



بسم الله الرحمن الرحيم



دانشگاه صنعتی شریف

پروژه درس تحلیل سری های زمانی

پیش بینی روند قیمت سهام شرکت نتفلیکس به کمک مدل های آریمایی ، هموارسازی نمایی و شبکه عصبی

دانشجو : احمد امامی

شماره دانشجویی : ۹۹۲۰۷۵۲۱

استاد درس : دکتر مجید خدمتی

تابستان ۱۴۰۰

## فهرست مطالب

مقدمه	۴
مرور ادبیات	۵
بررسی داده ها و رسم نمودار زمانی	۶
انتخاب مدل (model selection)	۸
ایستا کردن فرایند	۸
Acf , pacf	۹
فیت کردن مدل آریما	۱۱
پیش بینی با استفاده از مدل آریما	۱۲
هموارسازی نمایی	۱۴
هموارسازی نمایی ساده	۱۴
پیش بینی های مدل هموارسازی نمایی ساده	۱۸
مدل هموارسازی نمایی دوبل	۱۹
پیش بینیهای حاصل از مدل هموارسازی نمایی دوبل	۲۳
مدل شبکه عصبی بازگشتی	۲۴
طراحی شبکه عصبی	۲۵
پیشبینی های مدل شبکه عصبی بازگشتی	۲۸
مدل کردن ناپایداری دادهها به کمک آرچ و گارچ	۳۰
مدل آرچ	۳۰
مدل گارچ	۳۱
فیت کردن مدل گارچ	۳۱
پیش بینی ناپایداری تغییرات قیمت به کمک گارچ	۳۵
نتیجه گیری	۳۶
پیوست ۱	۳۷
پیوست ۲	۴۷



منابع..... ۴۸

## مقدمه

سرمایه‌گذاران بازارهای بورس همواره علاقمندند از روند بعدی قیمت‌ها مطلع شوند. فعالان این بازار درصدد دستیابی و به‌کارگیری روشهایی هستند تا بتوانند با پیشبینی آتی قیمت سهام، سود سرمایه‌ی خود را افزایش دهند. افزایش دسترسی به داده‌های تاریخی، توجه سهامداران را به استفاده از سری‌های زمانی برای پیش‌بینی مقادیر آینده قیمت سهام به خود جلب کرده است. در واقع با استفاده از روش‌های سری‌های زمانی به پیش‌بینی رفتار آینده سهام با توجه از رفتار فعلی و گذشته آن می‌پردازند. نیاز نهادها و سرمایه‌گذاران به تدوین برنامه و استراتژی‌های علمی برای تصمیم‌گیری درخصوص سرمایه‌گذاری، موجب گسترش پژوهش‌های انجام شده در این زمینه و توسعه مدل‌های موجود شده است.

مدل کلاسیک آریمای یکی از رایج‌ترین مدل‌های سری زمانی می‌باشد که استفاده از آن در بحث تحلیل قیمت سهام و داده‌های مالی بدلیل نوسان زیاد داده‌ها در این حوزه چالش برانگیز می‌باشد و ممکن است عملکرد خوبی از خود نشان ندهد. به همین دلیل استفاده از مدل‌های آرچ و گارچ برای مدلسازی واریانس داده‌ها در این حوزه مرسوم می‌باشند. مدل دیگر سری‌های زمانی که کاربرد گسترده‌ای در حوزه‌های مالی دارد مدل هموارسازی نمایی می‌باشد که با توجه به اینکه داده‌ها دارای روند ثابت یا خطی می‌باشند از هموارسازی نمایی ساده یا دوبل استفاده می‌گردد.

از دیدگاه هوش مصنوعی مدل شبکه عصبی مصنوعی<sup>۱</sup>، به ویژه با توانایی یادگیری الگوهای موجود در داده‌ها، یکی از محبوب‌ترین مدل‌ها محسوب می‌شود. در تحقیقات صورت گرفته در حوزه‌های مالی، مدل‌های مختلفی در رابطه با مسئله پیش‌بینی حرکات قیمت دارایی‌ها ارائه شده است که یکی از معروفترین آن‌ها شبکه عصبی بازگشتی می‌باشد که عملکرد بسیار مناسبی در حوزه تحلیل‌های مالی دارد و در تحقیق پیش رو از آن بهره می‌گیریم.

در پژوهش پیش رو قصد داریم با استفاده از مدل‌های سری‌های زمانی و شبکه عصبی مصنوعی به پیش‌بینی قیمت سهام نتفلیکس در ۳ ماهه دوم سال ۲۰۲۱ میلادی بپردازیم و نتایج را با یکدیگر مقایسه کرده و بهترین مدل پیش‌بینی را برگزینیم. سهام شرکت نتفلیکس یکی از ۷ سهام برتر در بازار سرمایه است که توجه سرمایه‌گذاران را به خود جلب کرده است. البته وجود شرکت‌های رقیبی همانند Apple TV+ و بسیاری دیگر باعث ایجاد نگرانی‌های جدی در مورد رقابت با نتفلیکس شده است که در تحقیق پیش رو سعی داریم با پیش‌بینی رشد قیمت‌ها در آینده این نگرانی‌ها را به طرز چشمگیری کاهش دهیم.

مراحل انجام کار در پروژه پیش رو بصورت زیر می‌باشد:

ابتدا به بررسی اولیه داده‌های سری زمانی می‌پردازیم. در بخش بعد به انتخاب مدل‌های مناسب و ارزیابی عملکرد هر مدل و پیش‌بینی داده‌های train با استفاده از مدل‌های انتخابی می‌پردازیم و درنهایت با استفاده از معیار میانگین مربعات خطا<sup>۲</sup> بهترین مدل پیش‌بینی را ارائه می‌دهیم

<sup>1</sup> Artificial Neural Network

<sup>2</sup> MSE

## مرور ادبیات

جستجو و تحقیق در رابطه با مدلی بهینه برای پیش‌بینی قیمت سهام دارای تاریخچه گسترده‌ای است. در این بخش از تحقیق به مطالعات انجام شده در این زمینه که سعی در تولید نتایج بهتر با توجه به پویا بودن این داده‌ها داشتند می‌پردازیم.

[1] Tansel et al به کمک ۳ مدل خطی، شبکه عصبی مصنوعی و مدل ژنتیک به تحلیل قیمت سهام پرداخت و در نهایت به کمک میزان دقت حاصله، میزان زمان محاسباتی<sup>3</sup> و راحت بودن مدلسازی این ۳ مدل را مقایسه نمود. مدل بهینه‌سازی خطی بهترین نتایج را ارائه نمود و پس از آن مدل ژنتیک بهترین عملکرد را روی داده‌های سهام داشت. [2] نیز به مقایسه عملکرد مدل شبکه عصبی مصنوعی و آریمای در پیش‌بینی میزان شاخص سهام بورس کره جنوبی پرداخت. در این مقاله مدل آریمایی نتایج دقیق تری را نسبت به مدل شبکه عصبی ارائه کرد. البته این نتیجه حاصله برای افق‌های زمانی کوتاه مدت قابل ارائه بود. [3] نیز به مقایسه مدل‌های هیبریدی شبکه عصبی مصنوعی و آریمای در پیش‌بینی میزان شاخص بازار بورس در هند پرداخت و در این مقاله نیز این نتیجه حاصل شد که مدل آریمای دقیق تری را پیش‌بینی مقدار قیمت سهام ارائه می‌دهد.

