

به نام خدا



دانشگاه تهران



دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین چهارم

نام و نام خانوادگی	مهرسا همتپناه – محمد هادی بابالو
شماره دانشجویی	۸۱۰۱۹۹۳۸۰ - ۸۱۰۱۹۹۵۸۴
تاریخ ارسال گزارش	۱۴۰۲، ۱۰، ۰۴

فهرست

۴	پاسخ ۱. پیش‌بینی سری زمانی
۴	۱-۱. دانلود داده‌ها
۵	۱-۲. کاوش در داده‌های سری زمانی و آشنایی با تئوری‌ها و کتابخانه‌های معروف
۶	۳-۱ TimeSeriesSplit
۹	۴-۱. آماده‌سازی ورودی و خروجی مدل
۱۰	۴-۵. مدل‌های شبکه عصبی حافظه‌دار
۲۳	۶-۱ Naïve Forecast
۲۴	پاسخ ۲ - پیش‌بینی افکار خودکشی در رسانه‌های اجتماعی
۲۵	۱-۲. پیش‌پردازش داده
۲۹	۲-۲. ساخت ماتریس جاسازی
۳۰	۳-۲. آموزش مدل‌های یادگیری عمیق
۳۱	۴-۲. مقایسه نتایج

شکل‌ها

- شکل ۱ - لیست اسامی تمام سهم‌های موجود در SP500 ۴
- شکل ۲ - لیست سهام‌هایی که از سال ۲۰۱۰ رکورد شده ۴
- شکل ۳ - داده‌های دانلود شده سهام‌ها ۵
- شکل ۴ - تعداد داده‌های null در مقادیر سهام‌ها ۵
- شکل ۵ - تعداد داده‌های null در هر روز ۶
- شکل ۶ - نمودار هیستوگرام close price return ۷
- شکل ۷ - نمایش داده‌ها به صورت CandleStick ۸
- شکل ۸ - cross-validation در داده سری زمانی ۹
- شکل ۹ - معماری مدل LSTM ۱۰
- شکل ۱۰ - معماری شبکه GRU ۱۱
- شکل ۱۱ - معماری شبکه Bidirectional LSTM ۱۳
- شکل ۱۲ - پیش‌بینی سهم AMZN در split برابر با ۱ و seed برابر با ۰ ۱۵
- شکل ۱۳ - پیش‌بینی سهم AMZN در split برابر با ۲ و seed برابر با ۰ ۱۵
- شکل ۱۴ - پیش‌بینی سهم AMZN در split برابر با ۳ و seed برابر با ۰ ۱۶
- شکل ۱۵ - پیش‌بینی سهم IBM در split برابر با ۱ و seed برابر با ۰ ۱۶
- شکل ۱۶ - پیش‌بینی سهم IBM در split برابر با ۲ و seed برابر با ۰ ۱۶
- شکل ۱۷ - پیش‌بینی سهم IBM در split برابر با ۳ و seed برابر با ۰ ۱۷
- شکل ۱۸ - پیش‌بینی سهم KO در split برابر با ۱ و seed برابر با ۰ ۱۷
- شکل ۱۹ - پیش‌بینی سهم KO در split برابر با ۲ و seed برابر با ۰ ۱۷
- شکل ۲۰ - پیش‌بینی سهم KO در split برابر با ۳ و seed برابر با ۰ ۱۸
- شکل ۲۱ - پیش‌بینی سهم AAPL در split برابر با ۱ و seed برابر با ۰ ۱۸
- شکل ۲۲ - پیش‌بینی سهم AAPL در split برابر با ۲ و seed برابر با ۰ ۱۸
- شکل ۲۳ - پیش‌بینی سهم AAPL در split برابر با ۳ و seed برابر با ۰ ۱۹
- شکل ۲۴ - پیش‌بینی سهم MFST در split برابر با ۱ و seed برابر با ۰ ۱۹
- شکل ۲۵ - پیش‌بینی سهم MFST در split برابر با ۲ و seed برابر با ۰ ۱۹
- شکل ۲۶ - پیش‌بینی سهم MFST در split برابر با ۳ و seed برابر با ۰ ۲۰
- شکل ۲۷ - نتایج معیارها برای سهام AMZN ۲۰

۲۰ شکل ۲۸ - نتایج معیارها برای سهام AMZN
۲۱ شکل ۲۹ - نتایج معیارها برای سهام IBM
۲۱ شکل ۳۰ - نتایج معیارها برای سهام IBM
۲۱ شکل ۳۱ - نتایج معیارها برای سهام KO
۲۲ شکل ۳۲ - نتایج معیارها برای سهام KO
۲۲ شکل ۳۳ - نتایج معیارها برای سهام AAPL
۲۲ شکل ۳۴ - نتایج معیارها برای سهام AAPL
۲۳ شکل ۳۵ - نتایج معیارها برای سهام MSFT
۲۳ شکل ۳۶ - نتایج معیارها برای سهام MSFT
۲۴ شکل ۳۷ - نتایج معیارهای مختلف بر روی سهامها برای مدل Random Walk
۲۵ شکل ۳۸ - توزیع کلاس‌ها
۲۶ شکل ۳۹ - word cloud برای توییت‌های non-suicidal
۲۶ شکل ۴۰ - word cloud برای توییت‌های suicidal
۲۷ شکل ۴۱ - نمونه توییت قبل و بعد از پیش‌پردازش
۲۷ شکل ۴۲ - نمودار توزیع طول توییت
۲۸ شکل ۴۳ - نمودار توزیع کلمات در توییت‌های non-suicidal
۲۹ شکل ۴۴ - نمودار توزیع کلمات در توییت‌های suicidal
۳۰ شکل ۴۵ - لود کردن مدل word2vec
۳۰ شکل ۴۶ - حلقه آموزش مدل‌ها
۳۱ شکل ۴۷ - ارزیابی عملکرد مدل در حین آموزش
۳۱ شکل ۴۸ - نمودار accuracy و loss برای LSTM روی داده آموزش
۳۲ شکل ۴۹ - نمودار accuracy و loss برای LSTM روی داده ارزیابی
۳۲ شکل ۵۰ - نمودار accuracy و loss برای 2 Layer LSTM روی داده آموزش
۳۳ شکل ۵۱ - نمودار accuracy و loss برای 2 Layer LSTM روی داده ارزیابی
۳۳ شکل ۵۲ - نمودار accuracy و loss برای CNN + 2 Layer LSTM روی داده آموزش
۳۴ شکل ۵۳ - نمودار accuracy و loss برای CNN + 2 Layer LSTM روی داده ارزیابی
۳۵ شکل ۵۴ - نمودار accuracy و loss برای LSTM
۳۶ شکل ۵۵ - نمودار accuracy و loss برای 2 Layer LSTM
۳۷ شکل ۵۶ - نمودار accuracy و loss برای CNN + 2 Layer LSTM

پاسخ ۱. پیش‌بینی سری زمانی

۱-۱. دانلود داده‌ها

در این قسمت با استفاده از کتابخانه yahoo finance و کد های کمکی داده شده در صورت پروژه، داده را دانلود می کنیم.

	Symbol	Security	GICS Sector	GICS Sub-Industry	Headquarters Location	Date added	CIK	Founded
0	MMM	3M	Industrials	Industrial Conglomerates	Saint Paul, Minnesota	1957-03-04	66740	1902
1	AOS	A. O. Smith	Industrials	Building Products	Milwaukee, Wisconsin	2017-07-26	91142	1916
2	ABT	Abbott	Health Care	Health Care Equipment	North Chicago, Illinois	1957-03-04	1800	1888
3	ABBV	AbbVie	Health Care	Biotechnology	North Chicago, Illinois	2012-12-31	1551152	2013 (1888)
4	ACN	Accenture	Information Technology	IT Consulting & Other Services	Dublin, Ireland	2011-07-06	1467373	1989
...
498	YUM	Yum! Brands	Consumer Discretionary	Restaurants	Louisville, Kentucky	1997-10-06	1041061	1997
499	ZBRA	Zebra Technologies	Information Technology	Electronic Equipment & Instruments	Lincolnshire, Illinois	2019-12-23	877212	1969
500	ZBH	Zimmer Biomet	Health Care	Health Care Equipment	Warsaw, Indiana	2001-08-07	1136869	1927
501	ZION	Zions Bancorporation	Financials	Regional Banks	Salt Lake City, Utah	2001-06-22	109380	1873
502	ZTS	Zoetis	Health Care	Pharmaceuticals	Parsippany, New Jersey	2013-06-21	1555280	1952

503 rows × 8 columns

شکل ۱ - لیست اسامی تمام سهم های موجود در SP500

	Symbol	Security	GICS Sector	GICS Sub-Industry	Headquarters Location	Date added	CIK	Founded
0	MMM	3M	Industrials	Industrial Conglomerates	Saint Paul, Minnesota	1957-03-04	66740	1902
2	ABT	Abbott	Health Care	Health Care Equipment	North Chicago, Illinois	1957-03-04	1800	1888
5	ADBE	Adobe Inc.	Information Technology	Application Software	San Jose, California	1997-05-05	796343	1982
7	AES	AES Corporation	Utilities	Independent Power Producers & Energy Traders	Arlington, Virginia	1998-10-02	874761	1981
8	AFL	Aflac	Financials	Life & Health Insurance	Columbus, Georgia	1999-05-28	4977	1955
...
495	WYNN	Wynn Resorts	Consumer Discretionary	Casinos & Gaming	Paradise, Nevada	2008-11-14	1174922	2002
496	XEL	Xcel Energy	Utilities	Multi-Utilities	Minneapolis, Minnesota	1957-03-04	72903	1909
498	YUM	Yum! Brands	Consumer Discretionary	Restaurants	Louisville, Kentucky	1997-10-06	1041061	1997
500	ZBH	Zimmer Biomet	Health Care	Health Care Equipment	Warsaw, Indiana	2001-08-07	1136869	1927
501	ZION	Zions Bancorporation	Financials	Regional Banks	Salt Lake City, Utah	2001-06-22	109380	1873

290 rows × 8 columns

شکل ۲- لیست سهام هایی که از سال ۲۰۱۰ رکورد شده

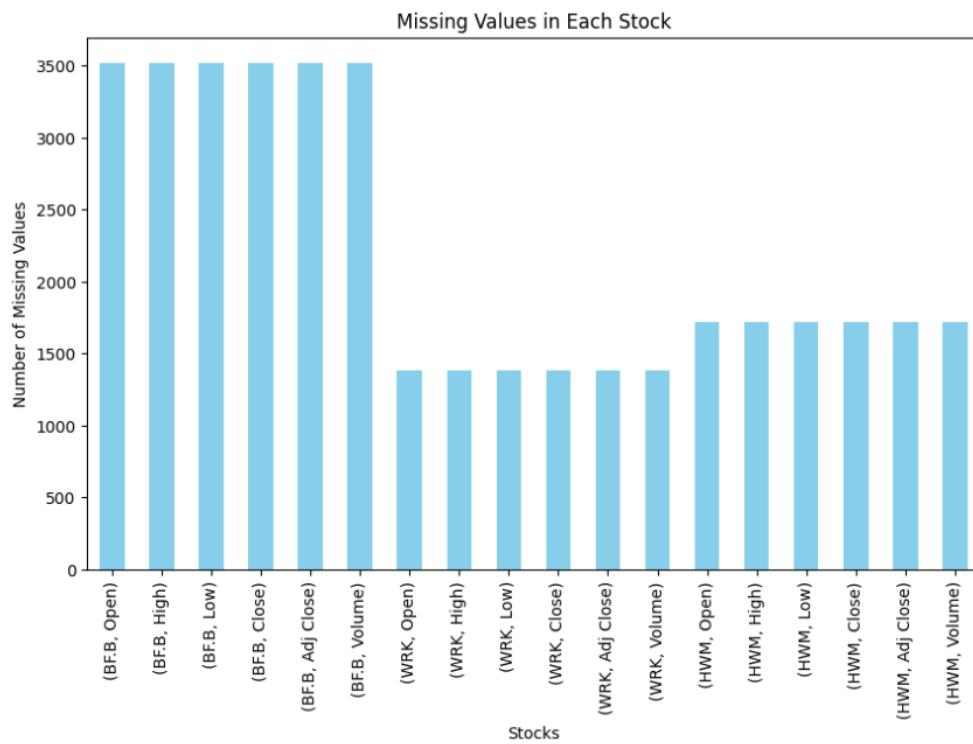
	SYK						KEY				...	HD
Date	Open	High	Low	Close	Adj Close	Volume	Open	High	Low	Close	...	Low
2010-01-04	51.279999	52.270000	50.830002	52.160000	43.596558	3982800	5.66	5.97	5.65	5.94	...	28.5
2010-01-05	52.330002	52.689999	51.930000	52.650002	44.006107	2238300	5.88	6.19	5.88	6.17	...	28.2
2010-01-06	52.660000	53.529999	52.500000	53.459999	44.683128	3250400	6.17	6.20	6.06	6.13	...	28.7
2010-01-07	53.349998	55.360001	53.349998	55.250000	46.179245	4095500	6.07	6.48	6.04	6.39	...	28.7
2010-01-08	55.680000	56.119999	55.209999	55.419998	46.321320	3535500	6.35	6.56	6.34	6.50	...	28.6
...
2023-12-18	291.390015	292.359985	288.500000	291.470001	291.470001	1416000	14.44	14.46	14.10	14.11	...	350
2023-12-19	291.059998	292.339996	289.660004	291.980011	291.980011	1433800	14.15	14.41	14.04	14.36	...	350
2023-12-20	292.250000	293.839996	288.149994	288.309998	288.309998	1162400	14.40	14.54	14.07	14.08	...	348
2023-12-21	289.339996	296.089996	288.899994	296.029999	296.029999	1046200	14.27	14.39	14.15	14.28	...	347
2023-12-22	297.619995	298.420013	294.230011	297.500000	297.500000	1068700	14.37	14.51	14.20	14.29	...	346

3518 rows × 1740 columns

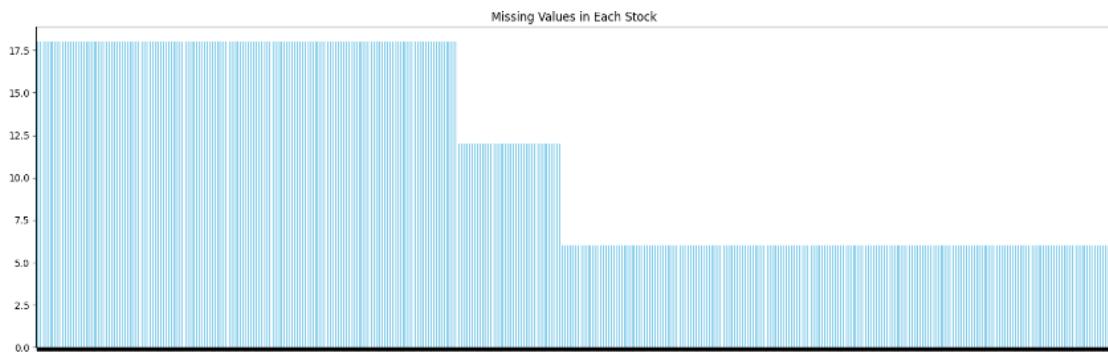
شکل ۳ - داده های دانلود شده سهام ها

۱-۲. کاوش در داده های سری زمانی و آشنایی با تئوری ها و کتابخانه های معروف

بطور کلی در داده های دانلود شده ۳۹۶۹۰ داده null داریم که توزیع آن ها در شکل های زیر قابل مشاهده است. برای خوانا تر شدن شکل ها در آن ها تنها سهام ها با مقدار داده null آورده شده اند.



شکل ۴ - تعداد داده های null در مقادیر سهام ها



شکل ۵ – تعداد داده های **null** در هر روز

بطور کلی در مواجهه با داده های null value دو راه وجود دارد: کل رکورد حاوی اطلاعات را حذف کنیم یا اطلاعات گمشده را با استفاده از روش های مختلف پر کنیم که روش دوم ممکن است باعث ایجاد بایاس شود. از آنجایی که داده های time series دارای ویژگی زمانی هستند، تنها برخی از روش های آماری برای داده های سری زمانی مناسب هستند.

می توان مقادیر گمشده را با مقدار میانگین (mean) یا میانه (median) داده ها در ستون دارای مقدار null ، پر کرد. اما در شرایطی که سری زمانی دارای مؤلفه های فصلی و روند باشد، این تکنیک ها به درستی کار نمی کنند. این بدین دلیل است که مؤلفه های فصلی و روند در هنگام نسبت دادن داده های از دست رفته در این روش ها در نظر گرفته نمی شوند. همچنین بیشتر داده های بورس نیز روند افزایشی یا کاهشی و فصلی دارند.

