقع، به‌جای پیش‌بینی مقدار خود قیمت که اغلب پایدار و نزدیک به مقدار قبلی است، مدل باید یاد بگیرد که چه مقدار افزایش یا کاهش در قیمت رخ خواهد داد.

۲. بررسی استفاده از الگوریتم‌های یادگیری ماشین سنتی

اگرچه مدل‌های مبتنی بر LSTM و ترنسفورمر به‌صورت ذاتی برای سری‌های زمانی طراحی شده‌اند، اما در بسیاری از موارد، مدل‌های یادگیری ماشین سنتی مانند:

- XGBoost

- LightGBM

- Random Forest

- Support Vector Regression (SVR)

می‌توانند نتایجی رقابتی یا حتی بهتر ارائه دهند، به‌ویژه در شرایطی که داده‌ها مهندسی ویژگی (Feature Engineering) خوبی داشته باشند.

مزایای بالقوه این مدل‌ها:

- مدل‌های سنتی کمتر دچار تکرار مقادیر قبلی یا lag می‌شوند و وابستگی کمتری به توالی زمانی دارند.
 - سرعت آموزش و تست این مدل‌ها بسیار بالا است و برای تست‌های سریع و پیاده‌سازی‌های سبک‌تر مناسب‌اند.
 - قابلیت درک بهتر ویژگی‌های پیچیده و غیرخطی زمانی که مهندسی ویژگی مناسبی صورت گرفته باشد.
 - امکان تفسیر ویژگی‌ها (feature importance) که می‌تواند به تحلیل داده و شناخت مؤثرترین ورودی‌ها کمک کند.
- یک رویکرد مفید می‌تواند ترکیب خروجی مدل‌های زمانی با خروجی مدل‌های یادگیری ماشین باشد.

۳. پیشنهادات تکمیلی

- **پیش‌بینی طبقه‌بندی‌شده (classification-based):** به جای پیش‌بینی مقدار دقیق، جهت حرکت قیمت را پیش‌بینی کنیم (مثلاً: بالا / پایین / بدون تغییر). این کار به‌ویژه برای تصمیم‌گیری‌های معاملاتی ساده، بسیار مؤثر است.

نحوه جداسازی داده‌ها

پیش از معرفی و بررسی مدل‌ها، لازم است به نحوه‌ی تقسیم‌بندی داده‌ها اشاره شود. به منظور ارزیابی عملکرد واقعی مدل‌ها در شرایطی که مشابه داده‌های آینده باشند، کل داده‌ها به سه بخش آموزش (Training)، ارزیابی (Validation) و آزمون (Testing) تقسیم شده‌اند.

در این پروژه، ۸۰٪ از داده‌ها برای آموزش مدل، ۱۰٪ از داده‌ها برای ارزیابی و ۱۰٪ باقی‌مانده برای تست مدل در نظر گرفته شده است. کلیه‌ی نمودارهایی که در ادامه برای بررسی عملکرد مدل‌ها ارائه می‌شوند، مربوط به داده‌های تست (Test Set) هستند و مدل در طول آموزش این بخش از داده را هرگز ندیده است. به این ترتیب، تحلیل‌های انجام‌شده منعکس‌کننده‌ی توان واقعی مدل در تعمیم به داده‌های جدید می‌باشد.

مدل اول

اولین مدل پیاده‌سازی‌شده یک شبکه عصبی بازگشتی مبتنی بر LSTM (Long Short-Term Memory) بود که برای پیش‌بینی متغیرهای OHLC (قیمت باز، بالا، پایین و بسته) جفت‌ارز EUR/USD طراحی شد. این مدل با هدف استفاده از الگوهای زمانی موجود در داده‌های تاریخی بازار فارکس توسعه یافت. مدل از یک معماری متوالی (Sequential) تشکیل شده است که ترکیبی از

سه لایه شبکه LSTM دارد. این لایه‌ها به مدل کمک می‌کنند تا الگوهای پیچیده و تغییرات قیمت را در طول زمان بهتر درک کند. استفاده از سه لایه به جای یک لایه باعث می‌شود مدل ما قوی‌تر عمل کند و بتواند هم الگوهای ساده (مثل بالا و پایین رفتن قیمت) و هم الگوهای پیچیده‌تر (مثل تغییرات ناگهانی بازار) را یاد بگیرد.

داده‌های ورودی

داده‌های ورودی شامل 60 کندل متوالی در تایم‌فریم 15 دقیقه‌ای بود که شامل قیمت بسته‌شدن و حجم معاملات می‌باشند.

هدف مدل

هدف این مدل، پیش‌بینی همزمان چهار متغیر OHLC برای کندل بعدی (15 دقیقه بعد) بر اساس داده‌های 60 کندل قبلی بود.