[4] Sterba and hilovska نیز بیان کردند که هر دو مدل آریمای و شبکه عصبی مصنوعی به نتایج بسیار خوبی در بسیاری از سری‌های زمانی در دنیای واقعی دست پیدا می‌کنند و به طور خاص مدل‌های آریمایی در پیش‌بینی سری‌هایی با روند خطی بهتر عمل می‌کنند در حالی که مدل‌های شبکه عصبی در سری‌هایی با الگوهای غیر خطی عملکرد مناسب تری دارد. در مطالعه ای مشابه بر روی داده‌های مالی که توسط [5] گزارش شد، بیان گردید که مدل شبکه عصبی عملکرد مناسب تری نسبت به مدل آریمای دارد.

[6] Yao et al نیز از آریمای و شبکه عصبی برای بررسی داده‌های سهام استفاده نمود و نشان داد شبکه عصبی نتایج بهتری را ارائه می‌کند. [7] Hansen et al. نیز در مطالعه ای مشابه به همین نتیجه دست پیدا کرد و بیان کرد که مدل شبکه عصبی توانایی کشف الگوهای پنهانی را در داده‌های سهام دارد. [8] Prybutok et al. نیز با بررسی و مقایسه این دو مدل در پیش‌بینی میزان غلظت ازن به این نتیجه رسید که مدل شبکه عصبی عملکرد بهتری دارد.

همان طور که مشاهده می‌شود نقطه نظرات متفاوتی در مورد عملکرد مدل‌های شبکه عصبی و مدل‌های کلاسیک آریمایی در کاربردهای مختلف وجود دارد. علت این موضوع نیز به کار رفتن داده‌های متفاوت در این تحقیقات است که باعث می‌شود عملکرد مدل‌ها را تحت تاثیر قرار دهد. در این پروژه ما سعی داریم با تجربیات گذشته و مدل‌های در دسترس بررسی جامعی بر روی قیمت سهام شرکت نفت‌فلیکس داشته باشیم و در نهایت بهترین مدل انتخابی را ارائه کنیم.

---

<sup>3</sup> Computational time

## بررسی داده ها و رسم نمودار زمانی

اولین مرحله از هر تحلیل سری زمانی رسم نمودار زمانی و تشخیص وجود ترند یا فصلی بودن در داده ها می باشد. همان طور که میدانیم یکی از مهم ترین فرضیات در مدل سازی های سری زمانی *stationary* بودن داده ها می باشد پس قبل از هرگونه مدلسازی بر روی دیتای مدنظر باید از این موضوع اطمینان حاصل کنیم. به کمک نمودار زمانی داده ها میتوانیم دید مناسبی از داده های خود به دست آوریم و تشخیص دهیم آیا ترند و یا سیزنالیته در داده ها موجود است یا خیر. در صورت دیده شدن هر یک از این علائم باید به کمک تبدیل های مربوطه داده های خود را *stationary* کرده تا بتوانیم تحلیل های مربوطه را انجام بدهیم.

تصویر زیر قیمت روزانه سهام شرکت نتفلیکس را از ابتدای سال ۲۰۱۸ میلادی نشان می دهد که مطابق با انتظار ما نیز هست. با توجه رشد صنعت فیلم سازی و اینکه در دهه اخیر مردم تمایل بیشتری به تماشای فیلم و سریال مورد علاقه شان در منزل دارند انتظار یک روند صعودی را داشتیم. همچنین با توجه به ثبات کم و ناپایداری بازار سهام انتظار دیده شدن نوسان های بسیار زیادی نیز بودیم که در شکل میتوان این موضوع را نیز مشاهده کرد.



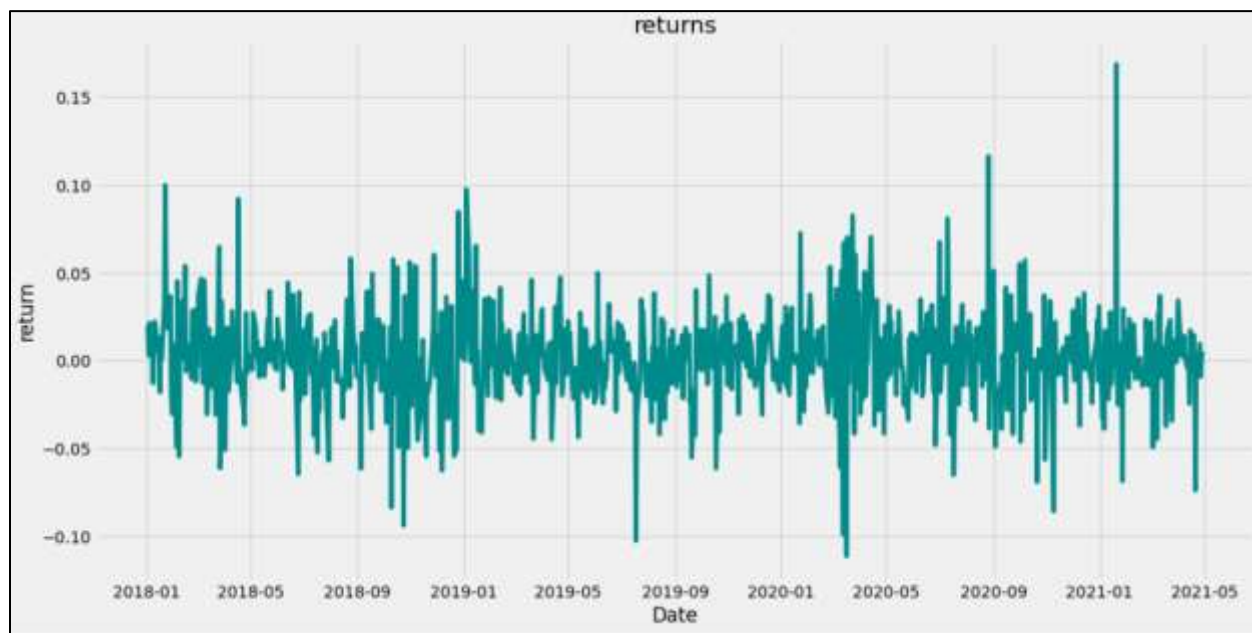
علاوه بر نمودار های گرافیکی میتوان به طور آماری نیز نگاهی به دیتاست مورد بررسی مان بیندازیم. برای این مهم نیز از دستورات موجود در پایتون بهره می گیریم که خلاصه ای از داده ها ما را به شکل زیر ارائه می دهد:

count	837.000000
mean	381.247312
std	88.776410
min	201.070007
25%	315.339996
50%	359.970001
75%	466.929993
max	586.340027

همان طور که مشاهده می شود ۸۷۳ عدد مشاهده در دسترس داریم که میانگین قیمت سهام در این بازه ۳۸۱ دلار و انحراف معیار آن نیز ۸۸.۷۷ گزارش شده که باز هم نشان می دهد داده های ما دارای نوسان زیادی هستند. در ادامه نیز اطلاعاتی از جمله چارک های اول و دوم و سوم و نیز ماکسیمم و می نیمم داده ها گزارش شده است.