روش های دیگر Backward Fill (NOCB) و Forward Fill (LOCF) هستند. در روش اول یعنی LOCF مقدار داده گمشده با داده قبل غیر null آن پر می شود. این نشان دهنده این باور است که اگر یک مقدار گم شده باشد، بهترین حدس این است که از آخرین باری که اندازه گیری شده است متفاوت نباشد. در روش دوم یعنی NOCB مقدار داده گمشده با اولین داده غیر null بعد از آن پر می شود. با توجه به توضیحات می توان دریافت که بهتر است برای داده های null ای که در وسط یا انتهای سری زمانی هستند و تاریخ شروع قطعی دارند، از پر کردن به جلو (LOCF) و برای داده های null ای که شروعی ندارند و در ابتداء سری زمانی قرار دارند، از پر کردن به عقب (NOCB) استفاده کنیم.

روش های دیگر که برای داده های null پراکنده کاربرد بیشتری دارد، درون یابی خطی و درون یابی اسپلاین (Spline Interpolation) هستند. در درون یابی برای تخمین مقادیر گمشده به داده های قبل و بعد از مقدار گمشده نگاه می کنیم.

تمامی متدها زمانی که داده گمشده در میانه سری زمانی باشد کاربرد دارند.

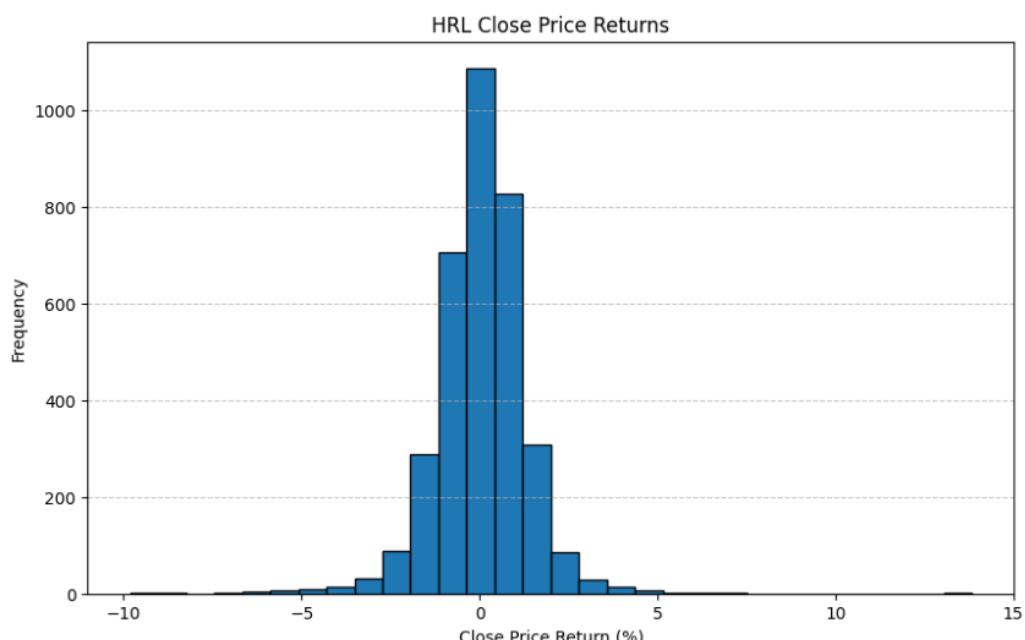
در ادامه نمودار هیستوگرام close price return را برای یک سهام تصادفی رسم کردیم. شکل زیر نشان دهنده نمودار برای سهم HRL است.

همانطور در شکل نیز قابل مشاهده است، نمودار توزیع شبه نرمال دارد.

طبق قضیه حد مرکزی می دانیم که مجموع (یا میانگین) تعداد زیادی از متغیرهای تصادفی مستقل و با توزیع یکسان ، بدون توجه به شکل توزیع اصلی آن متغیرها، به سمت یک توزیع نرمال میل می کند. در نتیجه توزیع close price return به دلیل قضیه حد مرکزی نرمال فرض می شود که این می تواند نشان دهنده این باشد که تغییرات متوالی قیمت سهم ها مستقل و غیرقابل پیش بینی هستند. در زمینه قیمت سهام، این متغیرها می توانند عوامل متعددی مانند اخبار بازار، شاخص های اقتصادی و احساسات سرمایه گذاران باشند.

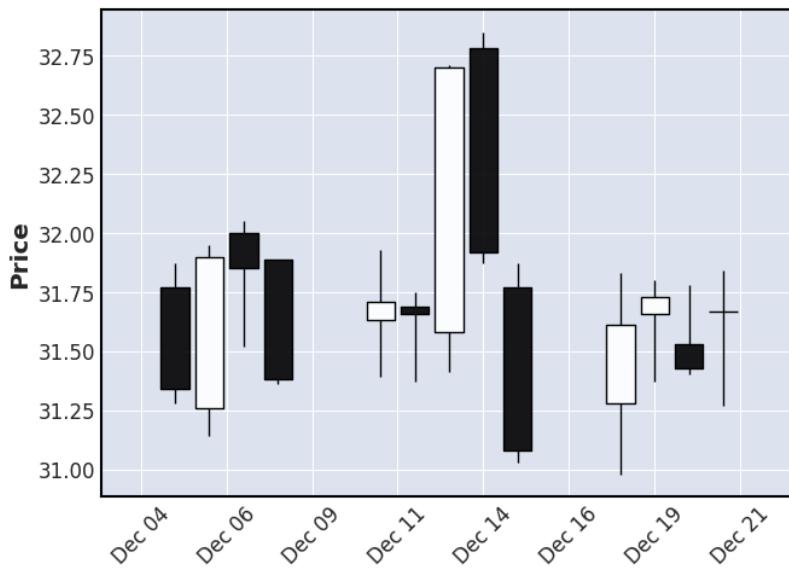
از طرفی همان طور که در صورت پژوهه هم توضیح داده شده است نظریه گام تصادفی یا Random Walk Theory بیان می کند که تغییرات قیمت سهام دارای توزیع یکسانی بوده و مستقل از همدیگر هستند و هیچ الگوی خاصی ندارند.

در نتیجه ارتباط این نمودار و نظریه گام تصادفی از این فرض ناشی می شود که تغییرات متوالی قیمت مستقل هستند، که به این معنی است که تغییرات قیمت گذشته نمی تواند تغییرات قیمت را در آینده پیش بینی کند. به عبارت دیگر، اگر تغییرات قیمت سهام به صورت تصادفی باشد، به دلیل قضیه حد مرکزی غیرقابل پیش بینی بوده و از توزیع نرمال پیروی می کند.



شکل ۶ - نمودار هیستوگرام close price return

شکل زیر نشان دهنده داده ها به صورت CandleStick با کمک کتابخانه MPL Finance است. برای خوانا بودن شکل تنها داده های یک ماه اخیر در نظر گرفته شده است.



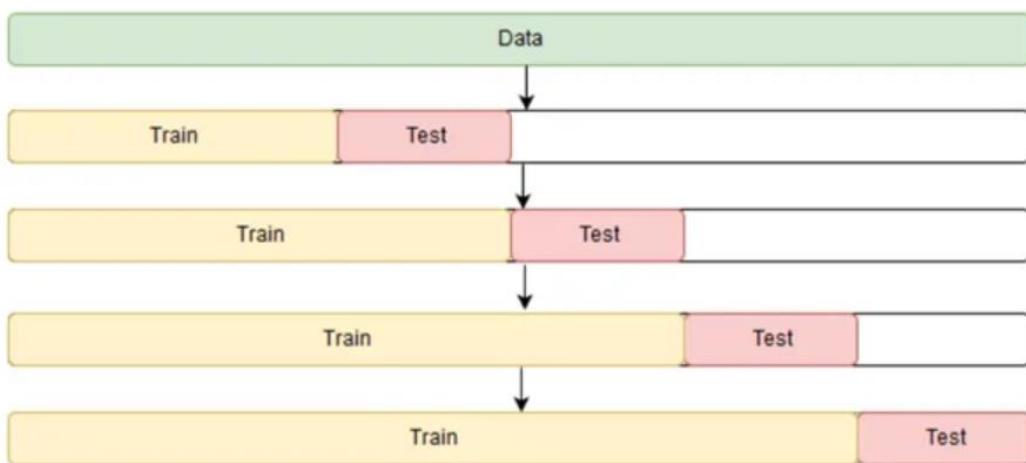
شکل ۷ - نمایش داده ها به صورت CandleStick

TimeSeriesSplit .۳-۱

روش cross-validation یک روش آماری است که برای تخمین عملکرد مدل های یادگیری ماشین استفاده می شود. در cross-validation داده های رایج در یادگیری ماشین نمونه هایی از داده ها به صورت تصادفی و رندوم انتخاب می شوند و سپس نمونه های انتخاب شده به دو مجموعه داده آموزش و ارزیابی تقسیم می شوند. به عنوان مثال، در K-fold cross-validation روش، مجموعه داده به تعدادی قسمت به نام fold تقسیم می شود. این مدل بر روی تمام fold ها به جز یکی آموزش داده می شود و روی fold باقیمانده آزمایش می شود. این فرآیند تا زمانی که مدل روی هر یک از fold ها تست شود تکرار می شود. اما در اعتبارسنجی داده های سری زمانی روش cross-validation مقداری متفاوت است زیرا در این داده ها هدف ما پیش بینی مقداری در روزهای بعدی برای یک سری زمانی با استفاده از داده های روزهای پیشین است. در نتیجه در این داده ها، مجموعه داده آموزش و ارزیابی به صورت تصادفی انتخاب نمی شوند زیرا استفاده از مقادیر و قیمت های آینده برای پیش بینی مقادیر در گذشته معنی ندارد. در حقیقت look ahead bias زمانی اتفاق می افتد که داده هایی که در آن زمان به راحتی در دسترس نبودند در شبیه سازی آن دوره زمانی استفاده شوند. در نتیجه اگر در cross-validation داده های سری زمانی از اطلاعات و قیمت های آینده برای پیش بینی قیمت سهم در گذشته یا حال استفاده کنیم، look ahead bias را

خواهیم داشت. در داده‌های سری زمانی، داده‌ها اغلب به داده‌های قبلی وابسته هستند و ترتیب زمانی برای مدل‌سازی و پیش‌بینی مقادیر آینده بسیار مهم است.

برای cross-validation در این داده‌ها با زیرمجموعه کوچکی از داده‌ها برای داده آموزشی شروع می‌کنیم و داده‌های روزهای بعدی را پیش‌بینی می‌کنیم. سپس همان نقاط داده پیش‌بینی شده که در قبل بخشی از داده تست بودند به عنوان بخشی از مجموعه داده آموزشی بعدی گنجانده می‌شود و نقاط داده بعدی پیش‌بینی می‌شود.



شکل ۸ - cross-validation در داده سری زمانی

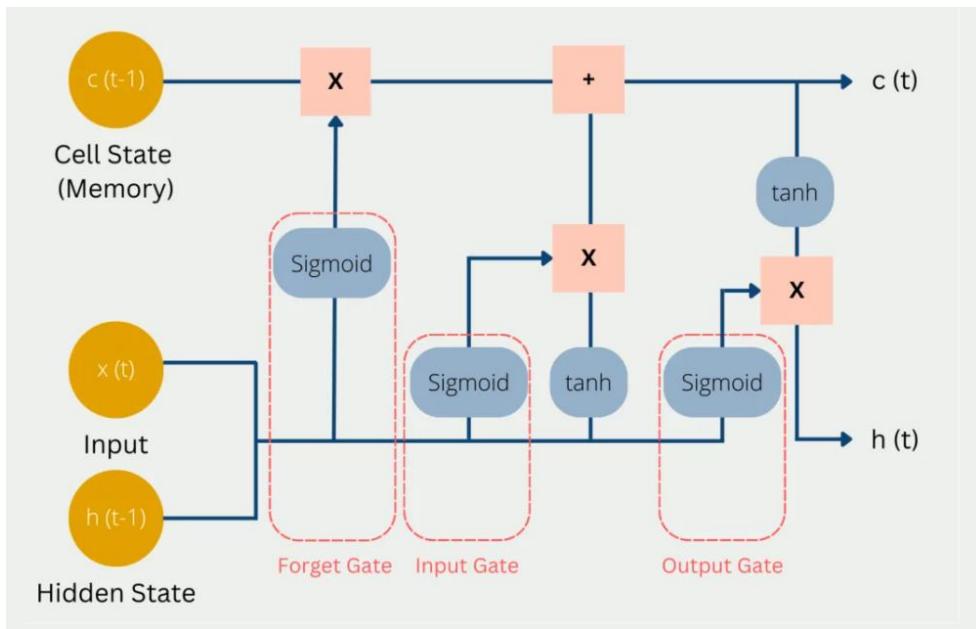
در این مرحله از کد داده شده در صورت پروژه داده را برای cross validation شدن آماده می‌کنیم. به دلیل طولانی بودن مدت زمان اجرای کد مقدار cv_n_split برابر یا ۳ قرار داده شده است.

۴-۱. آماده‌سازی ورودی و خروجی مدل

در این قسمت داده‌ها با توجه به توضیحات داده شده آماده شده اند با این تفاوت که برای سرعت بخشیدن به آموزش تنها ۳ مقدار برای seed در نظر گرفته شده است.

۱-۵. مدل‌های شبکه عصبی حافظه‌دار

LSTM •



شکل ۹ – معماری مدل LSTM

شبکه عصبی LSTM مانند شبکه RNN به صورت زنجیره‌ای پشت سرهم قرار می‌گیرد. بطور کلی شبکه‌های RNN دارای یک حلقه بازخورد هستند که به آن‌ها اجازه می‌دهد اطلاعات ورودی‌های قبلی را ذخیره کنند و این قابلیت آن‌ها را برای کارهایی مانند پیش‌بینی توالی، مدل‌سازی زبان و تشخیص گفتار مناسب می‌کند. دلیل طراحی LSTM و مشکل اصلی شبکه RNN حافظه کوتاه آن است که در نتیجه آن در بلند مدت شبکه RNN توانایی یادگیری اطلاعاتی را که در گام‌های زمانی بسیار قبل‌تر به شبکه داده شده است را از دست می‌دهد. دلیل این موضوع مفهومی به‌نام محوش‌گی گرادیان (Gradient Vanishing) است.

شبکه LSTM از cell و گیت‌ها تشکیل شده است. اطلاعات در یک cell ذخیره می‌شود و تصمیم می‌گیرد که چه چیزی را ذخیره کند، پاک کند، بخواند و بنویسد. این عملیات‌ها از طریق گیت‌هایی امکان پذیرند. همانطور که در شکل ۹ قابل مشاهده است در این مدل سه گیت forget, input و output را داریم.

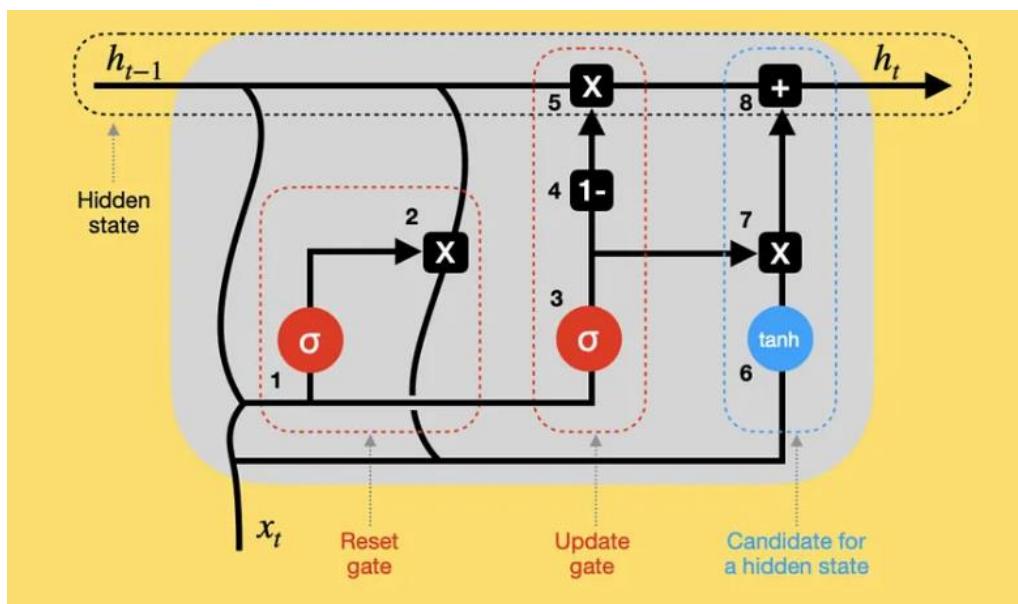
- گیت forget : این گیت تصمیم می‌گیرد کدام اطلاعات حفظ و کدام فراموش شود. اطلاعات ورودی گام جدید به همراه اطلاعات حالت نهان (Hidden State) گام قبلی به این گیت وارد می‌شوند و از تابع Sigmoid عبور می‌کنند. خروجی این تابع پس از عبور از Sigmoid میان

صفر تا ۱ قرار می گیرد. هر قدر عدد خروجی به صفر نزدیک‌تر باشد یعنی باید اطلاعات فراموش شود و هر قدر به ۱ نزدیک‌تر باشد یعنی باید حفظ شود.