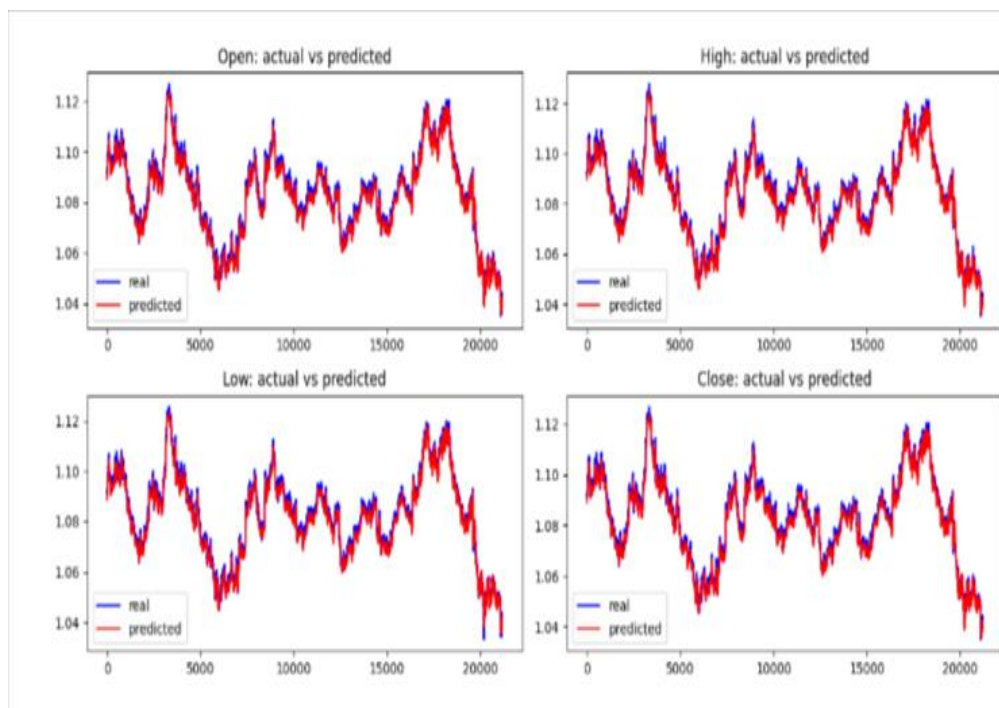
نتایج مدل

در نهایت، نتایج مدل برای پیش‌بینی متغیرهای OHLC به شرح زیر شد:

	MSE	RMSE	MAE	R2
Open	3×10^{-6}	0.0019	0.0017	0.98
High	3×10^{-6}	0.0019	0.0016	0.98
Low	4×10^{-6}	0.0020	0.0018	0.98
Close	4×10^{-6}	0.0020	0.0017	0.98

تحلیل و چالش‌های مدل

با وجود نتایج اولیه قابل قبول و حتی عالی از نظر معیارهای عددی مانند MSE، بررسی دقیق‌تر نمودارها و رفتار مدل نشان‌دهنده دو مشکل جدی است:

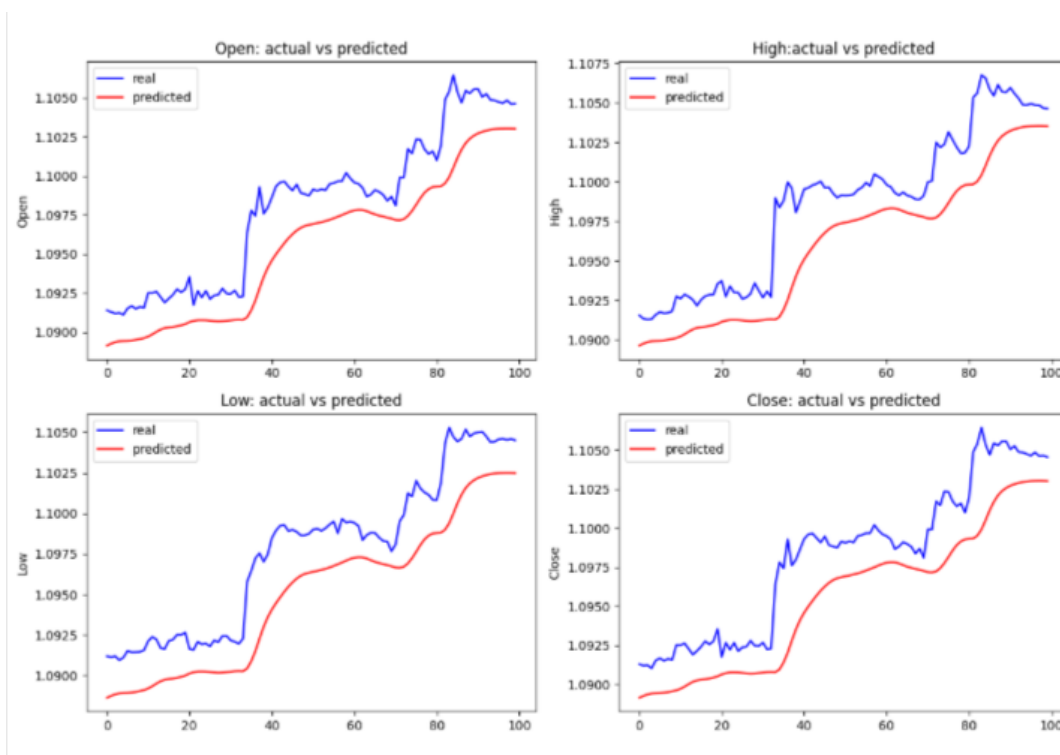


1. سادگی بیش از حد مدل:

این مدل تنها از دو متغیر ورودی (قیمت بسته شدن و حجم) برای پیش‌بینی استفاده کرده است. این محدودیت باعث شده که مدل نتواند نوسانات پیچیده و ناگهانی بازار را به خوبی یاد بگیرد. در نتیجه، پیش‌بینی‌های مدل بیش از حد نرم و هموار هستند و نمی‌توانند تغییرات سریع و کوتاه‌مدت قیمت را به درستی منعکس کنند. برای بهبود این مشکل، می‌توان از ویژگی‌های ورودی بیشتری (مانند اندیکاتورهای تکنیکال یا داده‌های بازارهای مرتبط) استفاده کرد یا معماری مدل را پیچیده‌تر کرد.

2. تاخیر (Lag) در پیش‌بینی:

یکی از مشکلات اصلی مدل، وجود تاخیر در پیش‌بینی‌هاست. مدل به جای پیش‌بینی دقیق تغییرات قیمت در تایم‌فریم بعدی، تمایل دارد مقادیری شبیه به آخرین متغیر هدف (آخرین کندل) را پیش‌بینی کند. این رفتار باعث می‌شود که مدل در واکنش به تغییرات ناگهانی بازار کند عمل کند و عملاً یک یا چند کندل عقب‌تر از داده‌های واقعی باشد. این تاخیر می‌تواند در کاربردهای واقعی (مثل معاملات فارکس) مشکل‌ساز باشد، زیرا پیش‌بینی‌های به موقع و دقیق از اهمیت بالایی برخوردارند.



مدل روند را می‌تواند دنبال کند اما تاخیر و همچنین سادگی مدل و عدم توانایی در پیش‌بینی نوسانات مشهود است.

برای رفع این چالش‌ها، در مرحله بعدی تعداد و تنوع ویژگی‌های ورودی به این امید که مدل بتواند الگوهای پیچیده‌تری را یاد بگیرد افزایش داده شد.

مدل دوم

مدل دوم، از لحاظ ساختاری و معماری همان شبکه عصبی LSTM سه لایه بود که می‌بایست مطابق قبل متغیرهای OHLC را پیش‌بینی کند. تفاوت این مدل با مدل قبل، افزایش ویژگی‌ها و متغیرهاست؛ بدین منظور اندیکاتورهایی مانند ATR، EMA، RSI، MACD، bollinger bands، Stochastic Oscillator به دیتاست اولیه اضافه شدند. همچنین تعداد کندل‌های ورودی به مدل هم از 60 به 90 افزایش پیدا کرد.

داده‌های ورودی

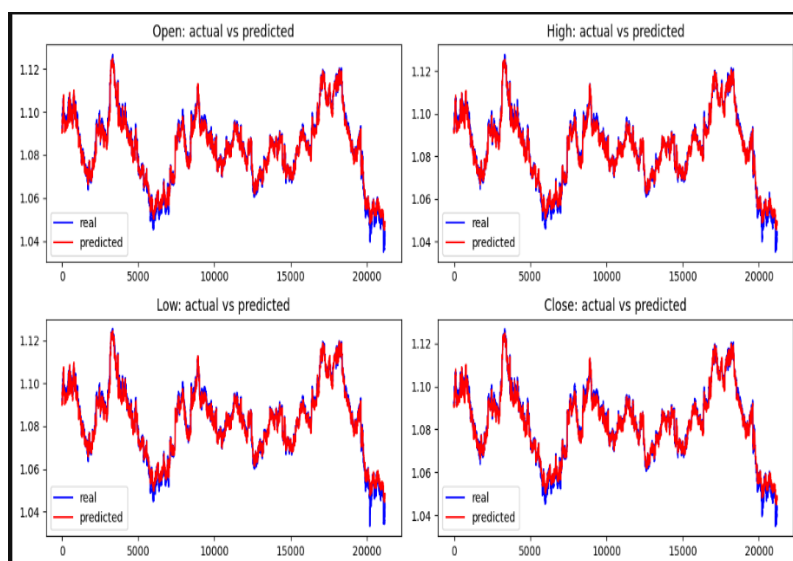
داده‌های ورودی شامل 90 کندل متوالی در تایم‌فریم 15 دقیقه‌ای بود که شامل قیمت بسته‌شدن و حجم معاملات و خروجی اندیکاتورهای ذکر شده می‌باشند.