اما نکته ای که در بازار سهام و به طور کلی بازارهای مالی از جمله بازار های ارز رمز ها مطرح است تغییرات کوتاه مدت است بطوری که وقتی از پیش بینی در این بازار صحبت می شود معمولا پیش بینی تغییرات است نه قیمت مطلق یعنی ما می خواهیم بدانیم که در طی روزهای آتی مثلا ۷ روز آینده تغییرات قیمت به چه شکل خواهد بود. در واقع هدف ما این است که بدانیم و پیش بینی کنیم که قیمت در طی چند روز یا چند هفته آتی چه روندی را طی می کند و صرفا پیش بینی قیمت مد نظر نیست. به همین علت معمولا برای تحلیل های سری زمانی بر روی داده های بازار سهام داده ها را به شکل درصد تغییر قیمت نسبت به روز گذشته ( نرخ بازگشت) گزارش می کنند. به عنوان مثال اولین مقدار از این دیتاست تبدیل شده عبارت خواهد بود از اختلاف قیمت سهام در روز دوم از روز اول، تقسیم بر قیمت سهام در روز اول.

از نرخ بازگشت قیمتی استفاده های متعددی در تحلیل های سری زمانی می شود و در این پروژه نیز ما از این دیتاست استفاده خواهیم کرد. از آنجایی که نرخ بازگشت تقریبا ایستا است و نوسانات و ناپایداری های قیمت سهام را به خوبی نشان می دهد، معمولا از آن در مدل های آرچ و گارچ بهره می گیرند و تحلیل های مورد نظر را بر روی این داده ها انجام می دهند. در شکل زیر نمودار نرخ بازگشت قیمتی را مشاهده می کنید که از ابتدای سال ۲۰۱۸ میلادی تا ماه می سال ۲۰۲۱ رسم شده است:



همان طور که می بینید بر خلاف نمودار قیمت مطلق سهام، که در بالا رسم کردیم، نمودار نرخ بازگشت ایستا به نظر می رسد. علت آن نیز دیفرانسینگ انجام شده در دیتاست مان برای به دست آوردن تغییرات روزانه است.

## انتخاب مدل (model selection)

مدل های مختلفی را میتوان برای تحلیل داده های بازار سهام مورد استفاده قرار داد. به عنوان مثال با توجه به نوسان زیاد و ناپایداری بالای داده های بازار بورس، مدل های آرچ<sup>۴</sup> و گارچ<sup>۵</sup> نشان داده اند که در مدل کردن این نوسانات بسیار خوب عمل می کنند. مدل های arima نیز از مدل هایی هستند که در این پروژه از آن ها بهره خواهیم برد هرچند با توجه به نوسانات زیاد داده های ما به نظر نمی رسد عملکرد چندان مناسبی را برای پیش بینی داده هایمان داشته باشند. با توجه به اینکه نوسانات فصلی در داده های ما دیده نمیشود به نظر می رسد مدل sarima یا آریمای فصلی عملکرد مناسبی بر روی مدل ما نداشته باشد و میتوانیم آن را از لیست مدل های انتخابی کنار بگذاریم. یکی دیگر از مدل های مناسب، مدل هموارسازی نمایی خواهد بود که با توجه به روند صعودی و مشاهده ترند در داده های ما، انتظار می رود مدل های هموارسازی مرتبه<sup>۶</sup> و همچنین هموارسازی دوبل<sup>۷</sup> بتوانند عملکرد مناسبی را ارائه دهند.

برای پیش بینی داده های سری زمانی میتوان از الگوریتم های یادگیری ماشین نیز استفاده کرد که در بسیاری از مواقع عملکرد خیلی خوبی دارند. یکی از مهمترین الگوریتم های یادگیری ماشین که برای داده های متوالی من جمله داده های سری زمانی مورد استفاده قرار می گیرد، شبکه های عصبی بازگشتی<sup>۸</sup> هستند. از این روش در پروژه پیش رو برای تحلیل داده های خود بهره خواهیم گرفت و توضیحات بیشتری را نیز از شیوه کار این الگوریتم ارائه می کنیم.

## ایستادگی کردن فرایند

به منظور مدل سازی دیتاست خود را به دو دسته TRAIN و TEST تقسیم میکنیم. بدین منظور ۹۰ درصد داده هایمان را به منظور آموزش مدل و ۱۰ درصد دیگر را برای ارزیابی دقت پیش بینی مدل برمیگزینیم. علت استفاده از این نسبت نیز بازه زمانی بزرگ دیتاهای ماست و سعی داریم تنها بازه ای حدود ۲ ماه را پیش بینی کنیم و نه بیشتر. برای اینکه تشخیص دهیم کدام مدل برای فیت کردن به دیتای ما مناسب تر است نیاز است تا ابتدا نمودارهای acf و pacf را برای دیتاست train رسم کنیم. پیش از رسم pacf و acf باید توجه کنیم که باید فرایند را ایستا<sup>۹</sup> بنماییم. با توجه به نمودار رسم شده میتوان روند صعودی در قیمت سهام را مشاهده کرد پس به نظر میرسد اولین قدم در ایستاد سازی فرایند اعمال دیفرنسینگ بر روی داده ها است.

<sup>4</sup> arch

<sup>5</sup> garch

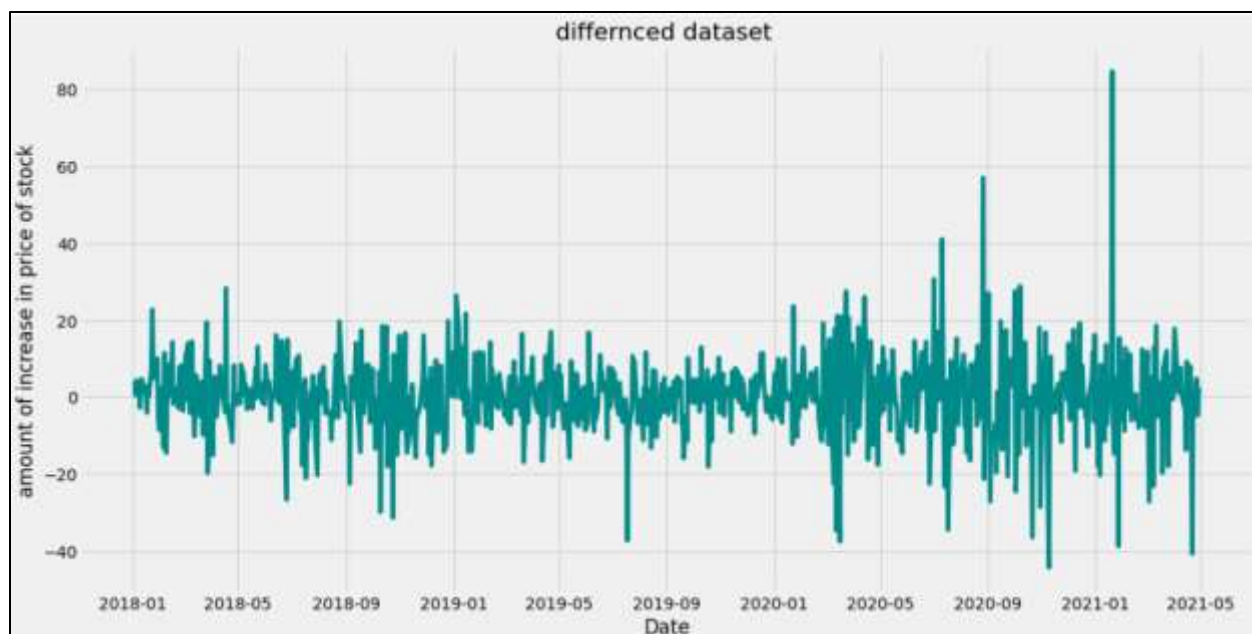
<sup>6</sup> Second order exponential smoothing