- گیت **input** : این گیت برای به روزرسانی مقادیر موجود در cell state تعییه شده است. اطلاعات ورودی گام جدید، به همراه اطلاعات hidden state گام قبلی، به این گیت وارد می‌شوند و از تابع Sigmoid عبور می‌کنند تا این تابع تصمیم بگیرد کدام اطلاعات دور انداخته و کدام به روزرسانی شوند. همچنین اطلاعات ورودی گام جدید، به همراه اطلاعات hidden state گام قبلی، به تابع Tanh وارد می‌شوند. درنهایت خروجی تابع Sigmoid و Tanh با هم ضرب می‌شوند تا تابع Sigmoid تصمیم بگیرد چه مقادیری از خروجی تابع Tanh باید حفظ شوند.
- گیت **output** : این گیت درنهایت تصمیم می‌گیرد که hidden state بعدی چه مقداری باشد. اطلاعات ورودی‌های قبلی را همراه خودش دارد. در ابتدا اطلاعات ورودی گام جدید به همراه اطلاعات hidden state گام قبلی به تابع Sigmoid وارد می‌شوند. مقدار آپدیت شده‌ی cell state به تابع Tanh وارد می‌شود. خروجی این دو تابع با هم ضرب می‌شود تا تصمیم گرفته شود چه اطلاعاتی را با خودش به گام بعدی ببرد. درنهایت cell state جدید و hidden state جدید به گام زمانی بعدی منتقل می‌شوند.

در این شبکه در هر مرحله محاسباتی، ورودی فعلی، حالت قبلی حافظه کوتاه مدت و حالت قبلی حالت پنهان را داریم که در واقع در حافظه طولانی مدت طی می‌شود. همچنین LSTM از دو مسیر برای پیش‌بینی استفاده می‌کند که یک مسیر برای حافظه طولانی مدت و یک مسیر برای حافظه کوتاه مدت است.

GRU •



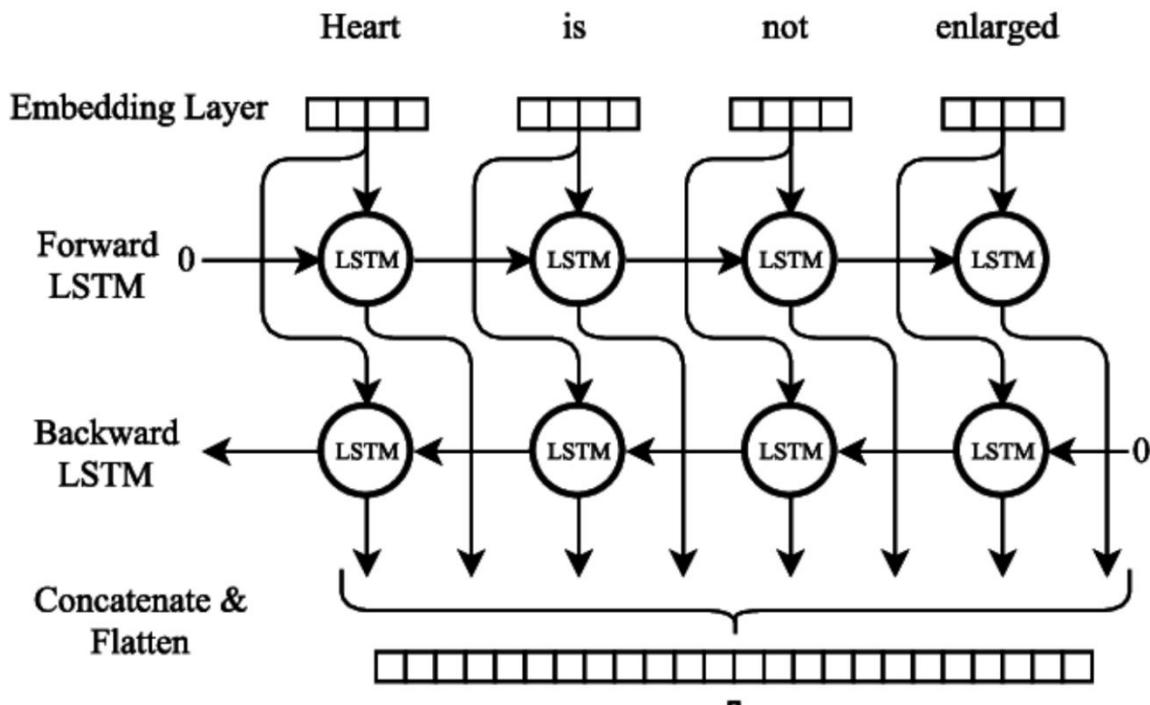
شکل ۱۰ – معماری شبکه GRU

شبکه عصبی واحد بازگشتی گیتی (Gated Recurrent Unit) یا به اختصار شبکه GRU نوع خاصی از شبکه عصبی بازگشتی RNN است. این شبکه تا حد زیادی مشابه شبکه عصبی LSTM است.

این شبکه عصبی بسیار مشابه شبکه LSTM است که در قسمت قبل توضیح داده شد، با این تفاوت که به جای سه گیت، فقط دو گیت تنظیم مجدد (Reset Gate) و گیت بهروزرسانی (Update Gate) دارد؛ همچنین شبکه GRU چیزی به نام حالت سلول (Cell State) ندارد و برای انتقال اطلاعات از حالت Nehan (Hidden State) استفاده می‌کند. تصویر ۱۰ ساختار شبکه GRU را نشان می‌دهد. GRU پیچیدگی کمتری نسبت به LSTM دارد زیرا گیت‌های کمتری دارد و این باعث می‌شود GRU از حافظه کمتری استفاده کند، سریعتر اجرا شود و سریعتر از LSTM آموزش ببیند. به طور کلی، GRU در داده‌های time series با پیچیدگی کم، از شبکه‌های LSTM بهتر عمل می‌کند، در حالی که در داده‌های time series با پیچیدگی بالا، LSTM بهتر عمل می‌کند. اگر مجموعه داده کوچک باشد، GRU ترجیح داده می‌شود، در غیر این صورت، از LSTM برای مجموعه داده‌های بزرگتر استفاده می‌شود.

○ گیت بهروزرسانی : این گیت دقیقاً مانند دو گیت فراموشی (Forget) و ورودی (Input) در شبکه LSTM عمل می‌کند. این گیت تصمیم می‌گیرد چه مقدار از اطلاعات گذشته، یعنی اطلاعاتی که در گام‌های قبلی داشتیم، به شبکه اضافه شود. در این گیت مقدار ورودی جدید (x_t) به همراه مقدار حالت Nehan گام قبلی (h_{t-1}) در وزن متناظر خود ضرب و سپس با هم جمع می‌شوند و به یک تابع sigmoid وارد می‌شوند تا خروجی میان بازه ۰ و ۱ قرار بگیرد. در زمان آموزش شبکه این وزن‌ها هر بار بهروزرسانی می‌شوند تا فقط اطلاعات مفید به شبکه اضافه شوند.

○ گیت تنظیم مجدد reset gate : این گیت تصمیم می‌گیرد چه مقدار از اطلاعات گذشته، یعنی اطلاعات گام‌های قبلی، فراموش شود. در اینجا هم مانند گیت قبل، مقدار ورودی جدید (x_t)، به همراه مقدار حالت Nehan گام قبلی (h_{t-1})، در وزن متناظر خود ضرب و سپس با هم جمع می‌شوند و به یک تابع sigmoid وارد می‌شوند تا خروجی بین بازه ۰ تا ۱ قرار بگیرد. تفاوتی که با گیت بهروزرسانی دارد این است که وزن‌هایی که مقدار ورودی و حالت Nehan گام قبلی در آن ضرب می‌شوند متفاوت است و این یعنی بردارهای خروجی در اینجا با بردار خروجی که در گیت بهروزرسانی داریم متفاوت خواهد بود.



شکل ۱۱ - معماری شبکه Bidirectional LSTM

Bi-LSTM هم نوعی از RNN ها است که دنباله ورودی‌ها را در هر دو جهت forward و backward پردازش می‌کند. هدف اصلی استفاده از چنین معماری‌ای دریافت اطلاعات مرتبط با context هم از element های عقب‌تر و هم از element های جلوتر در دنباله ورودی است. در واقع در این مدل forward pass، backward pass، forward pass Concatenation و backward pass در فاز عمل می‌کنیم. در فاز backward pass، مدل دنباله ورودی را به صورت برگشتی و از آخر به اول پردازش می‌کند. این کار باعث می‌شود که مدل از context های جلوتر در دنباله هم ممکن است برایش مفید باشد استفاده کند. در نهایت در فاز concatenation، خروجی هر دو بخش forward و backward pass به نحوی با هم ترکیب یا concatenate می‌شوند تا نمایش نهایی خروجی در هر مرحله ساخته شود. این طراحی مدل و استفاده از element های قبلی و بعدی باعث می‌شود که عملکرد مدل بر روی داده‌های sequential speech recognition و nlp و تسک‌های تحلیل سری زمانی بهبود یابد.

برای مقایسه مدل‌ها از MAE و MSE و MAPE استفاده شده است.

معیار اندازه گیری میانگین مطلق خطا، برابر است با میانگین تفاوت بین مقدار واقعی و مقدار پیش‌بینی شده بر روی تمام نمونه‌های آموزش است. هر چه این مقدار کمتر باشد، یعنی مدل عملکرد بهتری دارد.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

MSE •

این معیار میانگین مربعات خطاهای است یعنی میانگین مجذور اختلاف بین مقادیر برآورد شده و مقدار واقعی. با گرفتن تفاوت بین مقادیر پیش‌بینی شده و واقعی، مجذور کردن این تفاوت، و سپس میانگین‌گیری این تفاوت‌های مربع در مجموعه داده محاسبه می‌شود. پس این شاخص که مقداری همواره نامنفی دارد، هرچقدر مقدار آن به صفر نزدیکتر باشد، نشان دهنده میزان کمتر خطا و عملکرد بهتر مدل است.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

MAPE •

میانگین درصد مطلق خطا، میانگین بزرگی خطای تولید شده توسط یک مدل یا اینکه پیش‌بینی‌ها به طور متوسط چقدر دور و پرت هستند را اندازه گیری می‌کند. برای مثال اگر مقدار MAPE برابر با ۲۰ درصد باشد به این معنی است که میانگین درصد مطلق اختلاف بین پیش‌بینی‌ها و واقعیات ۲۰٪ است. به عبارت دیگر، پیش‌بینی‌های مدل به طور متوسط ۲۰ درصد از مقادیر واقعی کمتر است. مقدار MAPE کمتر نشان‌دهنده پیش‌بینی دقیق‌تر است. زمانی که MAPE برابر با ۰٪ به این معنی است که پیش‌بینی با مقدار واقعی یکسان است. بطور کلی هر چه مقدار MAPE کمتر باشد یعنی پیش‌بینی دقیق‌تر است.

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

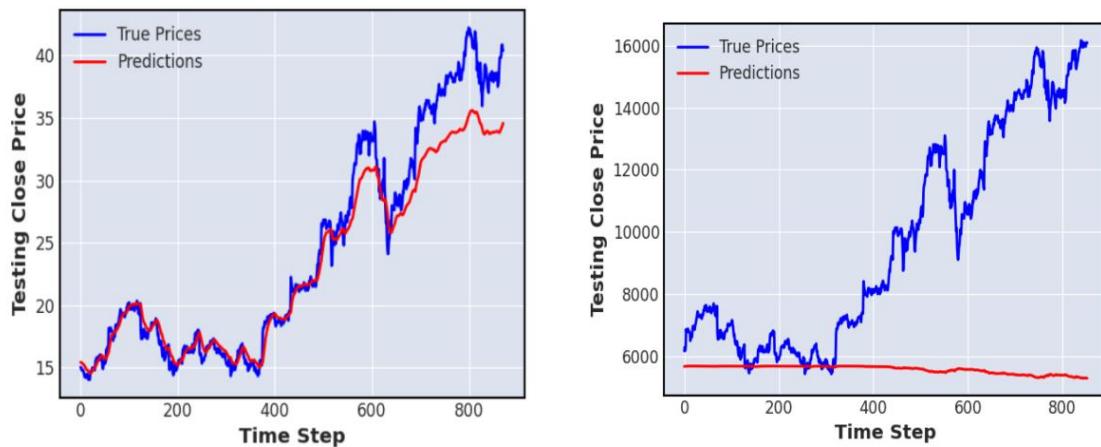
در ادامه نتایج بدست آمده از مدل‌ها را برای سهم‌های مختلف بررسی می‌کنیم.

ابتدا مدل LSTM یک بار برای داده‌های اسکیل شده و یک بار برای داده‌های اسکیل نشده آموزش داده می‌شود تا تاثیر اسکیل کردن داده برای این مدل مشخص شود.

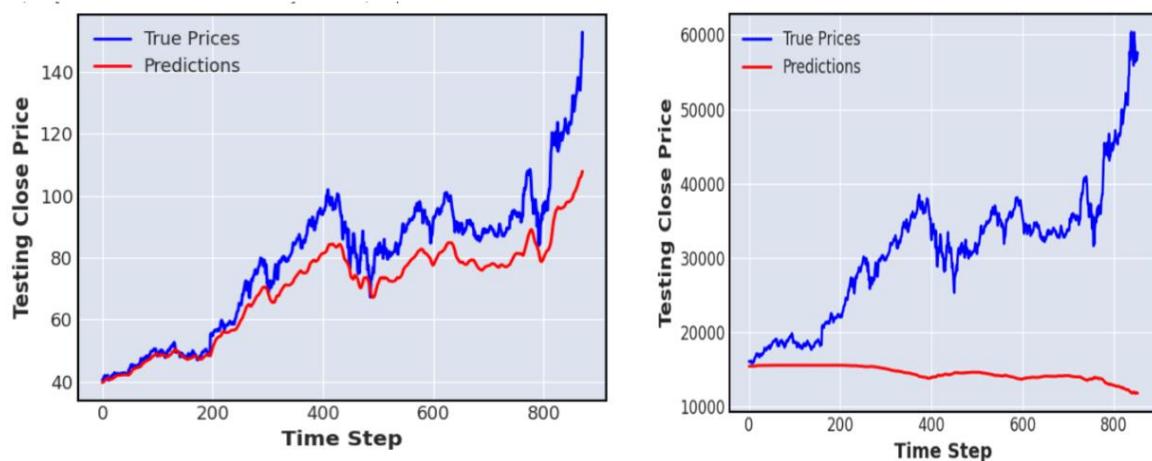
همانطور که از نتایج زیر قابل مشاهده است، مدل زمانی که بر روی داده اسکیل نشده آموزش می‌بیند، در تست در تقسیم‌های مختلف و seed‌های مختلف به نتایج بسیار ضعیفی می‌رسد و تقریباً هیچ بخشی از ترند افزایشی قیمت‌ها را به درستی پیش‌بینی نمی‌کند. اما زمانی که مدل بر روی داده‌های اسکیل شده

آموزش داده می شود، مدل می تواند حتی در split های اولیه نیز تا حد خوبی داده تست را به درستی پیش بینی کند و ترند افزایشی داشته باشد.

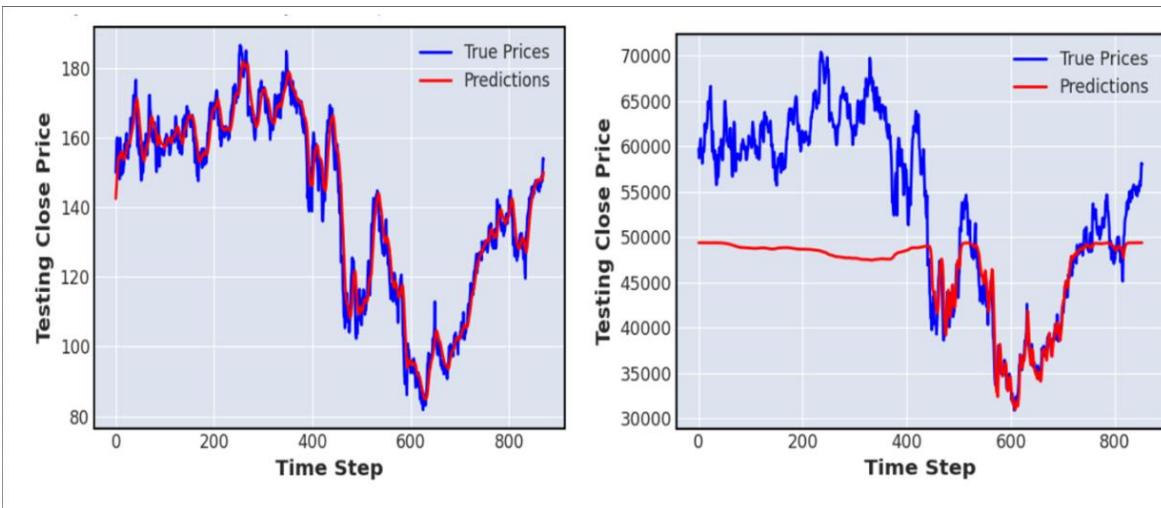
شکل های زیر به ترتیب نشان دهنده عملکرد مدل LSTM برای داده های اسکیل نشده در سمت راست و داده های اسکیل شده در سمت چپ برای سهم ها در split های ۱، ۲ و ۳ با seed برابر با مقدار ۰ است.