هدف مدل

هدف این مدل، مانند مدل قبل پیش‌بینی همزمان چهار متغیر OHLC برای تایم‌فریم بعدی (15 دقیقه بعد) بر اساس داده‌های 90 تایم‌فریم قبلی بود.

نتایج مدل

در نهایت، نتایج مدل برای پیش‌بینی متغیرهای OHLC به شرح زیر شد:



	MSE	RMSE	MAE	R2
Open	2×10^{-6}	0.0015	0.0010	0.99
High	2×10^{-6}	0.0015	0.0010	0.99
Low	3×10^{-6}	0.0016	0.0012	0.99
Close	3×10^{-6}	0.0016	0.0012	0.99

تحلیل و چالش‌های مدل

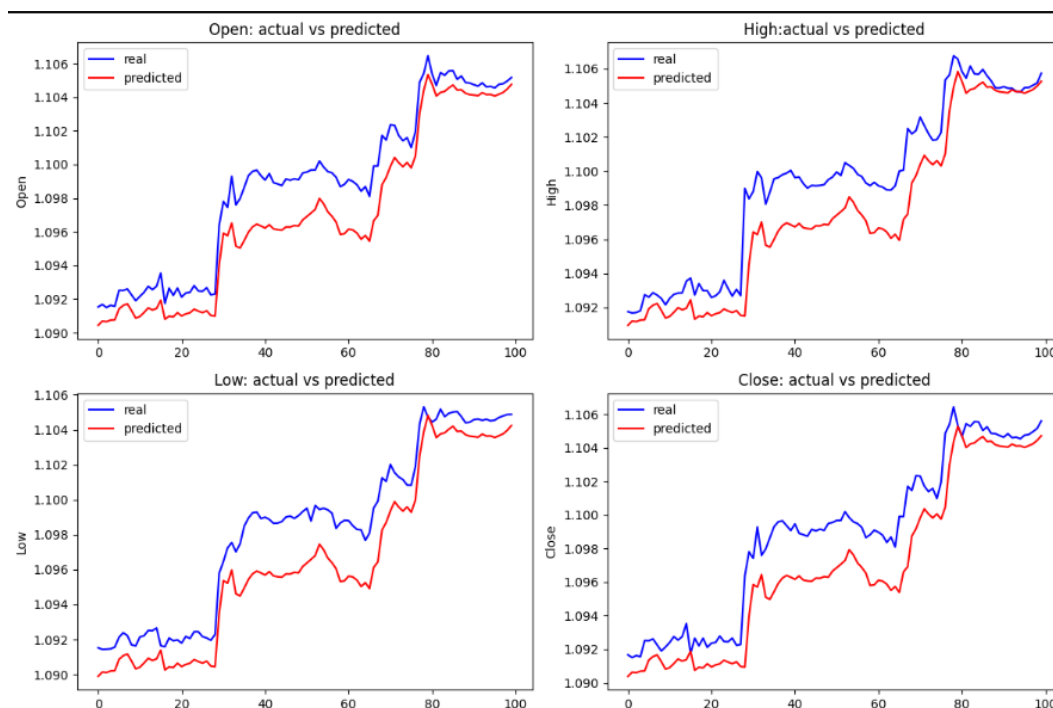
مدل دوم نتایج بهتری نسبت به مدل اول دارد. بدلیل اضافه شدن ویژگی‌های جدید به داده‌های ورودی مدل و در نتیجه پیچیده‌تر شدن مدل، نوسانات نسبت به مدل اول بهتر ره‌گیری شده‌اند.

1. سادگی بیش از حد مدل:

مدل همچنان ساده است؛ و در پیش‌بینی ظریف، ضعف دارد.

2. تاخیر (lag) در پیش‌بینی:

مشکل تاخیر در پیش‌بینی‌ها همچنان وجود دارد و در این مدل بهتر نیز دیده می‌شود. بنظر می‌رسد شبکه آموزش دیده شده متوجه اختلاف پایین مقادیر تایم فریم فعلی با تایم فریم بعدی می‌شود در نتیجه خیلی خود را برای یادگیری روند و... اذیت نمی‌کند و برای پیش‌بینی مقدار $t+1$ ، همان مقدار t را تقریباً کپی می‌کند. نتیجه در ظاهر خطای پایینی دارد اما با دقت در مدل می‌توان متوجه شد مدل در واقع چیزی یاد نمی‌گیرد!



مشکل تاخیر یا *lagging* مشهود است.