<sup>7</sup> Double exponential smoothing

<sup>8</sup> Recurrent neural network

<sup>9</sup> stationary



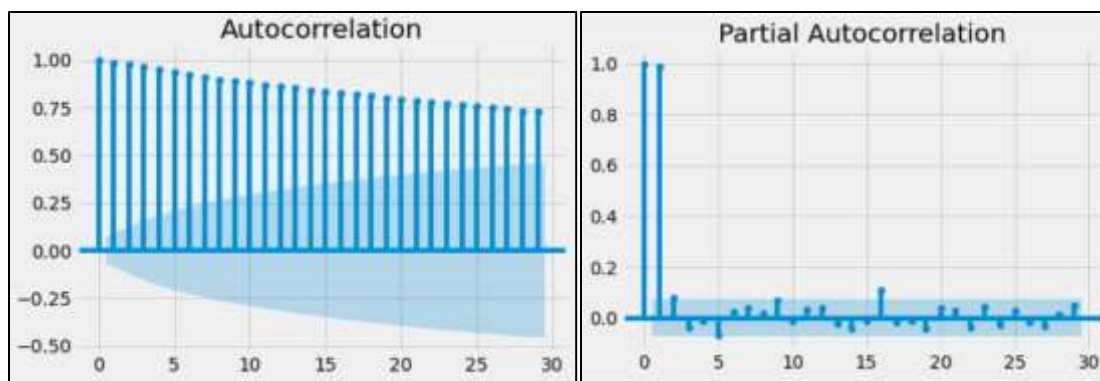


همان طور که مشاهده میشود داده های ما در یک میانگین ثابتی حرکت می کند و ایستا به نظر می رسند. برای اطمینان از ایستا بودن فرایند میتوان از آزمون های آماری نیز استفاده کرد. یکی از پرکاربرد ترین آزمون های آماری آزمون " Augmented Dickey–Fuller" است. با اجرای این تست بر روی داده های بدست آمده در بالا مقدار  $p$ -value بسیار کوچک خواهد بود و این نشان دهنده رد فرض صفر و پذیرش ایستا بودن فرایند است.

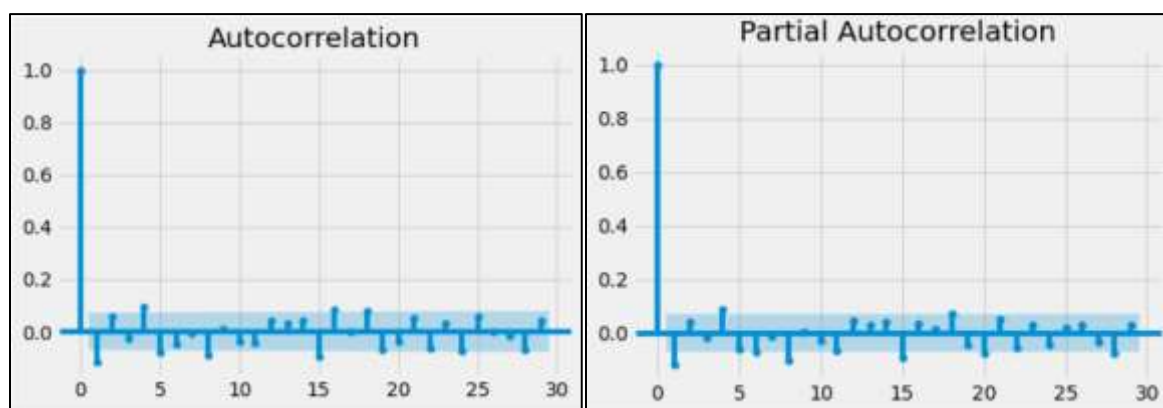
اما دیتاست تغییرات قیمتی با توجه به شکلی که بالاتر مشاهده کردیم به نظر ایستا می آید. برای اطمینان از این موضوع آماره تست "Augmented Dickey–Fuller" را بر روی آن اجرا می کنیم. این تست آزمون ریشه واحد است و با بررسی نتایج آن میتوان ایستایی یا غیرایستایی فرایند را نتیجه گرفت. مقدار  $p$ -value حاصله از این تست نیز بسیار کوچک است که نشان می دهد فرایند ایستا است و مشکلی از این نظر وجود ندارد. ( نتایج تست ها در فایل pdf کد پروژه موجود است)

## Acf , pacf

پس از این که فرایند ما ایستا شد حال نوبت آن است که نمودار های  $acf$  و  $pacf$  مرتبط به آن را رسم کنیم. این دو ابزار دید خوبی از فرایند مورد بحث ارائه خواهند کرد بخصوص زمانی که بخواهیم از مدل های  $arima$  برای فیت کردن دیتاهای خود استفاده کنیم. به کمک این دو ابزار میتوان حدس و تخمین هایی را از پارامتر های مدل ارائه داد اما نمیتوان به طور دقیق مدلی را ارائه نمود. در زیر این دو نمودار را برای داده های قیمت سهام قبل از اعمال دیفرنسینگ مشاهده می کنید:



همان طور که مشاهده میکنیم نمودار acf نیز به طور مجدد نشان از عدم ساکن بودن فرایند دارد زیرا مقدار همبستگی میان لگ ها به شکل نمایی نزول نمی کند. در نتیجه برای تشخیص اردر فرایند نیاز داریم تا یک بار دیفرنسینگ بر روی داده ها انجام داده و مجدداً آن را بررسی کنیم که دو نمودار acf و pacf برای داده های دیفرنس شده در زیر مشاهده می کنید:



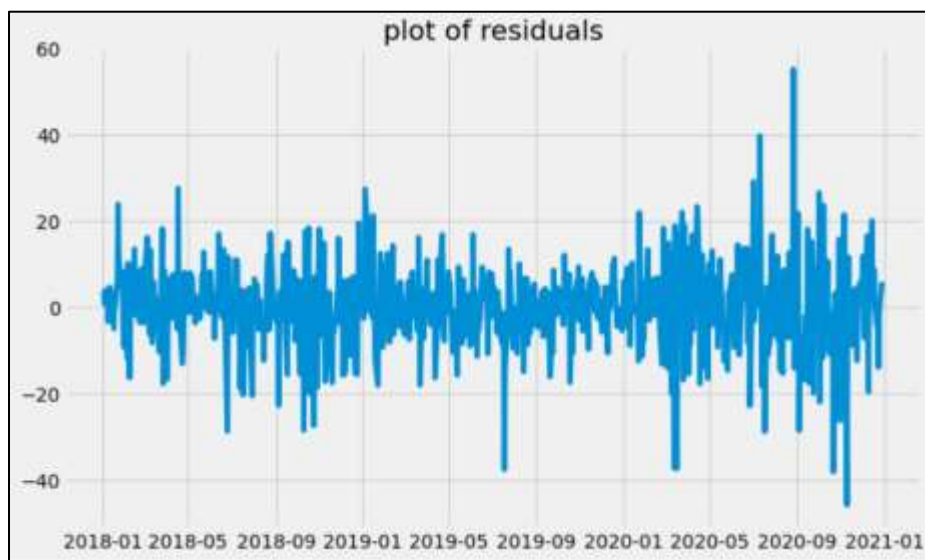
با توجه به نمودار acf مشاهده میکنیم که تقریباً هم بستگی قابل توجهی میان لگ های مختلف مشاهده نمی شود و در ۵ لگ ابتدایی لگ های ۱،۴،۵ خارج از حدود مشخصه هستند در نتیجه صحبت قطعی ای نمیتوان در ارتباط با فرایند MA کرد و در نتیجه نمیتوان اردر مشخصی را برای آن حدس زد. در طرف دیگر نمودار pacf را داریم که شرایط مشابهی در آن برقرار است و همه لگ ها تقریباً در داخل حدود کنترلی هستند و روند خاصی مشاهده نمی شود، در نتیجه نمیتوان اردر مشخصی را برای فرایند AR آن حدس زد.