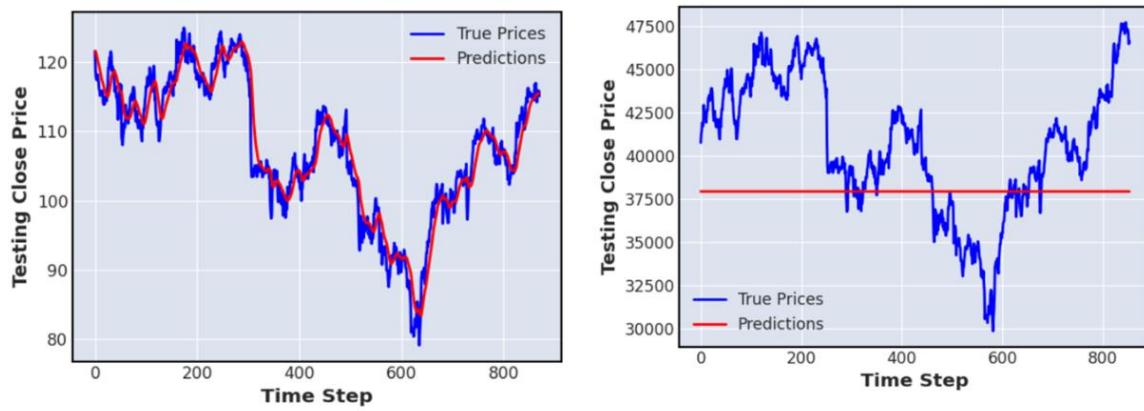
شکل ۱۲ - پیش بینی سهم AMZN در split برابر با ۱ و seed برابر با ۰



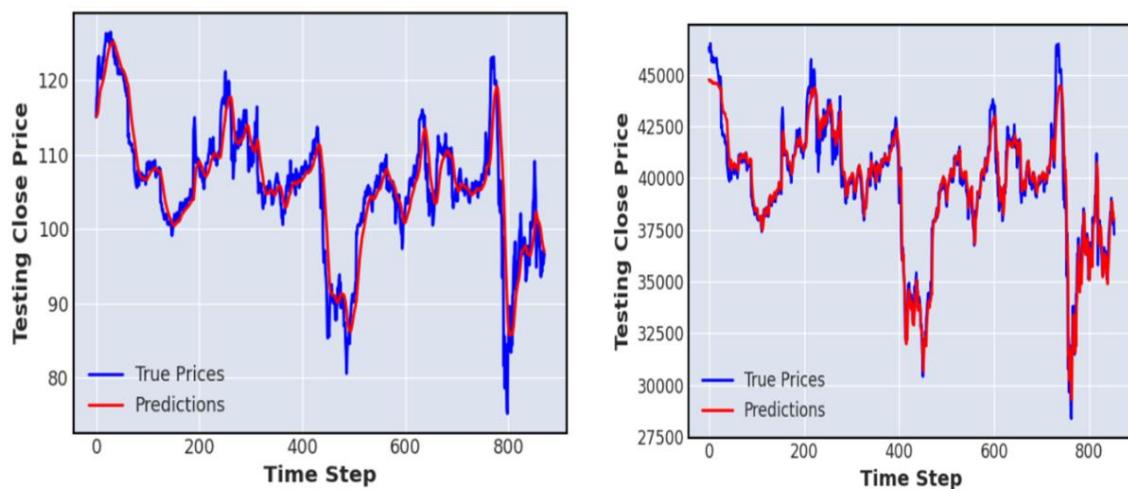
شکل ۱۳ - پیش بینی سهم AMZN در split برابر با ۲ و seed برابر با ۰



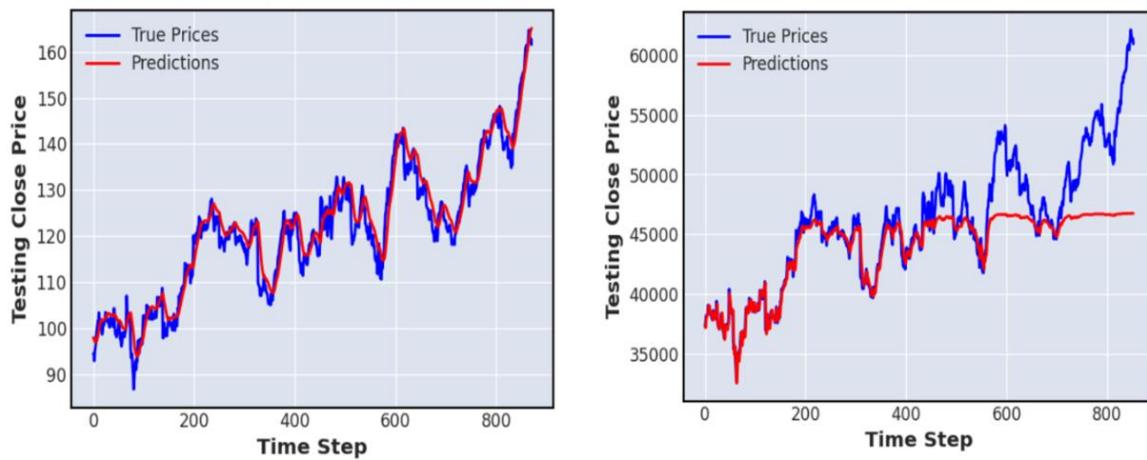
شکل ۱۴ - پیش بینی سهم **AMZN** در split برابر با ۳ و seed برابر یا



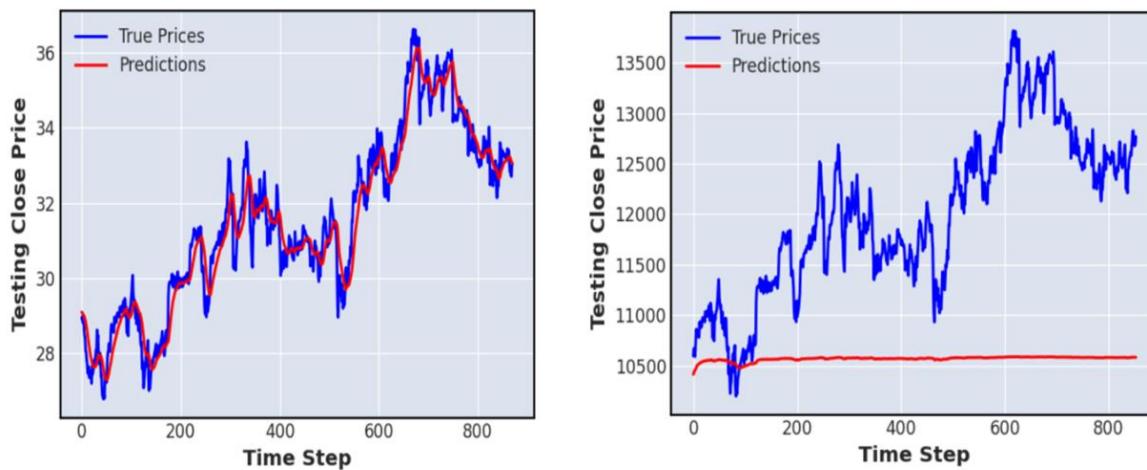
شکل ۱۵ - پیش بینی سهم **IBM** در split برابر با ۱ و seed برابر یا



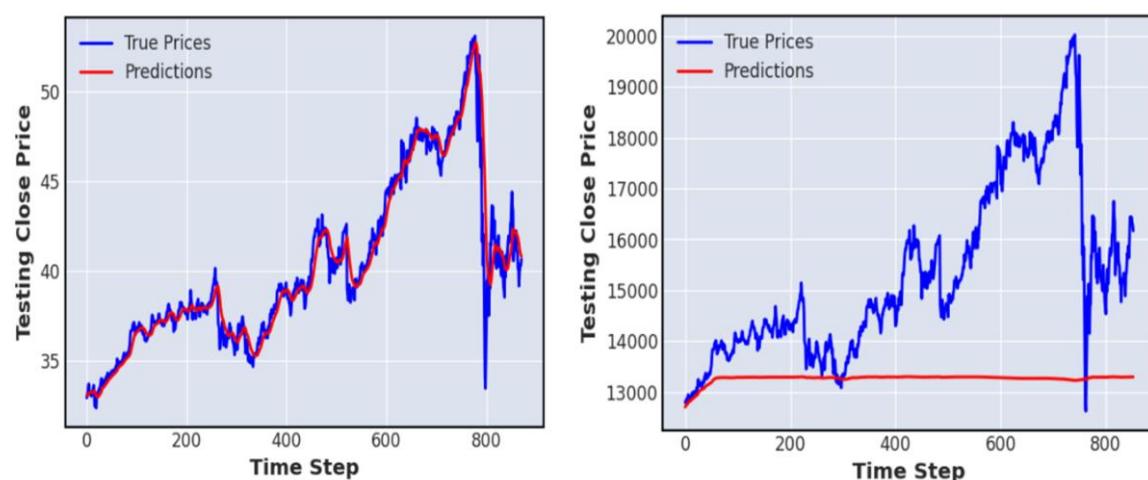
شکل ۱۶ - پیش بینی سهم **IBM** در split برابر با ۲ و seed برابر یا



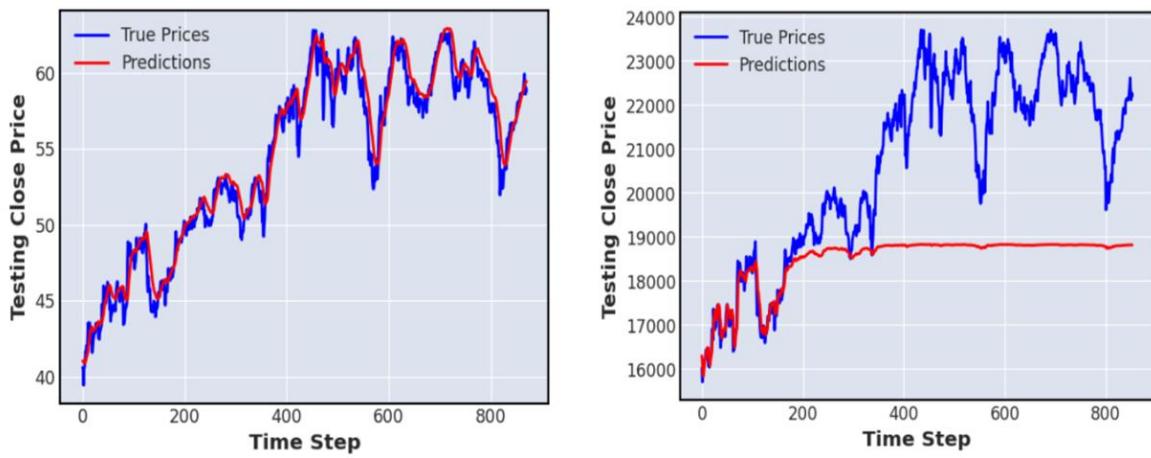
شکل ۱۷ - پیش بینی سهم IBM در split seed برابر با ۳ و برابر با ۰



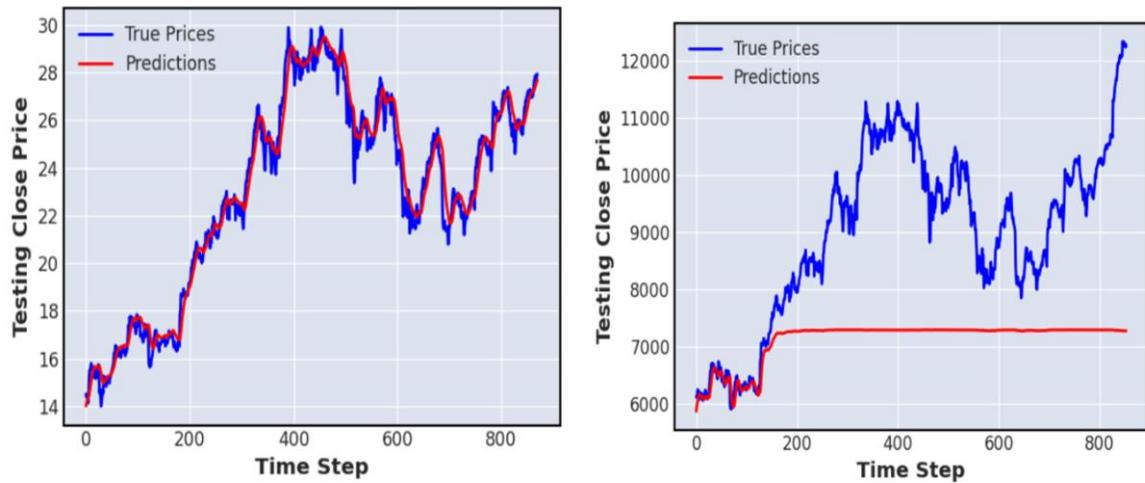
شکل ۱۸ - پیش بینی سهم KO در split seed برابر با ۱ و برابر با ۰



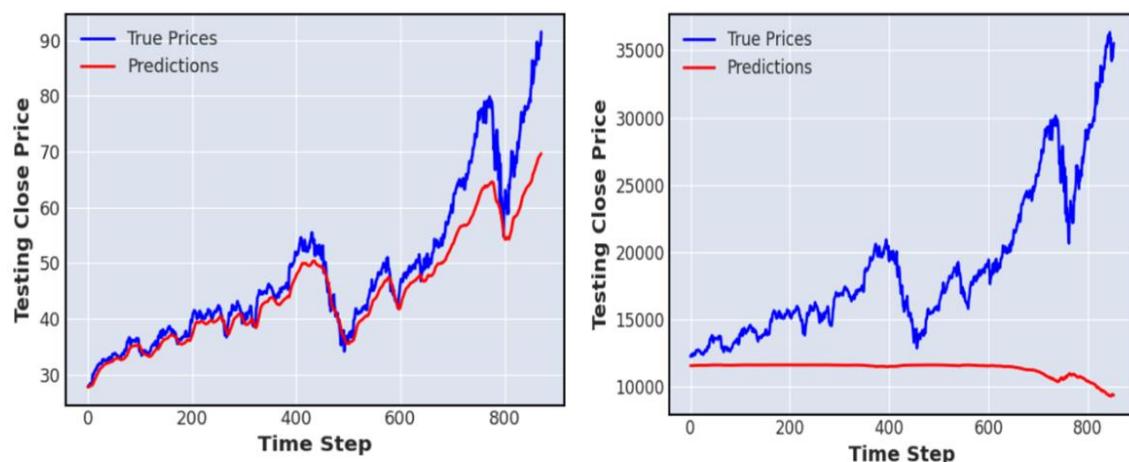
شکل ۱۹ - پیش بینی سهم KO در split seed برابر با ۲ و برابر با ۱



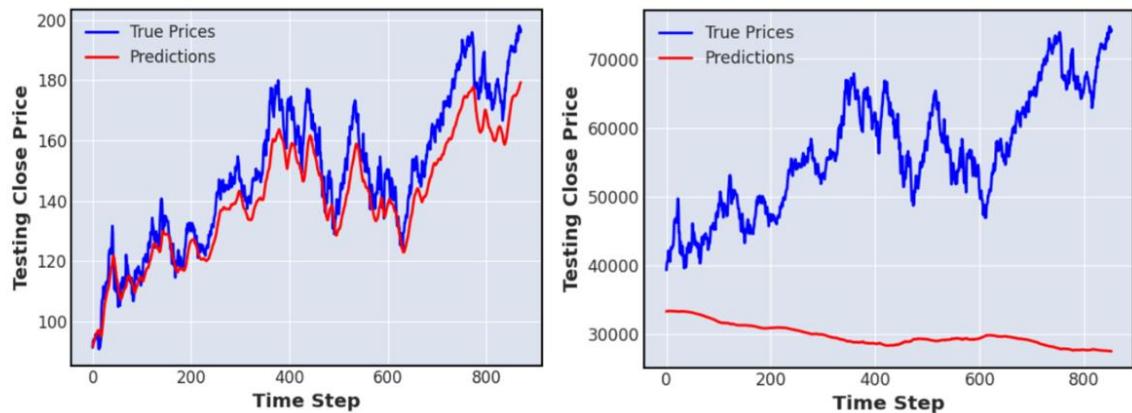
شکل ۲۰ - پیش بینی سهم **KO** در **split** برابر با ۳ و **seed** برابر يا .