مدل سوم

از آنجایی که هدف نهایی پیش‌بینی کندل بعد یا کندل‌های بعدی بود، در این مدل تصمیم گرفته شد تا بجای اینکه OHLC را پیش‌بینی کنیم، میانگین کندل به عنوان خروجی در نظر گرفته شود به این امید که شبکه ارتباط بین ورودی‌ها و خروجی که میانگین کندل بود پیدا کند و بتواند میانگین کندل را با دقت خوبی برای گام یا گام‌های بعد پیش‌بینی کند. معماری که برای این

مدل استفاده شد، معماری ترکیبی است که از لایه‌های LSTM برای یادگیری الگوهای زمانی و از مکانیزم MultiHeadAttention برای تمرکز روی بخش‌های مهم‌تر توالی‌ها استفاده می‌کند. لایه Attention به مدل کمک می‌کند تا روی بخش‌های مهم‌تر توالی (مثلاً تایم‌فریم‌هایی که تغییرات قیمتی بیشتری دارند) تمرکز کند و وابستگی‌های پیچیده‌تر بین داده‌ها را شناسایی کند.

داده‌های ورودی

داده‌های ورودی این شبکه به صورت مقادیر اندیکاتورهای متناظر با 30 کندل در تایم فریم 30 دقیقه هستند. ایده اینکه هیچکدام از متغیرهای OHLC به عنوان ورودی به مدل داده نشده‌اند، این بود که از تنبلی مدل جلوگیری شود. یعنی مدل دوباره همان مقادیر را کپی نکرده و سعی کند رابطه‌ای بین خروجی اندیکاتورها و میانگین کندل پیدا کند.

هدف مدل

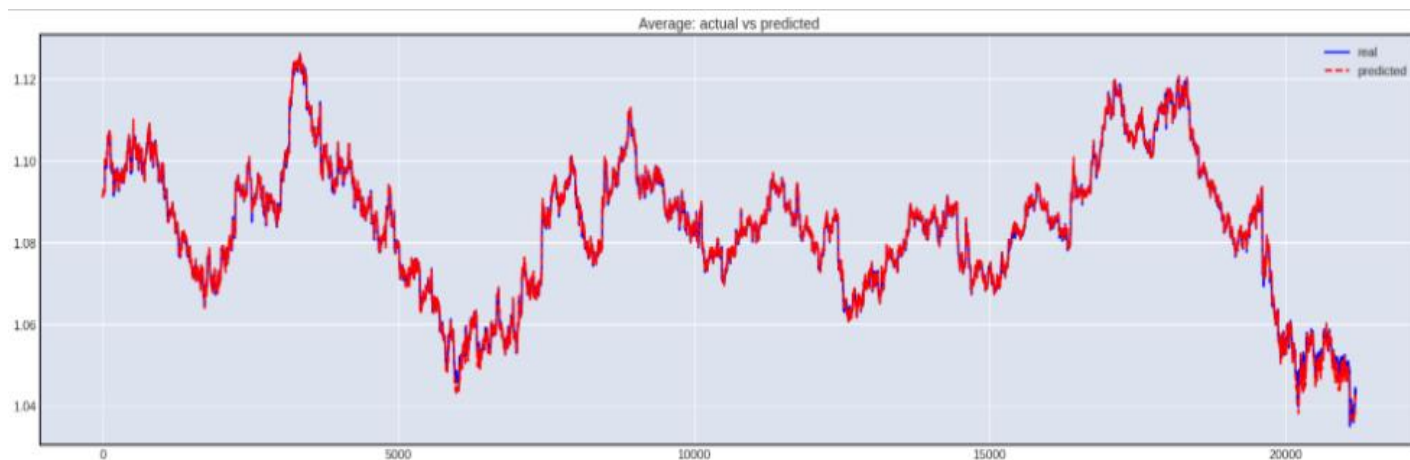
پیش‌بینی میانگین کندل بعدی (30 دقیقه بعد) بر اساس مقادیر اندیکاتورهای 30 کندل قبلی بود. ضمن اینکه خود متغیر میانگین کندل به شرح زیر محاسبه شد:

$$\text{میانگین کندل} = (\text{open} + \text{high} + \text{low} + \text{close}) / 4$$