با این تفاسیر و با توجه به اطلاعات بدست آمده حالت های مختلفی را برای مدل arima بر روی داده های خود مدل می کنیم و مدلی که کمترین میزان شاخص aic یا آکائیکه<sup>۱۰</sup> را دارد به عنوان مدل برگزیده برای فیت کردن داده های train استفاده می شود. با کمک دستور auto.arima و انتخاب شاخص aic برای انتخاب مدل و همچنین محدود کردن مقدار p، q به حداکثر ۵ ( برای جلوگیری از پیچیده شدن مدل ) دستور را اجرا می کنیم. بهترین مدل پیشنهادی برای فیت کردن به داده های دیفرنس شده قیمت سهام arma(0,5) یا همان میانگین متحرک با اردر ۵ است و در نتیجه میتوان نتیجه گرفت که مدل مناسب به منظور فیت کردن به داده های پیش از اعمال دیفرنسینگ نیز arima(0,1,5) است.

<sup>10</sup> Akaike information criterion

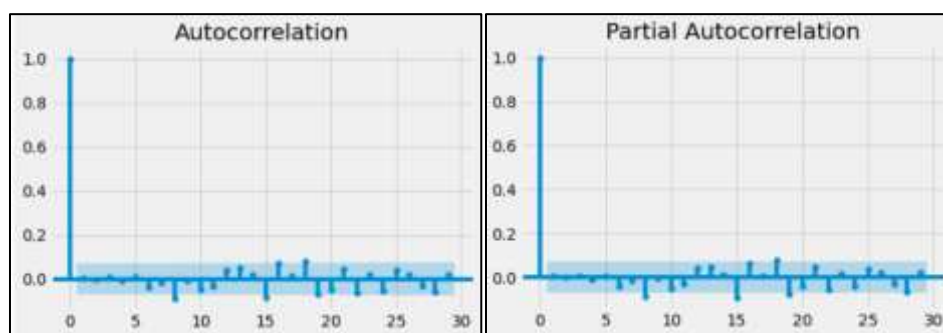
## فیت کردن مدل آریمما

پس از آنکه در قسمت قبل مدل  $\text{arima}(0,1,5)$  را به عنوان بهترین مدل برای فیت کردن به داده‌هایمان تشخیص دادیم حال نتایج حاصله از این مدل را در این بخش مشاهده میکنیم. اولین نموداری که مشاهده میکنیم نمودار رزیدوال‌ها یا تفاوت مقادیر فیت شده از مقدار واقعی داده‌های آموزشی است:



همان طور که میبینیم با توجه به میانگین فرایند رزیدوال‌ها تقریباً ساکن هستند هرچند نوسان‌های زیادی دیده می‌شود که نشان از عدم ثبات واریانس دارد که علت آن به نامناسب بودن مدل‌های آریمما برای پیش‌بینی داده‌های مالی برمیگردد.

برای بررسی بیشتر می‌توانیم  $\text{acf}$  و  $\text{pacf}$  باقی‌مانده‌های مدل فوق را رسم کنیم که در زیر میتوانیم مشاهده کنیم:

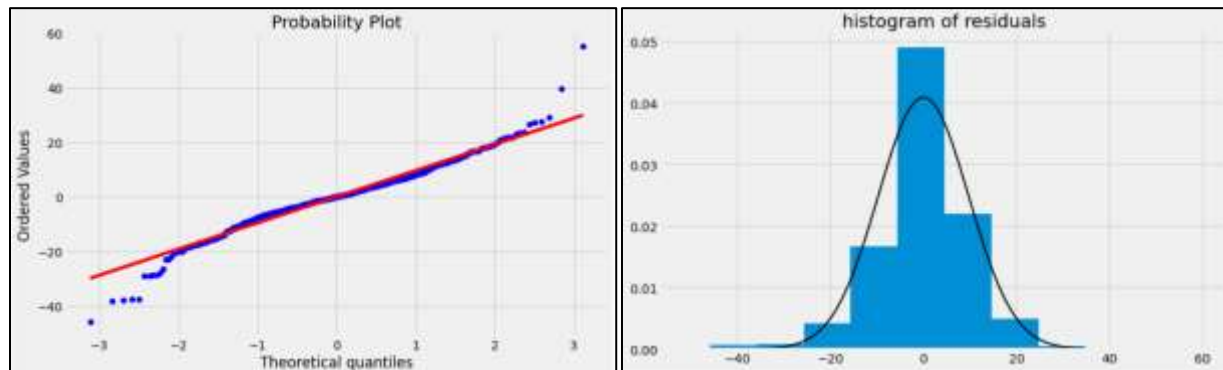


همان‌طور که مشاهده می‌شود تعدادی از لگ‌ها هنوز مقدار قابل توجهی دارند که نشان می‌دهد کماکان الگوهایی در داده‌های ما وجود دارد که مدل قادر به تشخیص آن‌ها نبوده است. برای بررسی بیشتر می‌توانی از آزمون آماری لیونگ باکس استفاده کنیم تا همبستگی سریالی را در داده‌ها تشخیص بدهیم. نتایج حاصل از این تست را در زیر مشاهده می‌کنید:

	lb_stat	lb_pvalue	bp_stat	bp_pvalue					
1	0.007739	0.929899	0.007708	0.930038	11	10.029214	0.527761	9.892236	0.540106
2	0.008730	0.995645	0.008694	0.995663	12	11.333266	0.500594	11.172074	0.514232
3	0.048189	0.997227	0.047891	0.997252	13	13.035429	0.445079	12.840375	0.460216
4	0.094221	0.998925	0.093557	0.998939	14	13.263981	0.505851	13.064077	0.521484
5	0.139683	0.999631	0.138597	0.999638	15	18.726336	0.226404	18.403275	0.242058
6	1.275105	0.973013	1.261972	0.973713	16	22.436012	0.129663	22.024391	0.142406
7	1.521556	0.981554	1.505482	0.982119	17	22.589018	0.163114	22.173541	0.178137
8	7.398120	0.494349	7.304107	0.504205	18	27.293636	0.073650	26.753369	0.083739
9	7.445984	0.590790	7.351273	0.600597	19	30.960230	0.040780	30.317843	0.047890
10	9.158638	0.517117	9.036670	0.528627	20	32.791281	0.035563	32.095468	0.042292

ستون اول و دوم به ترتیب آماره آزمون و مقدار  $p$ -value آزمون لیونگ باکس را نشان می‌دهد و دو ستون دیگر نیز به همین ترتیب آماره آزمون و  $p$ -value آزمون پیرس را نمایش می‌دهد. با توجه به مقادیر حاصل شده میتوان گفت که تا لگ ۱۸ ام همبستگی‌ای توسط آزمون تشخیص داده نمی‌شود و باقی‌مانده‌ها تقریباً وایت نویز هستند ولی بعد از آن به نظر می‌رسد مدل همبستگی‌ای را تشخیص می‌دهد. در نتیجه به طور کامل نمیتوان عنوان کرد که باقی‌مانده‌ها وایت نویزند و مقدار اندکی با وایت نویز بودن فاصله دارند.