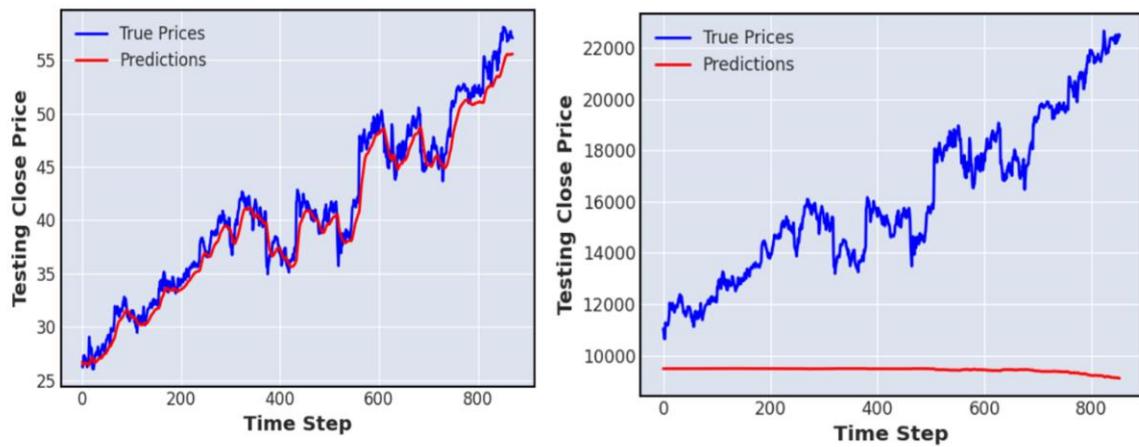
شکل ۲۱ - پیش بینی سهم **AAPL** در **split** برابر با ۱ و **seed** برابر يا .



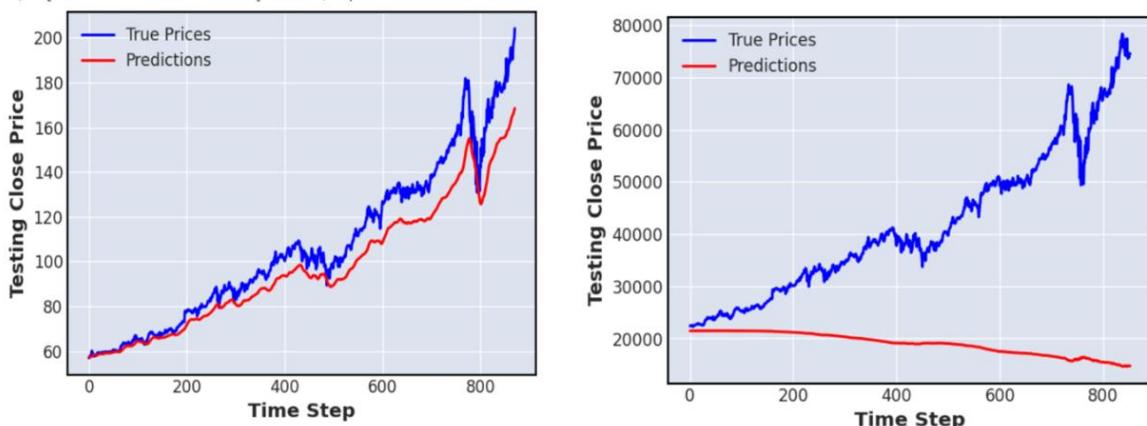
شکل ۲۲ - پیش بینی سهم **AAPL** در **split** برابر با ۲ و **seed** برابر يا .



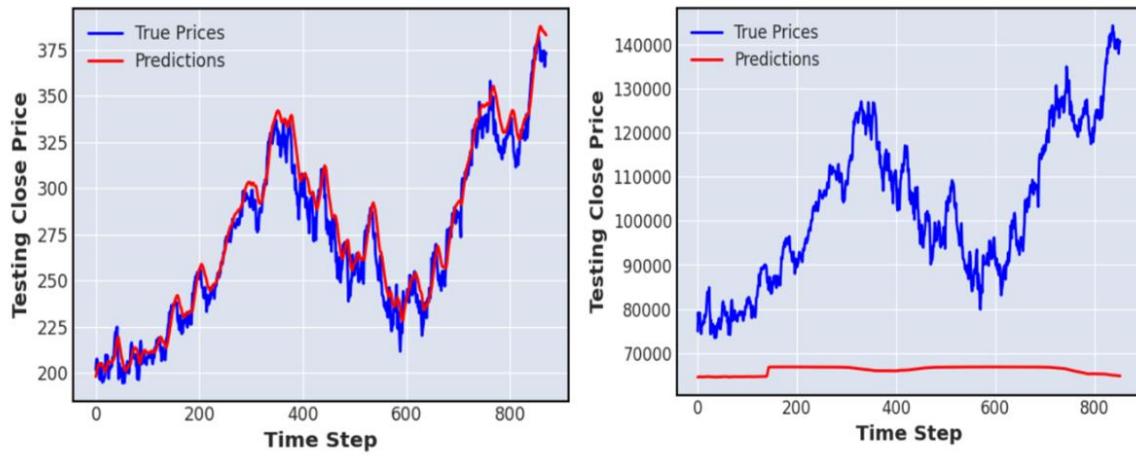
شکل ۲۳ - پیش بینی سهم AAPL در split برابر با ۳ و seed برابر يا .



شکل ۲۴ - پیش بینی سهم MFST در split برابر با ۱ و seed برابر يا .



شکل ۲۵ - پیش بینی سهم MFST در split برابر با ۲ و seed برابر يا .



شکل ۲۶ - پیش بینی سهم MFST در split برابر با ۳ و seed برابر با

نتایج معیارهای معرفی شده برای مدل‌های مختلف و سهام‌های مختلف را می‌توانید در شکل‌های زیر مشاهده کنید.



شکل ۲۷ - نتایج معیارها برای سهام AMZN

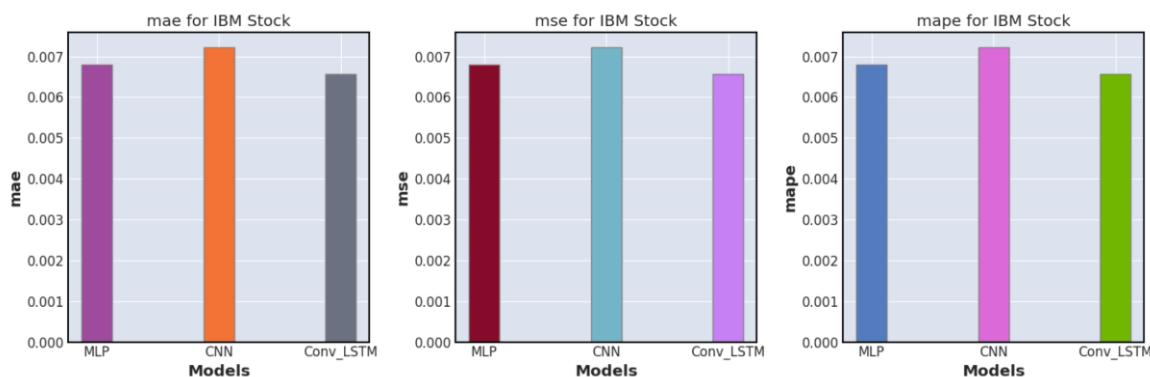


شکل ۲۸ - نتایج معیارها برای سهام AMZN

همانطور که مشاهده می‌شود، برای سهام AMZN مدل GRU با اختلاف کمی بهترین عملکرد را دارد و مدل LSTM هم به خاطر سادگی آن عملکرد مناسبی نسبت به باقی مدل‌ها ندارد.

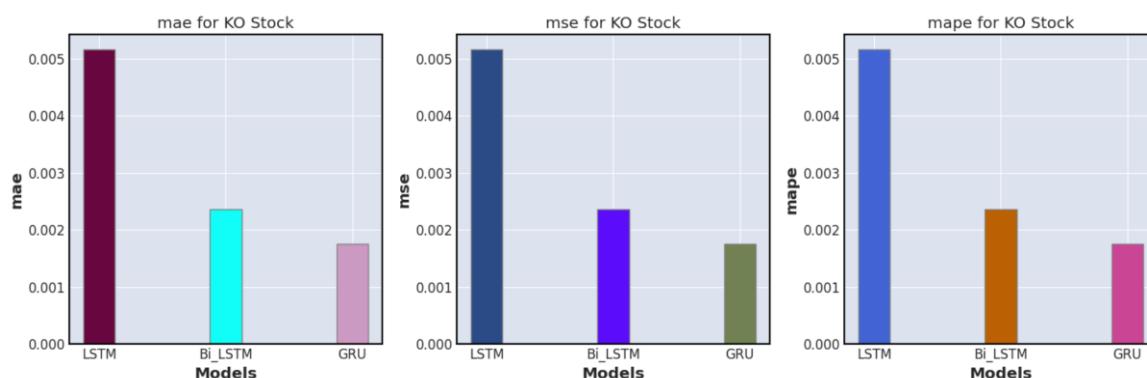


شکل ۲۹ - نتایج معیارها برای سهام IBM

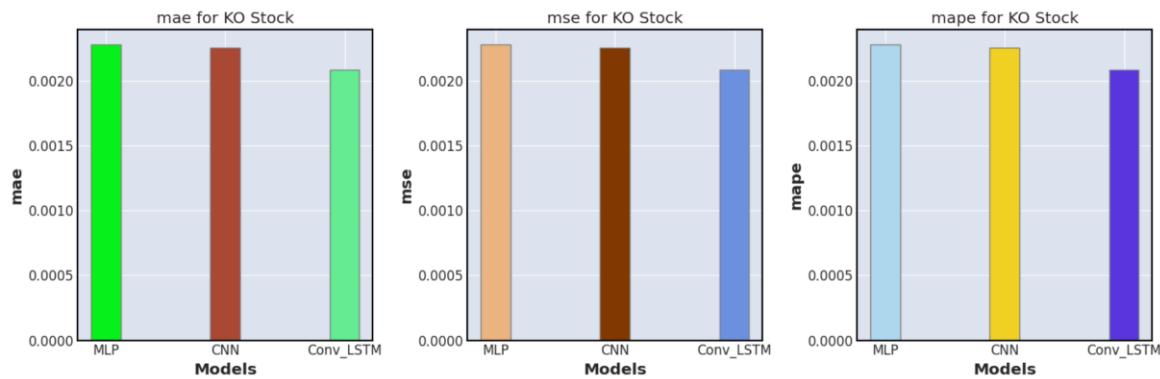


شکل ۳۰ - نتایج معیارها برای سهام IBM

برای سهام IBM هم مدل GRU با اختلاف کمی بهترین عملکرد را ارائه می‌دهد و مدل LSTM هم عملکرد مطلوبی برای ما ندارد.

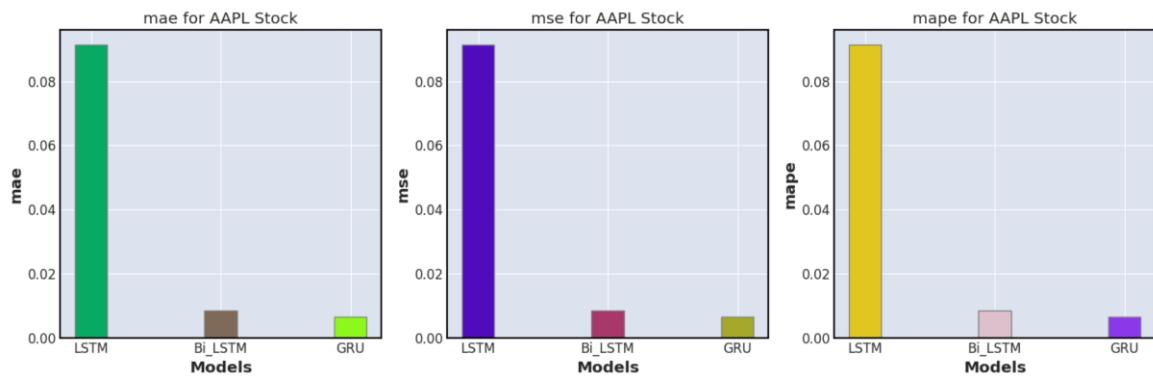


شکل ۳۱ - نتایج معیارها برای سهام KO



شکل ۳۲ - نتایج معیارها برای سهام KO

در سهام KO هم مشابه دو سهام قبلی، عملکرد مدل GRU با اختلاف کمی نسبت به سایر مدل‌ها بهتر و قابل قبول‌تر است.

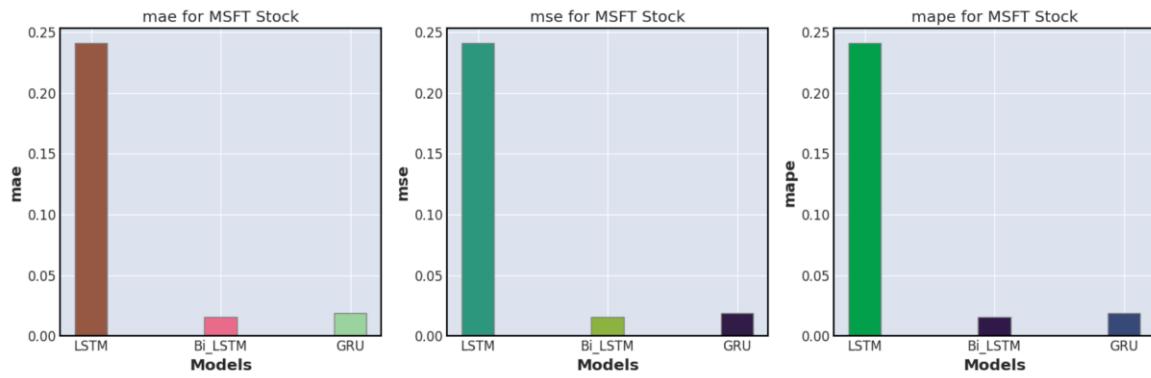


شکل ۳۳ - نتایج معیارها برای سهام AAPL

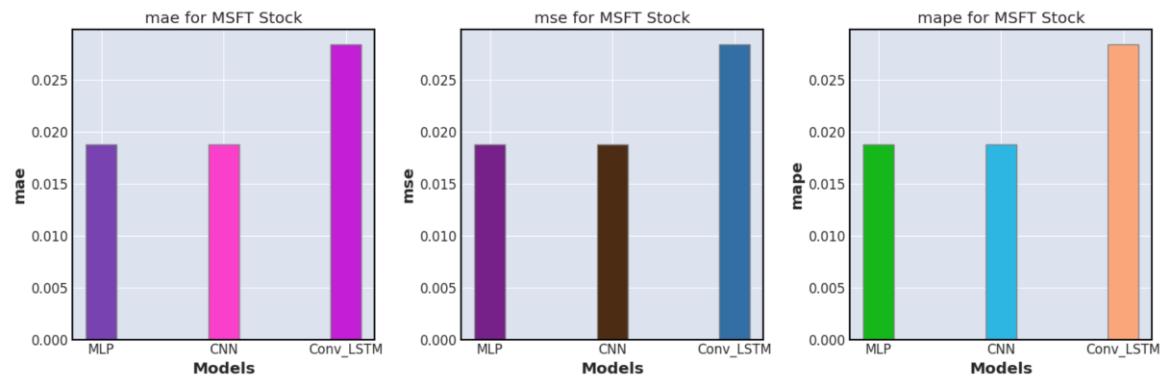


شکل ۳۴ - نتایج معیارها برای سهام AAPL

نتایج برای سهام AAPL نیز مطابق انتظار نشان از عملکرد بهتر مدل GRU دارد.



شکل ۳۵ - نتایج معیارها برای سهام MSFT



شکل ۳۶ - نتایج معیارها برای سهام MSFT

برای سهام MSFT اما مدل Bi-LSTM با اختلاف کمی عملکرد بهتری نسبت به باقی مدل‌ها ارائه می‌دهد. همچنین برای این سهام اختلاف مدل Conv-LSTM با باقی مدل‌ها بیشتر می‌شود و این مدل به خوبی برای این مدل عمل نمی‌کند.

همانطور که در نتایج هم دیده می‌شود، مدل LSTM به طور کلی نسبت به باقی مدل‌ها عملکرد ضعیفتری ارائه می‌دهد. اما باقی مدل‌ها عملکرد نزدیک به همی دارند و در بین آن‌ها نیز در اکثر موارد، مدل GRU با اختلاف کمی عملکرد بهتری از خود بر روی سهام‌ها به نمایش می‌گذارد.

۶-۱ Naïve Forecast

مقادیر زیر نشان دهنده معیارهای بدست آمده برای پیاده سازی روش random walk بر روی داده‌های بورس است.