نتایج مدل

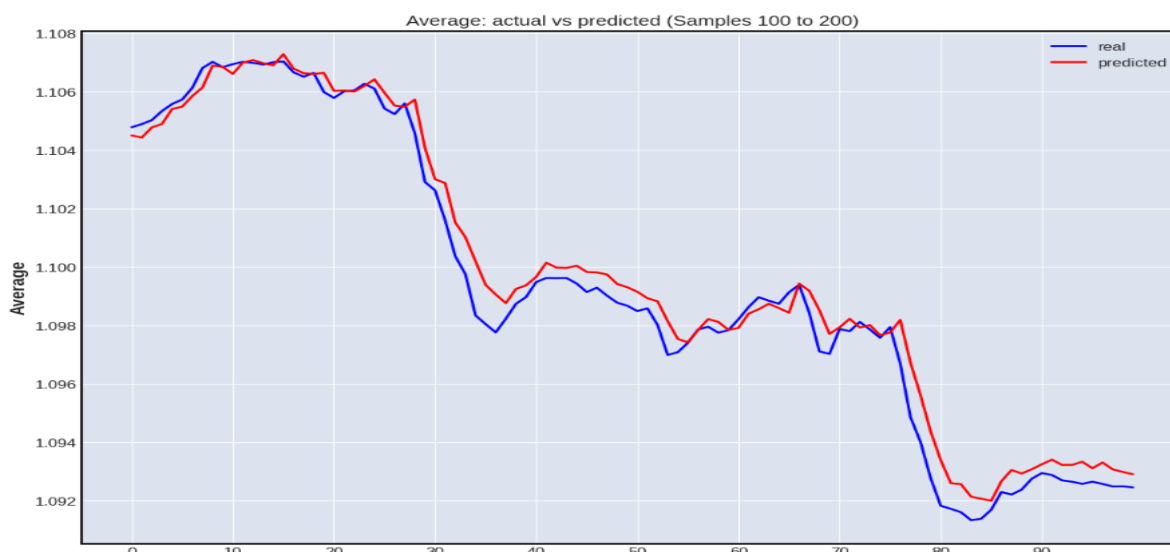
در نهایت، نتایج مدل برای پیش‌بینی میانگین کندل به شرح زیر شد:

	Average-model
MSE	5×10^{-6}
RMSE	0.0007
MAE	0.0005
R2	0.99



تحلیل و چالش‌های مدل

مدل جدید، بهتر نوسانات و روندها را دنبال می‌کند. اما همچنان با وجود اضافه کردن لایه Attention، لگ یا تاخیر در پیش‌بینی‌های مدل دیده می‌شود. با وجود انجام تغییرات متعدد در داده‌های ورودی، معماری مدل و... اما همچنان شبکه‌های LSTM تاخیر را در ذات خود داشتند.



مدل چهارم

با توجه به مشکل lagging در شبکه‌های LSTM، به سمت معماری ترنسفورمر مهاجرت کردیم. ترنسفورمرها از جمله معماری‌های تازه و قدرتمند برای پیش‌بینی سری‌های زمانی هستند که ابتدا در حوزه پردازش زبان طبیعی (NLP) معرفی شدند، اما به دلیل توانایی بالایی آن‌ها در مدل‌سازی وابستگی‌های بلندمدت و کوتاه‌مدت، به‌تازگی در پیش‌بینی سری‌های زمانی نیز بسیار مورد توجه قرار گرفته‌اند. مکانیزم Attention موجود در ترنسفورمرها به مدل اجازه می‌دهد تا به‌جای پردازش متوالی داده‌ها (مثل LSTM که داده‌ها را قدم‌به‌قدم بررسی می‌کند)، تمام نقاط داده در یک توالی را به‌صورت هم‌زمان ببیند و روی بخش‌های مهم‌تر تمرکز کند. با توجه به این توضیحات، مدل پایه خود را به ترنسفورمر تغییر دادیم تا مشکل lagging حل شود.

داده‌های ورودی

داده‌های ورودی این شبکه شامل متغیر Close و خروجی اندیکاتورهای متناظر با کندل مربوطه است. (تایم فریم 1 ساعته). ضمن اینکه طول توالی داده‌ها 128 است. یعنی مدل باید با توجه به مقادیر 128 کندل قبلی، کندل بعدی را پیش‌بینی کند.

هدف مدل

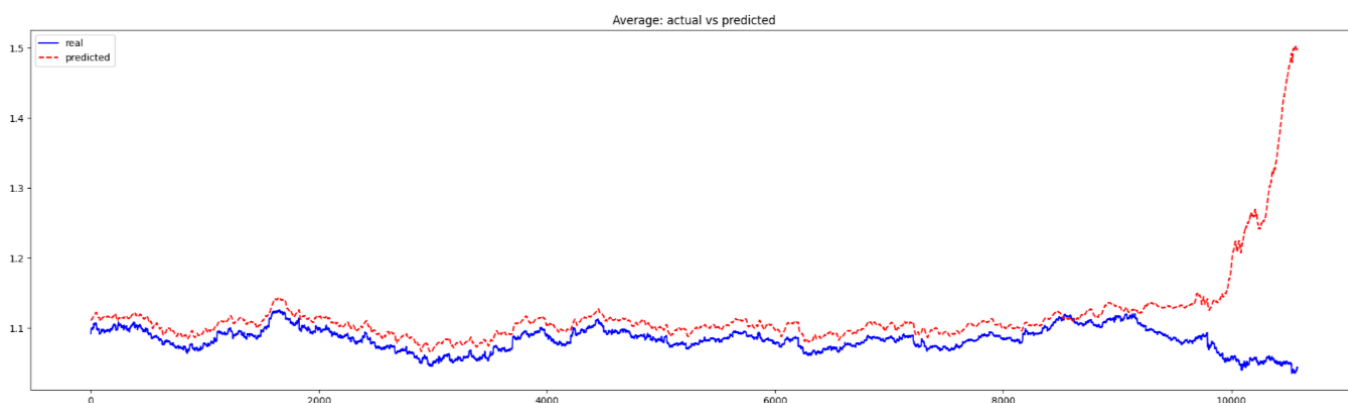
خروجی یا تارگت مدل متغیر Close است. سعی کردیم به کمک معماری ترنسفورمر Close یک ساعت بعدی را پیش‌بینی کنیم.