نمودارهای دیگری که باید بررسی کنیم نمودار توزیع احتمال و هیستوگرام است تا ببینیم که تا چه حد توزیع رزیدوال‌های مدل به مدل نرمال نزدیک است:

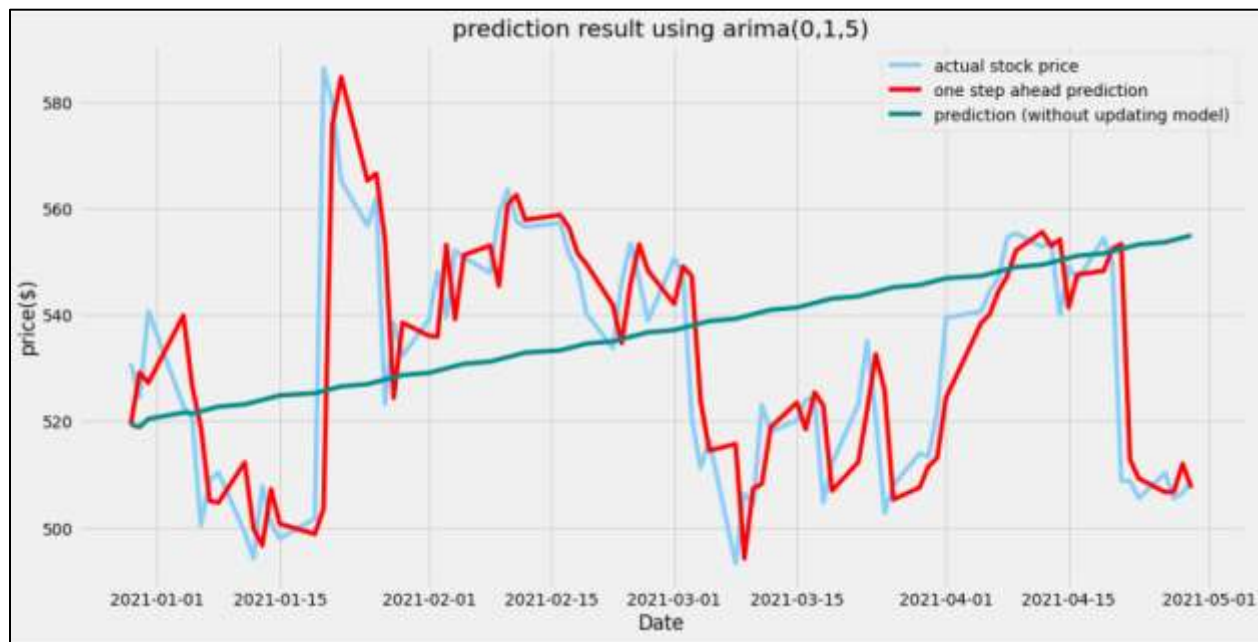


میبینیم که توزیع داده‌ها تقریباً دارای توزیع نرمال هست هر چند کمی انحراف در برخی نقاط مشاهده می‌شود. در ادامه به پیش بینی‌های حاصل از مدل نگاه می‌اندازیم.

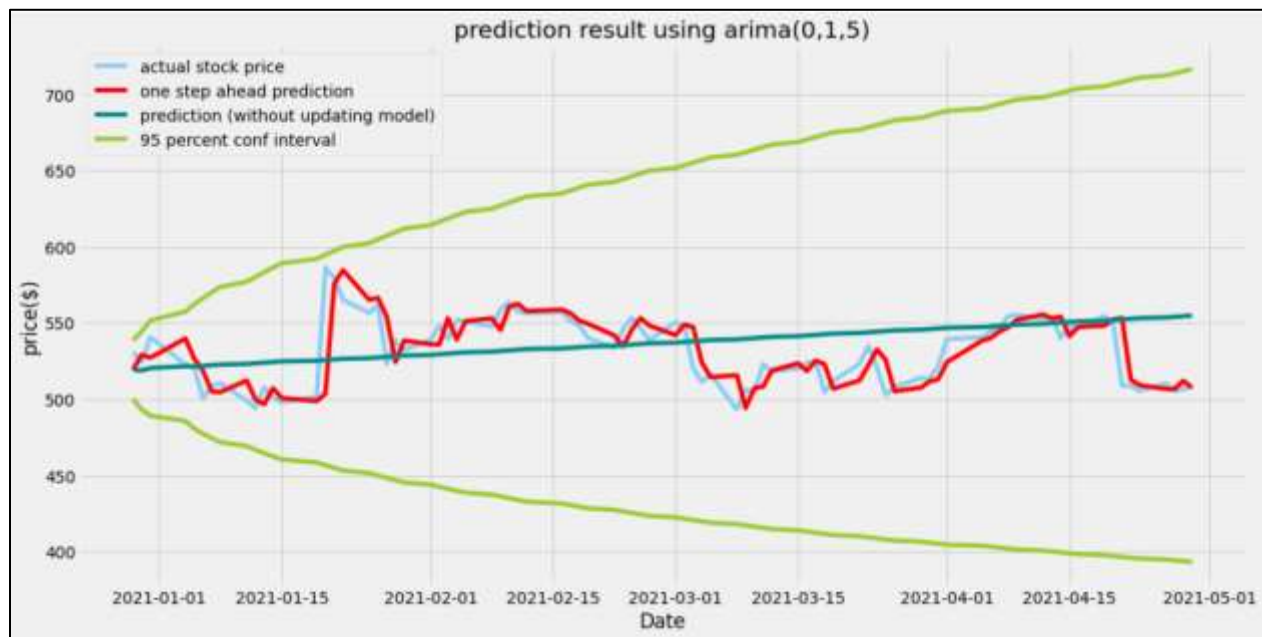
## پیش بینی با استفاده از مدل آریمای

در این بخش پیش بینی‌های حاصل از مدل آریمای انتخاب شده بررسی میکنیم تا ببینیم تا چه حد این مدل میتواند در پیش بینی قیمت سهام مدنظرمان درست عمل کند. همان طور که مشاهده می‌شود اگر افق پیش بینی دراز مدت را در نظر بگیریم و تمام داده‌های تست را به یک باره پیش‌بینی کنیم، خط نارنجی پیش‌بینی‌های حاصله را نشان می‌دهد که بسیار با واقعیت فاصله دارد اما خط قرمز رنگ پیش‌بینی‌های یک افق زمانی آینده هستند. یعنی با هر بار پیش‌بینی پریود بعدی مدل بروزرسانی شده و مدلمان

مقدار واقعی قیمت در پریود پیش‌بینی شده را در نظر می‌گیرد و مجدداً پریود بعدی را پیش‌بینی می‌کند. به وضوح می‌بینیم که این پیش‌بینی‌ها نتایج دقیق‌تری را ارائه می‌دهند.