	AMZN	IBM	KO	AAPL	MSFT
MAE	1.38	1.31	0.88	1.22	1.74
MSE	4.21	3.17	1.24	3.04	7.97
MAPE	4.16%	1.23%	2.60%	3.76%	2.23%

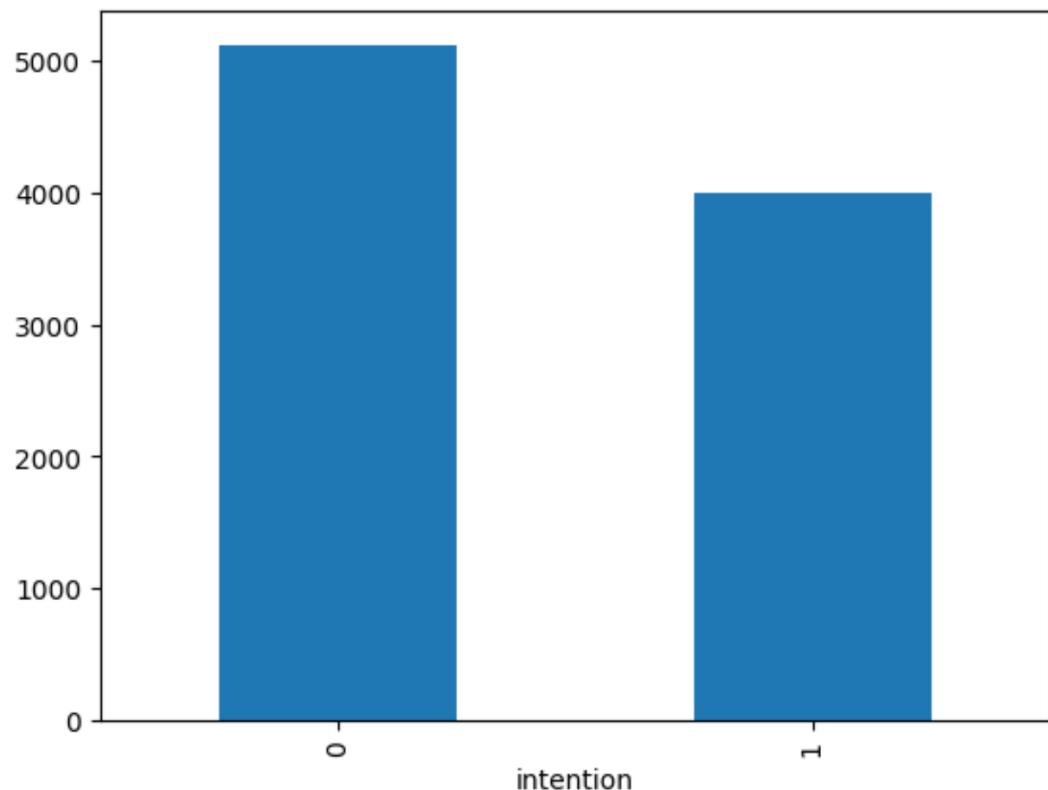
شکل ۳۷ - نتایج معیارهای مختلف بر روی سهام‌ها برای مدل **Random Walk**

همانطور که در نتایج پیداست عملکرد مدل Random Walk نسبت به مدل‌های پیاده‌سازی شده در بخش قبل و به خصوص مدل GRU، عملکرد خوبی از خود نشان نمی‌دهد. این نتیجه مطابق انتظار ما هم بوده است و از این مدل عموماً به عنوان مدل baseline برای تحلیل سری‌های زمانی استفاده می‌شود.

از دیگر نکات قابل توجه درباره نتایج این مدل می‌توان به این نکته اشاره کرد که نتایج این مدل خیلی بیشتر از مدل‌های قبلی وابسته به دیتاست ما و سهامی است که در حال بررسی آن هستیم و با تغییر آن عملکرد مدل هم تا حد زیادی تغییر می‌کند.

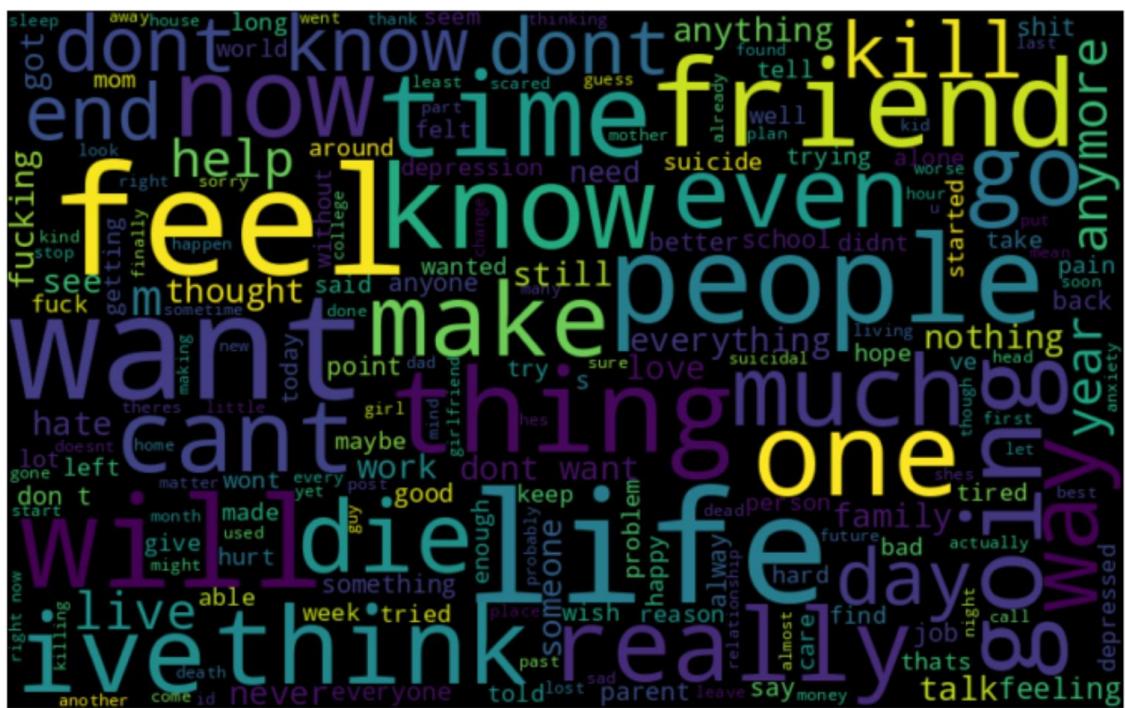
پاسخ ۲ - پیش‌بینی افکار خودکشی در رسانه‌های اجتماعی

۱-۲. پیش‌پردازش داده



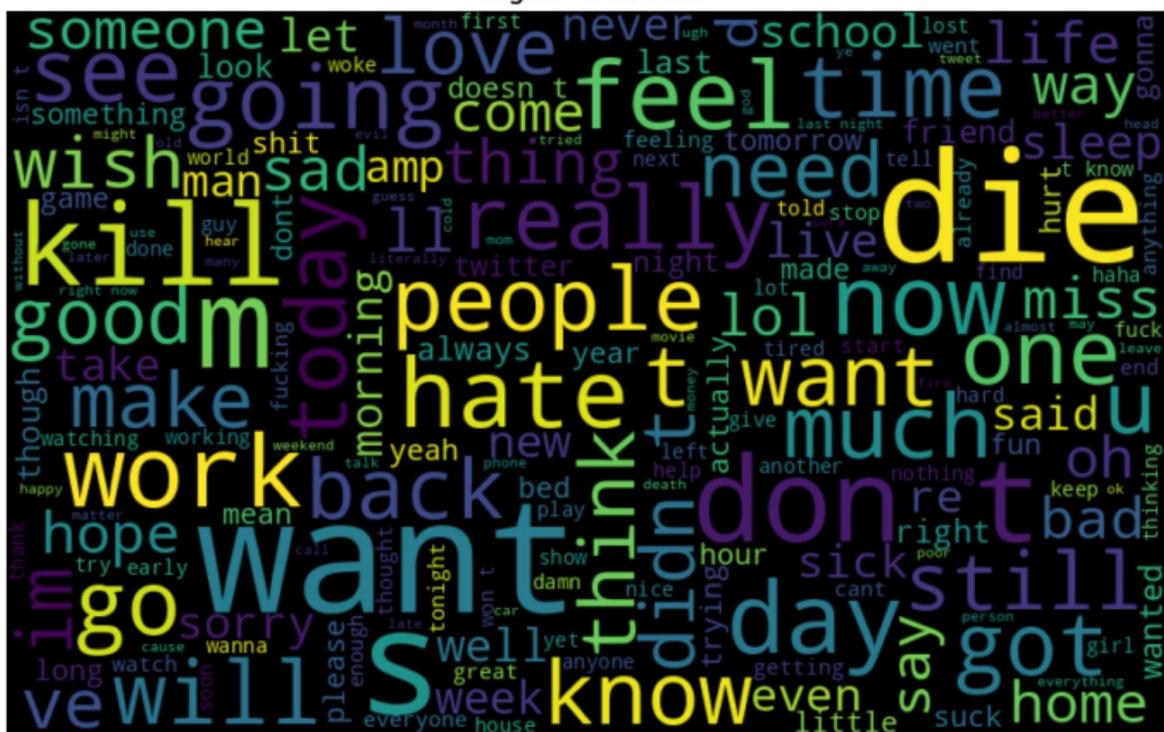
شكل ۳۸ - توزیع کلاس‌ها

Positive tweets



شکل - ۳۹ non-suicidal word cloud برای توپیت‌های

Negative tweets

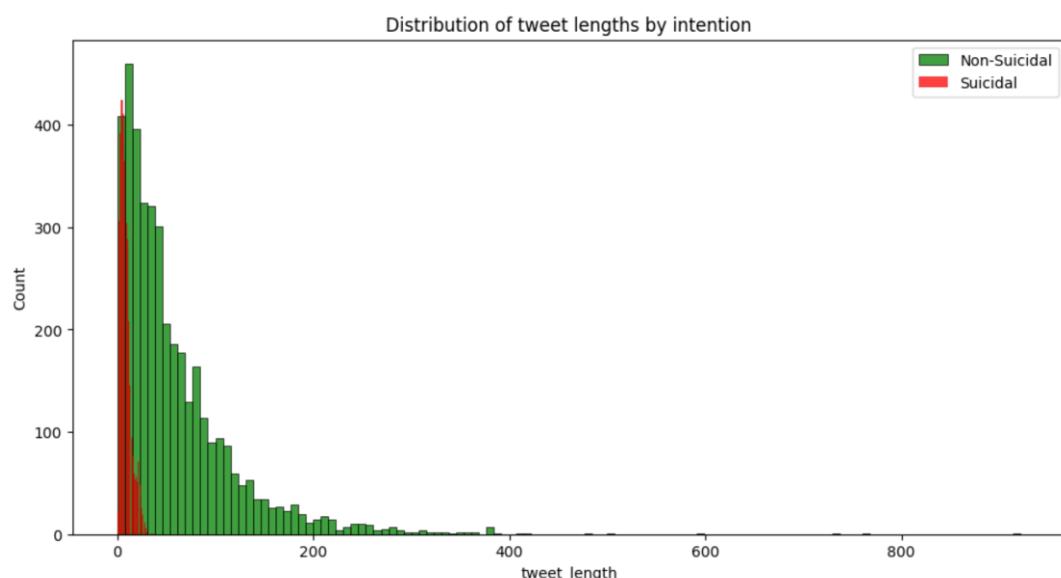


شکل ۴۰ - suicidal word cloud پرای توپیت‌های

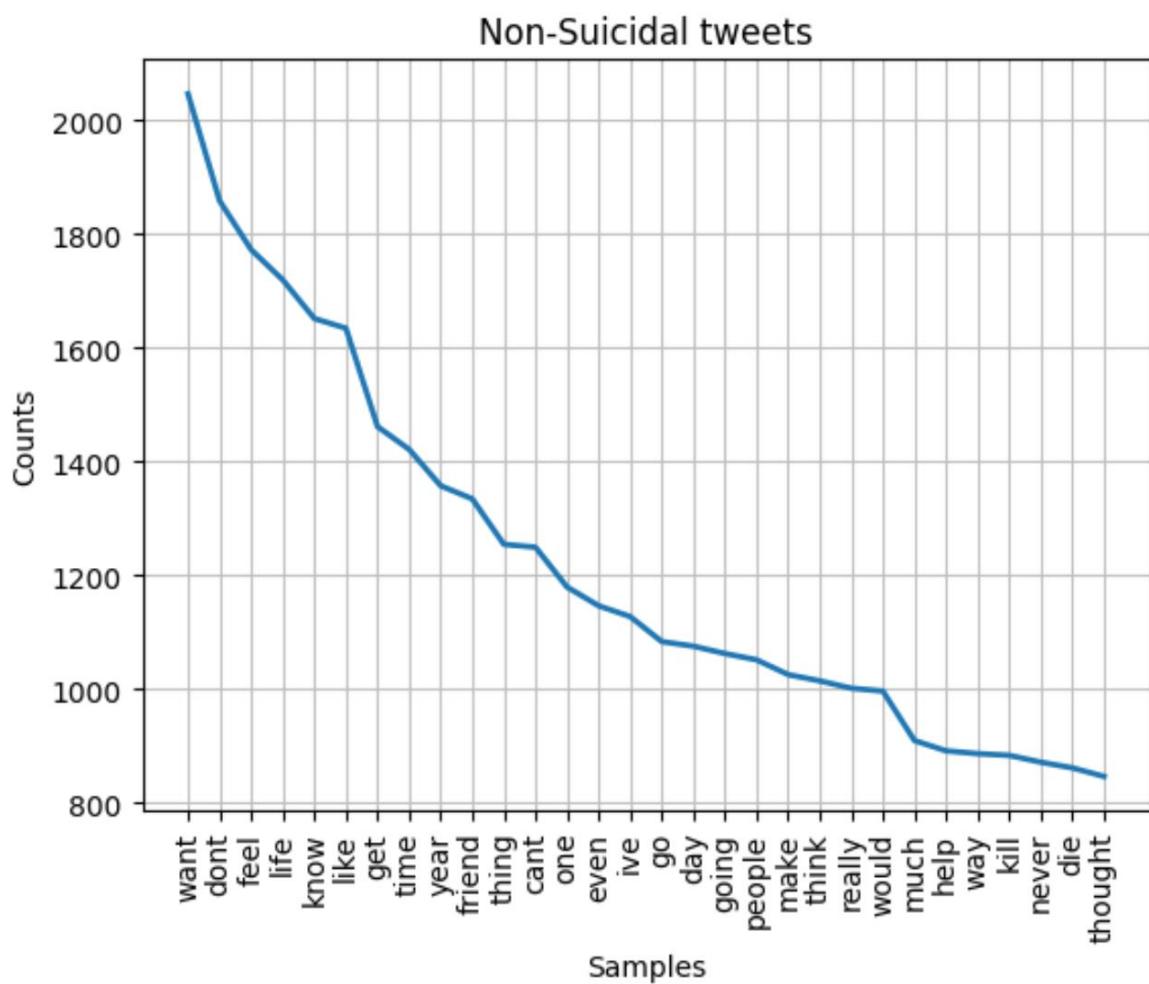
در این مرحله پیش‌پردازش‌های لازم مثل حذف رکوردهای تکراری از دیتاست، حذف mention‌ها، hashtags، لینک‌ها، کarakترهای غیر‌حروف و whitespace‌های اضافی، تبدیل اموجی‌ها به متن متناظر، حذف punctuation‌ها و تبدیل کarakترها به lower case انجام شد. سپس با استفاده از کتابخانه NLTK عملیات حذف stop word و lemmatization روی توییت‌ها اعمال شد.

```
Original tweet: my life is meaningless i just want to end my life so badly my life is completely empty and i dont want to have to create meaning in it creating meaning is pain how long will i hold back the urge to run my car head first into the next person coming the opposite way when will i stop feeling jealous of tragic characters like gomer pile for the swift end they were able to bring to their lives
Cleaned tweet: life meaningless want end badly completely empty dont create meaning creating pain long hold back urge run car head first next person coming opposite way stop feeling jealous tragic character like gomer pile swift able bring life
```