نتایج مدل

	Average-model
MSE	0.004
RMSE	0.06
MAE	0.0034
R2	-16.33

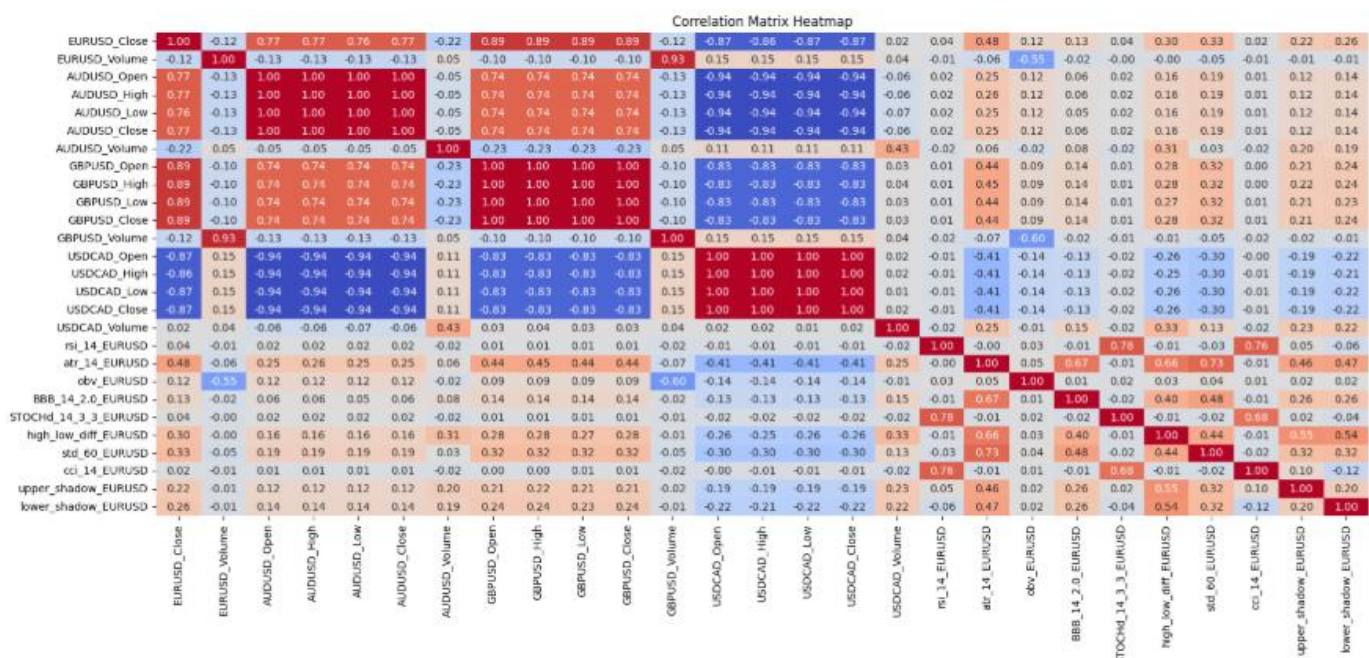
نتایج اولیه مدل که در جدول مقابل آمده‌است نسبت به مدل‌های قبلی قابل قبول نیستند،

اما مدل بعد و خلاقیت به خرج داده شده در مدل باعث بهبود چشمگیر نتایج شد



مدل پنجم

در این مدل، ابتکاری جدید در دستور کار قرار گرفت؛ به جای استفاده صرف از ویژگی‌ها و متغیرهای سنتی مانند OHLC و اندیکاتورهای تکنیکال جفت‌ارز EUR/USD، به صورت آزمایشی داده‌های جفت‌ارزهای مرتبط دیگر نیز به داده‌های ورودی اضافه شدند. این جفت‌ارزها شامل AUD/USD، GBP/USD و USD/CAD بودند که به دلیل ارتباط اقتصادی و همبستگی با EUR/USD انتخاب شدند. علاوه بر داده‌های OHLC خود این جفت‌ارزها، اندیکاتورهای تکنیکال مرتبط (مانند میانگین متحرک، RSI و MACD) نیز به مجموعه داده‌های اولیه اضافه شدند تا مدل بتواند الگوهای پیچیده‌تری را از داده‌ها استخراج کند.