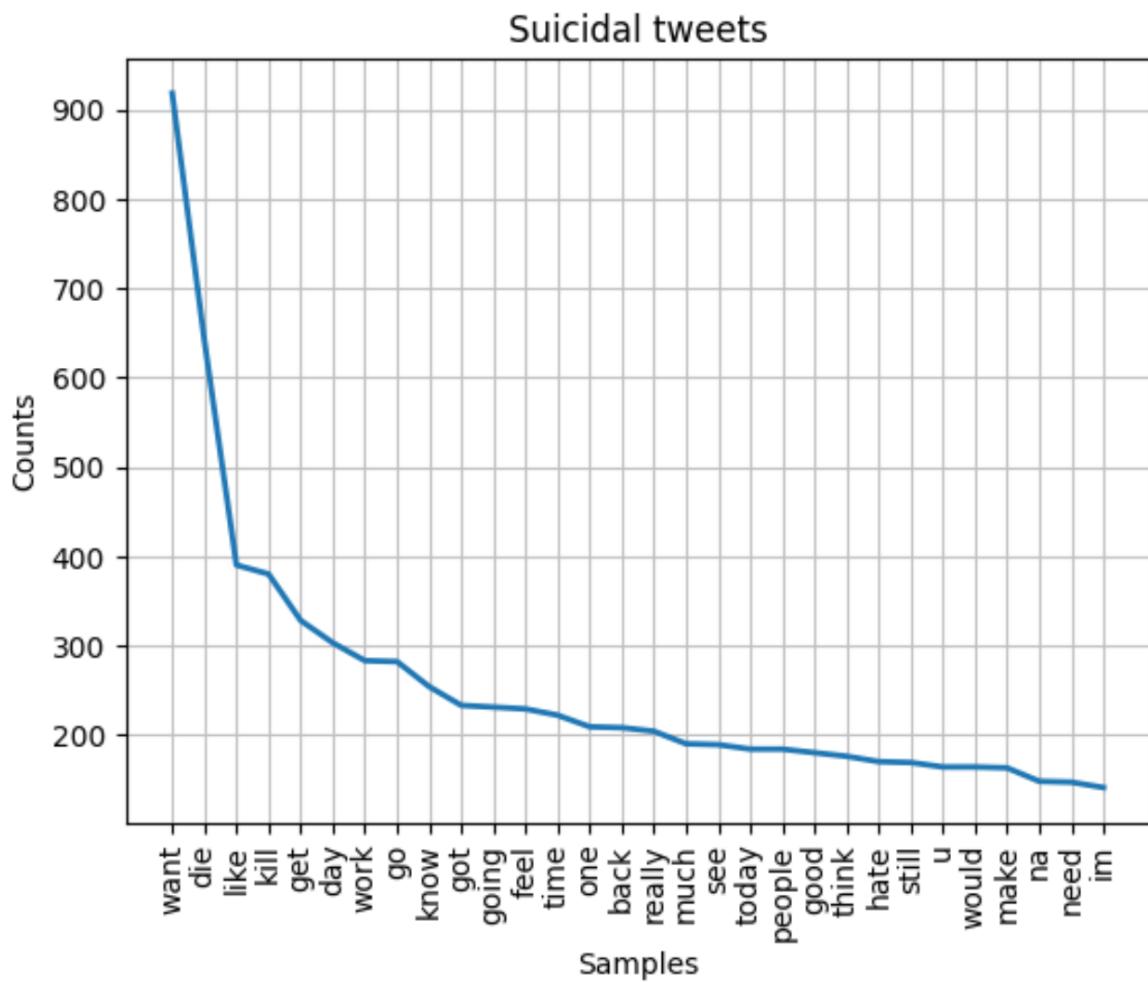
شکل ۴۱ - نمونه توییت قبل و بعد از پیش‌پردازش



شکل ۴۲ - نمودار توزیع طول توییت



شکل ۴۳ - نمودار توزیع کلمات در توییت‌های **non-suicidal**



شکل ۴۴ - نمودار توزیع کلمات در توییت‌های **suicidal**

۲-۲. ساخت ماتریس جاسازی

این مدل یک مدل pre-trained word embedding است که توسط Google توسعه داده شده است و تلاش می‌کند که شباهت‌های semantic بین کلمات در متون را برحسب co-occurrence آن‌ها در متن پیدا کرده و کلمات را به صورت برداری از اعداد represent کند. دو عنصر اصلی معماری این مدل، skip-gram و Continuous Bag of Words (CBOW) هستند. این مدل به طور خاص با استفاده از الگوریتم Word2Vec و بر روی دیتابیس متشکل از Google News آموزش داده شده است. در این مدل هر کلمه به یک بردار ویژگی به سایز ۳۰۰ نگاشت می‌شود. با استفاده از این روش کلماتی که معنای نزدیکی به هم دیگر دارند، دارای بردار ویژگی نزدیک‌تری به هم نیز هستند. همچنین در طول آموزش، این مدل به معنای کلمه در context نیز توجه می‌کند. از ویژگی‌های دیگر این مدل می‌توان به vocabulary size بزرگ آن اشاره کرد. به همین دلایل می‌توان از این مدل برای تسک‌های مختلف NLP، بازیابی اطلاعات، Document

تسکهای Classification و یا تشخیص شباهت متن استفاده کرد. این دلایل مجموعاً باعث می‌شوند که استفاده از مدل word2vec برای این تسک مناسب باشد.

```
word2vec_path = '/kaggle/working/word2vec-google-news-300'
```

```
# download and save the word2vec model
word2vec_model = gensim.downloader.load('word2vec-google-news-300', return_path=False)
```

شکل ۴۵ - لود کردن مدل word2vec

۳-۲. آموزش مدل‌های یادگیری عمیق

```
def train_tweet_classifier(model, train_dataloader, test_dataloader, criterion, optimizer, num_epochs=20):
    train_losses = []
    train_accs = []
    test_losses = []
    test_accs = []

    for epoch in range(num_epochs):
        print(f'----- Epoch {epoch+1}/{num_epochs} -----')
        model.train()
        train_loss = 0
        train_acc = 0
        for batch in tqdm.tqdm(train_dataloader):
            optimizer.zero_grad()

            tweet = batch['tweet']
            label = batch['label'].to(device)
            label = label.float()

            # tweet = tweet.to(device)
            # label = label.to(device)

            outputs = model(tweet)
            outputs = outputs.squeeze(1)
            # print(f' shape of outputs: {outputs.shape}\nshape of label: {label.shape}')
            # print(f'outputs: {outputs}\nlabel: {label}')
            loss = criterion(outputs, label)
            loss.backward()
            optimizer.step()

            train_loss += loss.item()
            pred_labels = torch.sigmoid(outputs)
            pred_labels = torch.round(pred_labels)
            train_acc += (pred_labels == label).sum().item()
            # train_acc += (outputs.argmax(1) == label).sum().item()

        train_loss /= len(train_dataloader.dataset)
        train_acc /= len(train_dataloader.dataset)

        train_losses.append(train_loss)
        train_accs.append(train_acc)
```

شکل ۴۶ - حلقه آموزش مدل‌ها

```

model.eval()
test_loss = 0
test_acc = 0
with torch.no_grad():
    for batch in tqdm.tqdm(test_dataloader):
        tweet = batch['tweet']
        label = batch['label'].to(device)
        label = label.float()

        # tweet = tweet.to(device)
        # label = label.to(device)

        outputs = model(tweet)
        outputs = outputs.squeeze(1)
        loss = criterion(outputs, label)

        test_loss += loss.item()
        pred_labels = torch.sigmoid(outputs)
        pred_labels = torch.round(pred_labels)
        test_acc += (pred_labels == label).sum().item()
        # test_acc += (outputs.argmax(1) == label).sum().item()

    test_loss /= len(test_dataloader.dataset)
    test_acc /= len(test_dataloader.dataset)

    test_losses.append(test_loss)
    test_accs.append(test_acc)

    print(f'Epoch {epoch+1}/{num_epochs}:')
    print(f'Training loss: {train_loss:.4f} | Training accuracy: {train_acc:.4f}')
    print(f'Testing loss: {test_loss:.4f} | Testing accuracy: {test_acc:.4f}')
    print()

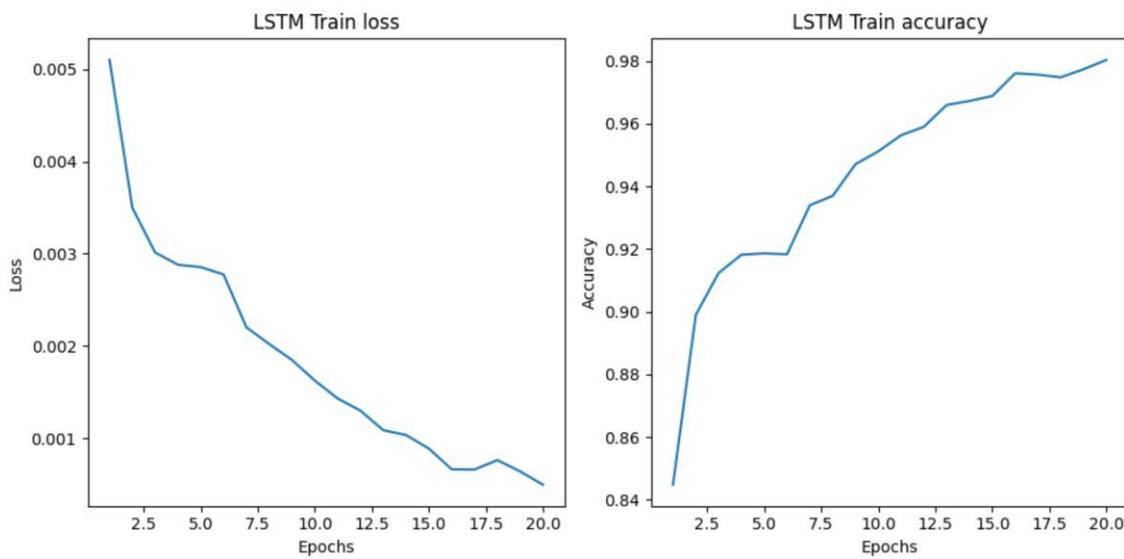
return train_losses, train_accs, test_losses, test_accs

```

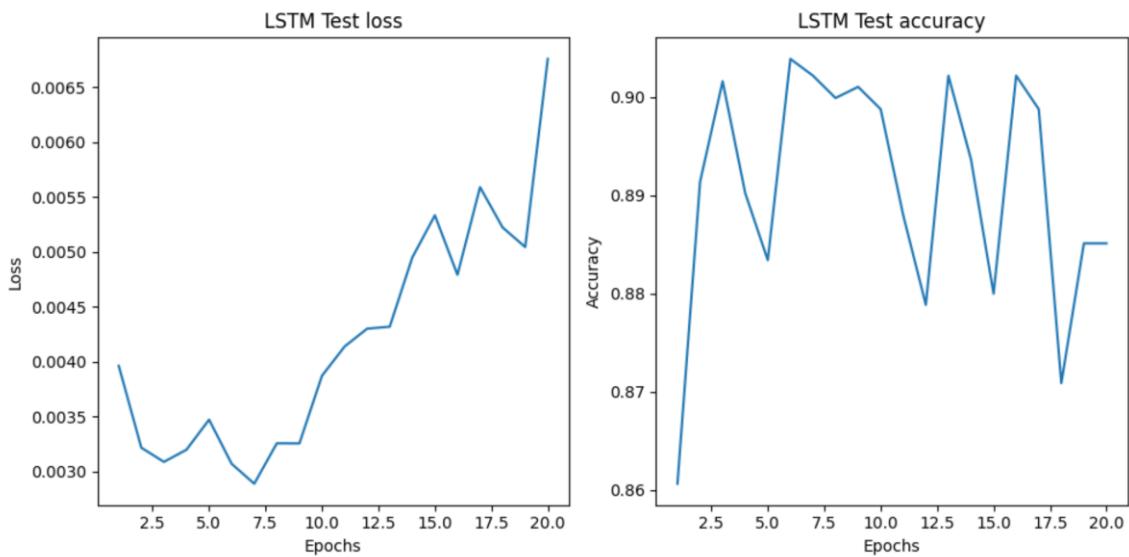
شکل ۴۷ - ارزیابی عملکرد مدل در حین آموزش

۴-۲. مقایسه نتایج

- **LSTM**



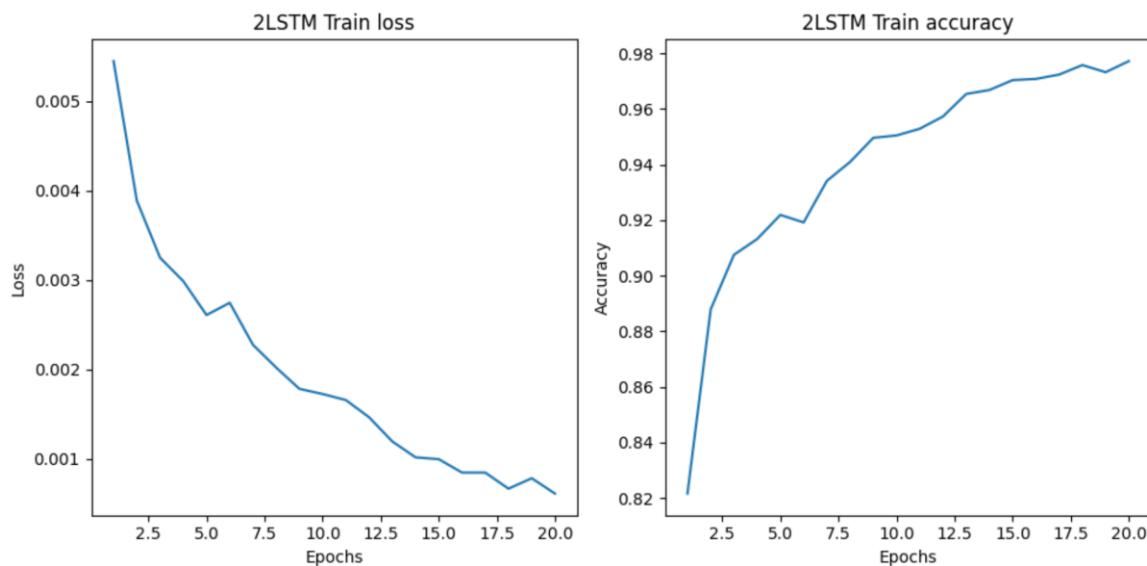
شکل ۴۸ - نمودار LSTM برای loss و accuracy روی داده آموزش



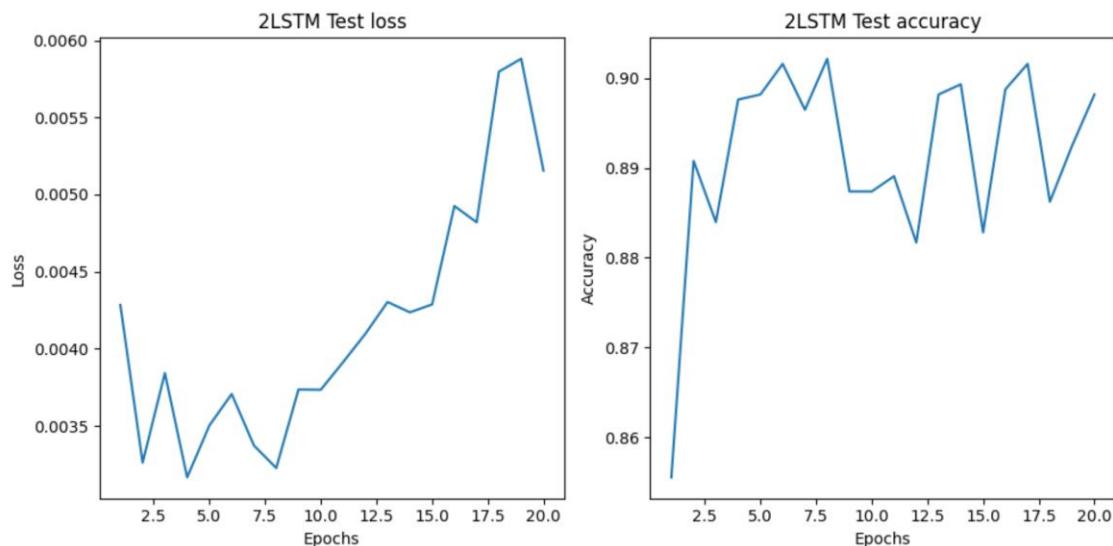
شکل ۴۹ - نمودار loss و accuracy برای LSTM روی داده ارزیابی

همانطور که می‌بینید دقیق نهایی مدل بر روی دادگان ارزیابی ما با استفاده از این مدل به ۸۸۵۱٪ می‌رسد.