اضافه کردن داده‌های جفت‌ارزهای مرتبط و اندیکاتورهای تکنیکال، یک رویکرد چندمتغیره (Multivariate) را در پیش‌بینی سری‌های زمانی ایجاد کرد. این روش به مدل اجازه می‌دهد تا به جای تمرکز صرف روی داده‌های EUR/USD، از اطلاعات بازارهای مرتبط نیز بهره‌برد و الگوهای جهانی‌تر و تاثیرات متقابل را شناسایی کند. انتظار می‌رود این ابتکار به بهبود عملکرد مدل، به‌ویژه در تشخیص تغییرات ناگهانی و کاهش تاخیر در پیش‌بینی‌ها، کمک کند.

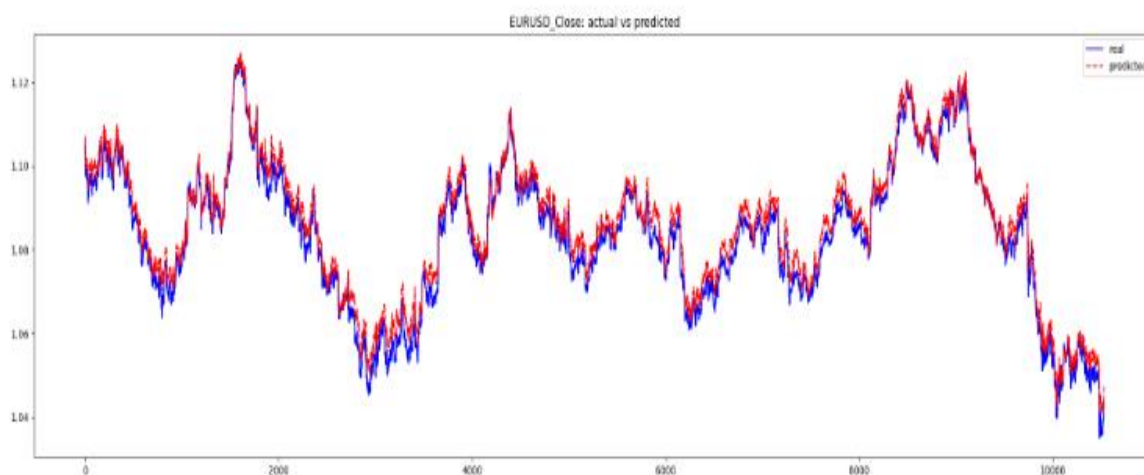
داده‌های ورودی

ورودی‌های مدل شامل داده‌های قیمتی (قیمت بسته شدن) جفت‌ارزهای EUR/USD، GBP/USD، AUD/USD و USD/CAD به همراه اندیکاتورهای تکنیکال متنوع مانند RSI، MACD، ATR، و اندیکاتورهای نوسانی (مانند سایه‌های کندل و اختلاف بالا و پایین) برای هر جفت‌ارز بود. همچنین، ویژگی‌های نسبتی (مانند نسبت‌های قیمتی بین جفت‌ارزها) و ویژگی‌های زمانی (ساعت و روز هفته) به داده‌ها اضافه شدند. در مجموع، 100 ویژگی برای بهبود پیش‌بینی استفاده شد.

هدف مدل

خروجی یا تارگت مدل متغیر Close است. سعی کردیم به کمک معماری ترنسفورمر و انبوه ویژگی‌ها Close یک ساعت بعدی را پیش‌بینی کنیم.

	Average-model
MSE	1×10^{-6}
RMSE	0.003
MAE	0.0027
R2	0.96



تحلیل و چالش‌های مدل

با توجه به تعداد بالای ویژگی‌های مدل، این شبکه خیلی بهتر از مدل‌های قبل نوسانات را می‌تواند نمایش دهد. اما، همچنان معضل تاخیر در شبکه وجود دارد بنظر می‌رسد هرچقدر شبکه‌های عصبی را پیچیده‌تر سازیم یا به تعداد ویژگی‌ها بیافزاییم شاید در یک روند طولانی خط پیش‌بینی کاملاً فیت باشد و معیارهای ارزیابی و خطا در پایین‌ترین مقدار باشند، اما همچنان مدل خیلی به خودش زحمت یادگیری نمی‌دهد و در نهایت سعی می‌کند مقادیر کندل‌های قبلی را تکرار کند.