- 2 Layer LSTM



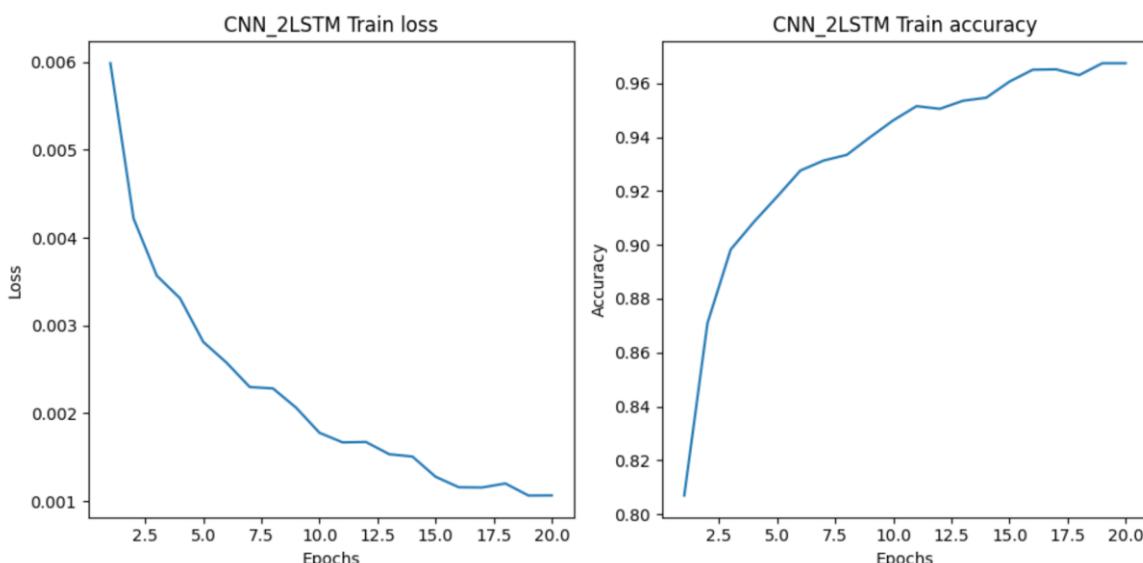
شکل ۵۰ - نمودار loss و accuracy برای 2 Layer LSTM روی داده آموزش



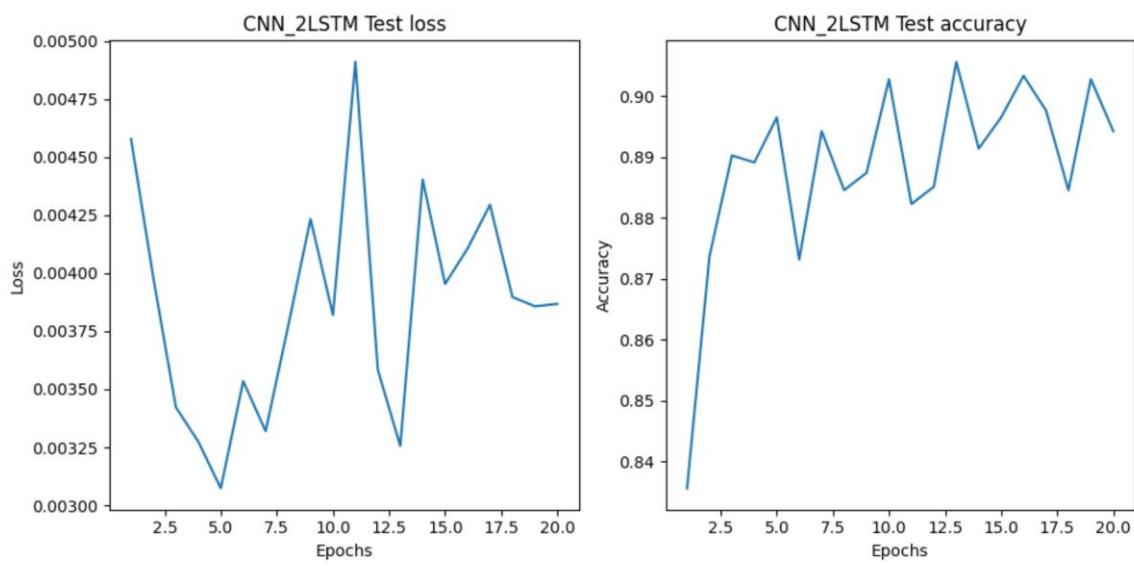
شکل ۵۱ - نمودار loss و accuracy برای ۲ Layer LSTM روی داده ارزیابی

همانطور که می‌بینید دقیق نهایی مدل بر روی دادگان ارزیابی ما با استفاده از این مدل به ۸۹۸۲٪ می‌رسد.

- CNN + 2 Layer LSTM



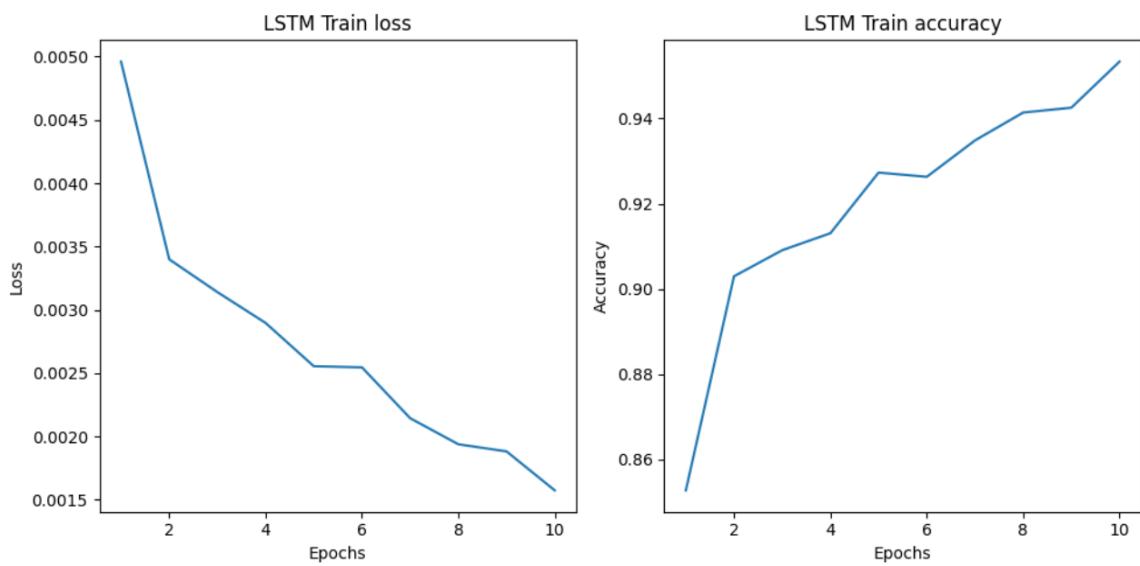
شکل ۵۲ - نمودار loss و accuracy برای CNN + 2 Layer LSTM روی داده آموزش



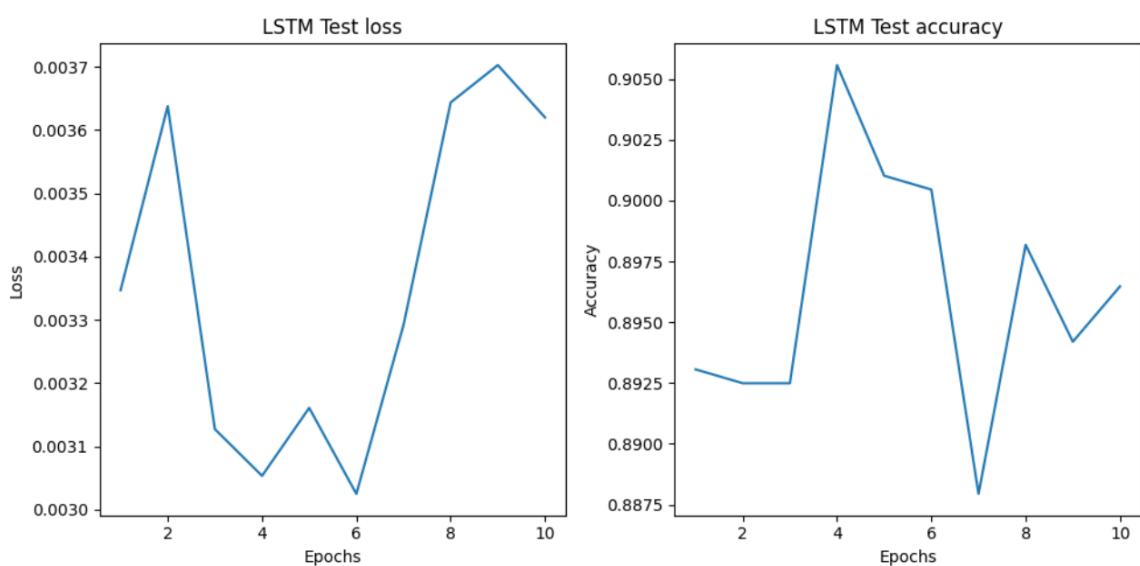
شکل ۵۳ - نمودار loss و accuracy برای CNN + 2 Layer LSTM روی داده ارزیابی

همانطور که می‌بینید دقیق نهایی مدل بر روی دادگان ارزیابی ما با استفاده از این مدل به ۹۰.۵۶٪ می‌رسد.

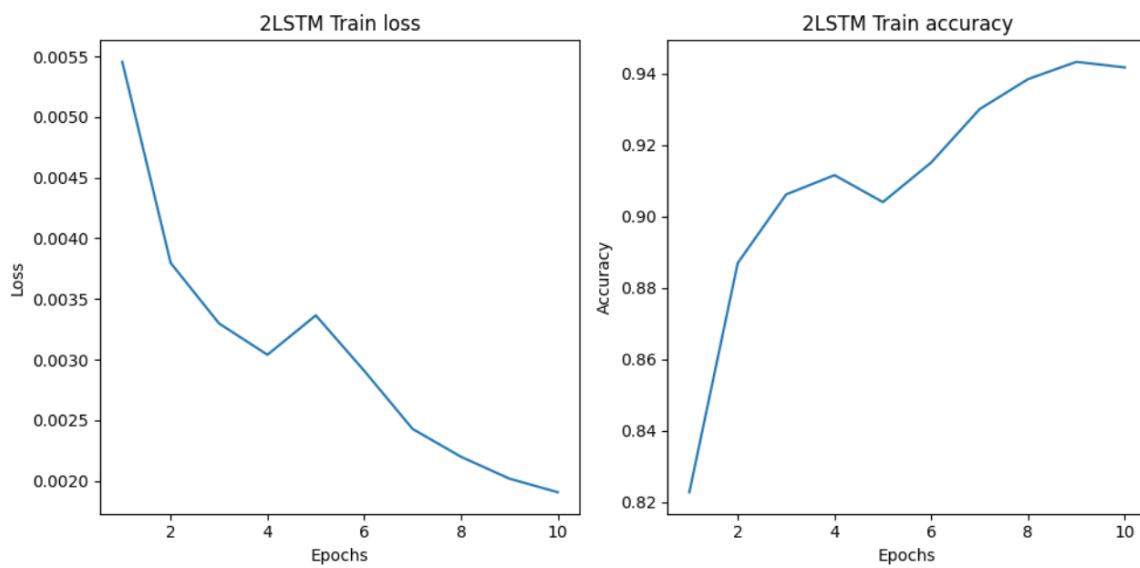
طبق نتایج مشاهده شده، عملکرد مدل‌ها مطابق انتظار ما بوده است و بهبود عملکرد مدل‌ها قابل مشاهده است. اما مشکلی که وجود دارد این است که آموزش مدل‌ها با استفاده از این هایپرپارامترها منجر به overfit شدن مدل‌ها می‌شود. نتایج نهایی با تغییر این مقادیر در شکل‌های زیر قابل مشاهده هستند.



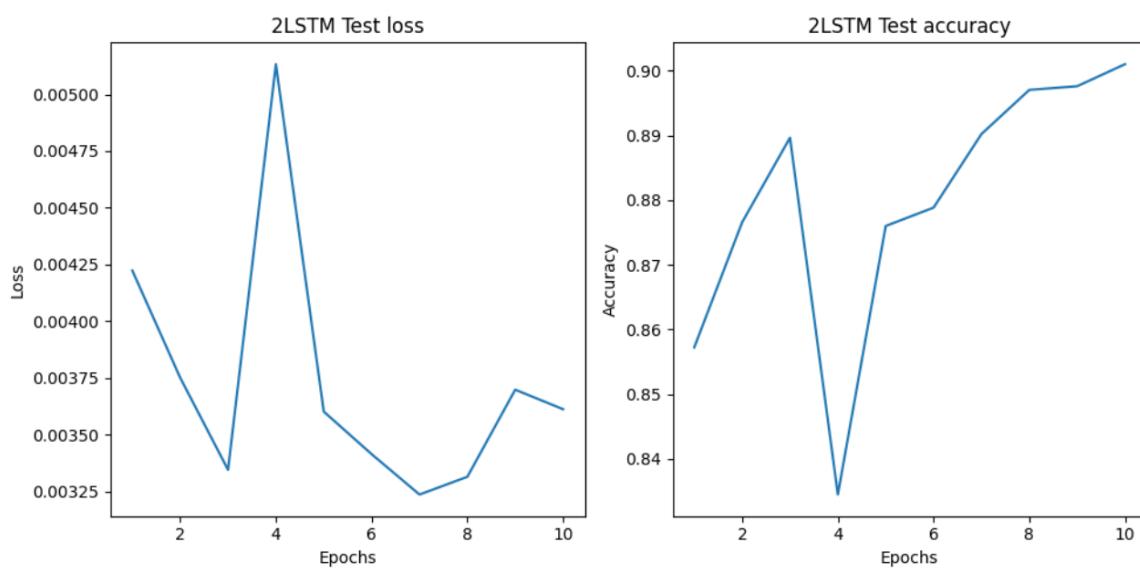
```
plot_training_summary('LSTM Test', test_losses_LSMT, test_accs_LSMT, num_epochs)
```



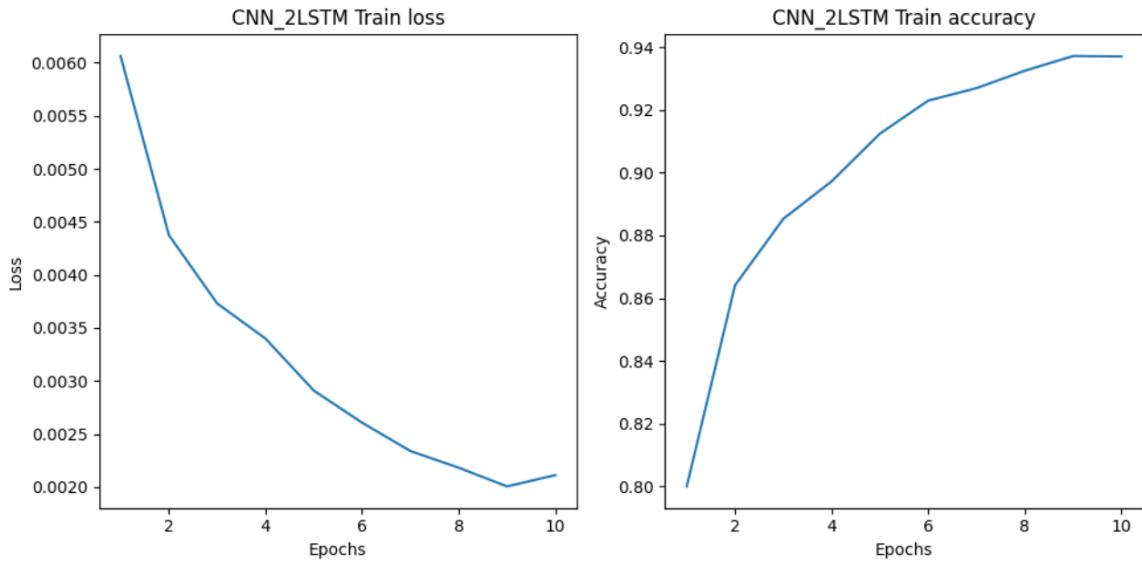
شکل ۵۴ - نمودار loss و accuracy برای LSTM



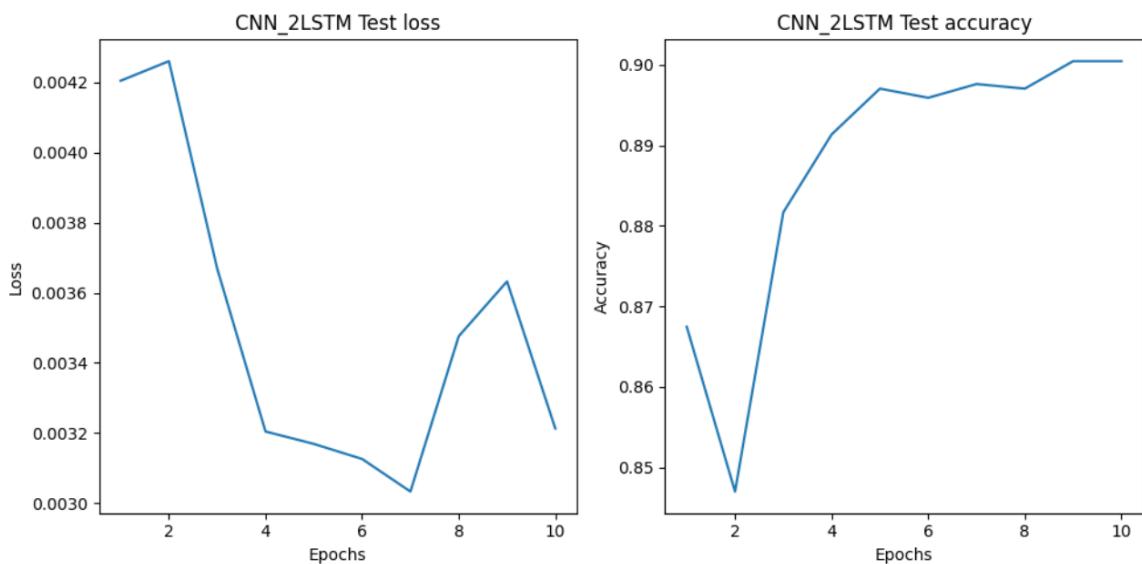
```
plot_training_summary('2LSTM Test', test_losses_2LSMT, test_accs_2LSMT, num_epochs)
```



شکل ۵۵ - نمودار loss و accuracy برای ۲ Layer LSTM



```
plot_training_summary('CNN_2LSTM Test', test_losses_CNN_2LSMT, test_accs_CNN_2LSMT, num_epochs)
```



شکل ۵۶ - نمودار loss و accuracy برای CNN + 2 Layer LSTM

همانطور که می‌بینیم دقت نهایی ما برای ۳ مدل CNN + 2 Layer LSTM و 2 Layer LSTM به ترتیب برابر با ۰,۹۱۰۵ و ۰,۹۰۱۰ و ۰,۸۹۶۵ هستند که پیشرفت عملکرد مدل را نشان می‌دهند و مطابق با نتایج مورد انتظار ما از مقاله نیز هستند.