علاوه بر شکل بالا می‌توانیم حدود فاصله اطمینان نیز برای پیش‌بینی‌های مدل آرمایی تعیین کنیم. در شکل زیر پیش‌بینی‌های مدل به همراه فاصله اطمینان ۹۵ درصد ارائه شده‌اند. در نهایت نیز مقادیر میانگین مربعات خطا را برای پیش‌بینی‌های یک افق آینده را حساب می‌کنیم و در ادامه این مقدار را با مقادیر حاصله از دیگر مدل‌ها مقایسه می‌کنیم.

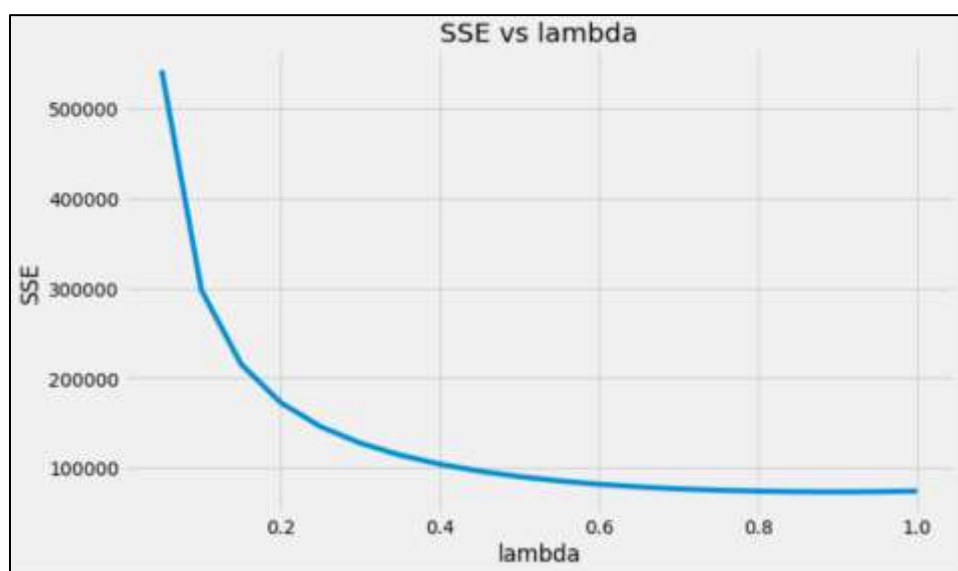


## هموارسازی نمایی

دومین دسته از مدل هایی که برای پیش بینی سری های زمانی مطرح شد مدل های هموار سازی نمایی بودند که خود شامل ۳ دسته ی ساده ، دوپل ، تریپل می شوند. مدل های ساده برای روند های ثابت مناسب تر است در حالی که مدل دوپل برای مشاهدات دارای روند صعودی یا نزولی و مدل تریپل برای داده های فصلی مناسب تر است. مدل های هموارسازی نمایی از دسته مدل هایی است که در صنعت نیز کاربرد زیادی دارد و برای پیش بینی مقدار فروش و یا تولید بسیار مورد استفاده قرار می گیرد. در این بخش از دو مدل هموارسازی نمایی و هموارسازی دوپل را برای تحلیل داده های خود بهره می گیریم. از آنجایی که روند فصلی در داده هایمان دیده نمی شود دیگر به مدل تریپل نمی پردازیم.

## هموارسازی نمایی ساده

قبل از فیت کردن مدل هموارسازی نمایی ساده به داده ها نیاز داریم که مقدار بهینه پارامتر آلفا یا ضریب تاثیر داده ی نزدیک تر در هموارسازی را پیدا کنیم. بدین منظور بازه ی اعداد بین ۰ و ۱ را با گامی برابر با ۰.۰۵ طی می کنیم و هر بار مدلمان را به داده های آموزشی خود فیت کرده مجموع مربعات خطا<sup>۱۱</sup> را محاسبه می کنیم. پس از انجام این گام  $\alpha=0.9$  بهینه ترین مقدار خواهد بود. البته میتوان از دستورات موجود در پایتون نیز بهره گرفت و آلفای بهینه را حساب کرد که در آن صورت  $\alpha=0.89$  به دست می آید که تفاوت چندانی با آن چه که به دست آوردیم وجود ندارد. نمودار SSE بر حسب آلفا را در نمودار زیر مشاهده می کنید:



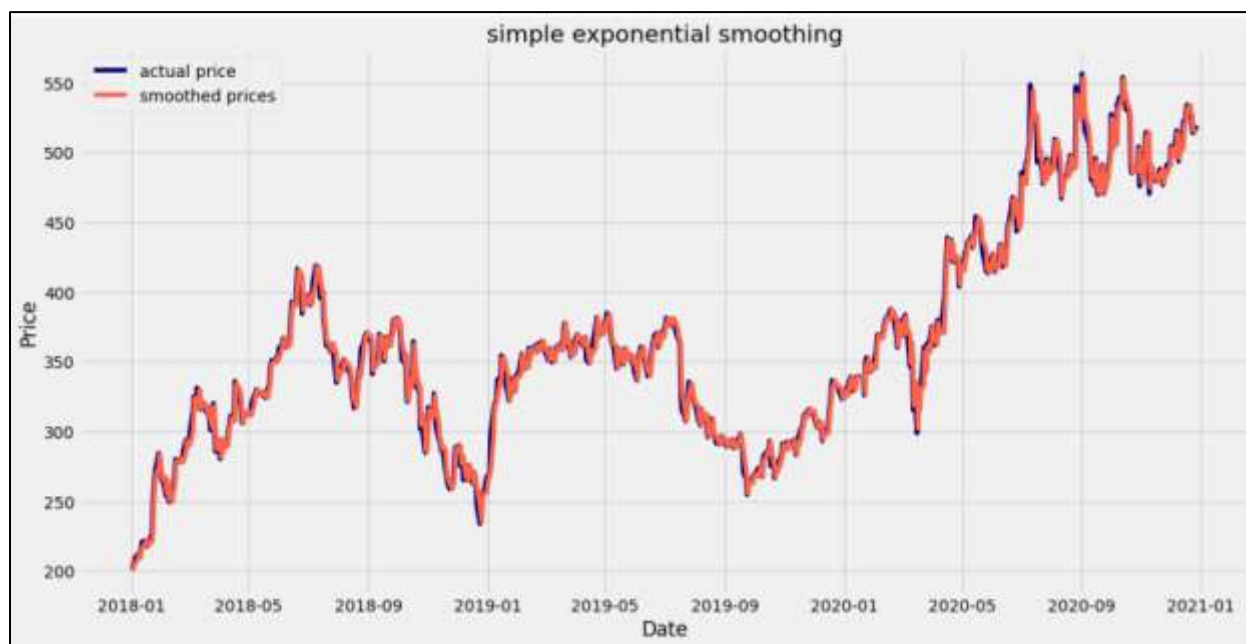
خلاصه ای از پارامترهای مدل فیت شده را در کادر زیر تماشا می کنید:

<sup>۱۱</sup> SSE

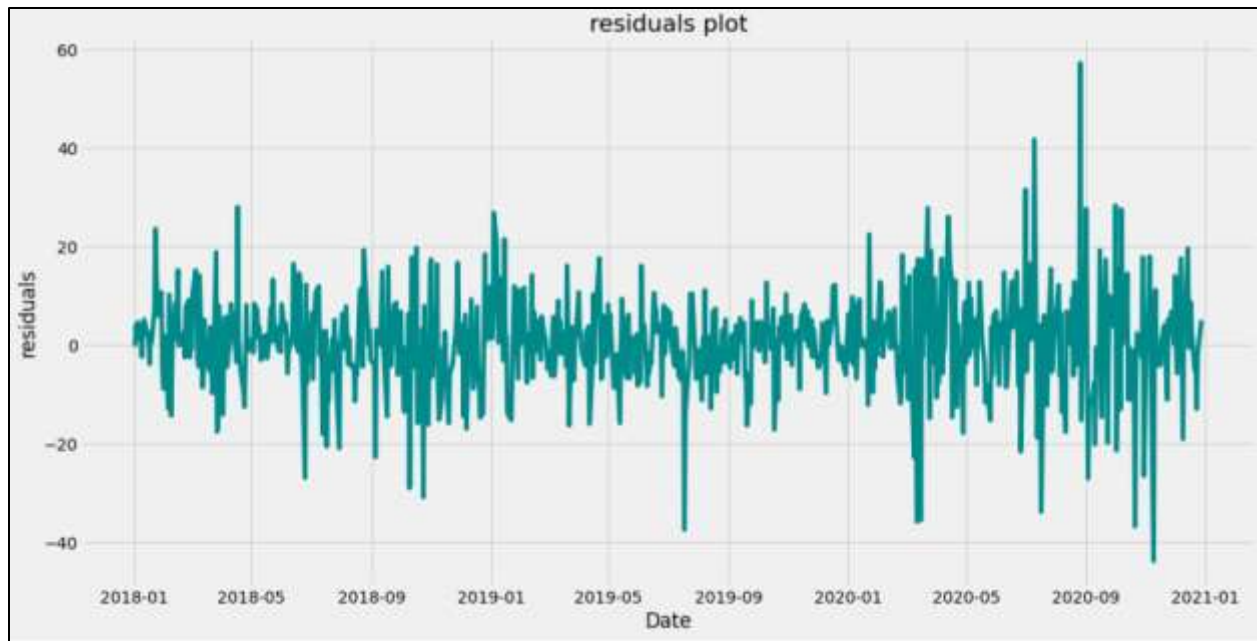


SimpleExpSmoothing Model Results			
Dep. Variable:	close	No. Observations:	753
Model:	SimpleExpSmoothing	SSE	72914.115
Optimized:	True	AIC	3447.448
Trend:	None	BIC	3456.696
Seasonal:	None	AICC	3447.502
Seasonal Periods:	None	Date:	Sat, 12 Jun 2021
Box-Cox:	False	Time:	20:46:29
Box-Cox Coeff.:	None		
	coeff	code	optimized
smoothing_level	0.8963960	alpha	True
initial_level	201.49371	1.0	True

در نمودار زیر مشاهداتمان را در کنار مقادیر فیت شده به کمک مدل هموارسازی نمایی ساده میبینیم. همانطور که به نظر می‌رسد مدل با دقت خوبی داده‌ها را فیت کرده است و هم‌پوشانی مناسبی دارد. این مسئله ممکن است این نگرانی را پیش آورد که اورفیت رخ داده و مدل در پیش بینی عملکرد مناسبی نداشته باشد. در ادامه این موضوع را نیز بررسی می‌کنیم.

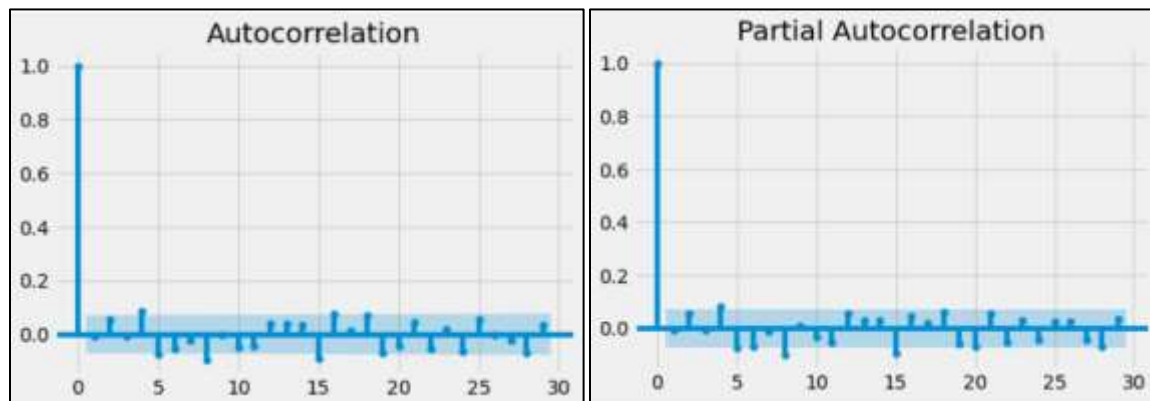


هم چنین رزیجوال‌های مدل بالا به شکل زیر خواهد بود:



برای بررسی رزیدوال‌های حاصل از فیت کردن مدل هموارسازی نمایی ساده به داده‌هایمان مانند قبل از ابزارهایی مثل  $acf$  ،  $pacf$  ، نمودار توزیع داده‌ها استفاده می‌کنیم.

در زیر دو نمودار  $acf$  و  $pacf$  مرتبط را مشاهده می‌نمایید:



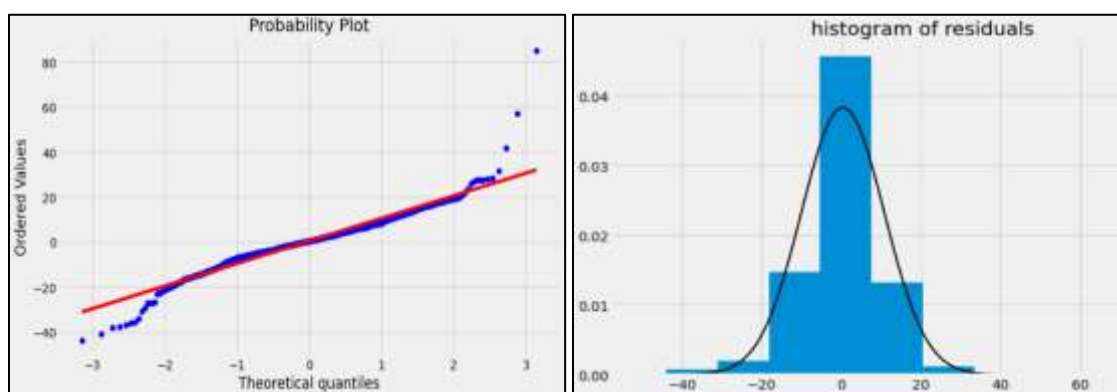
همان‌طور که مشاهده می‌شود می‌بینیم تمام لگ‌ها در محدوده‌ی مشخص واقع نشدند و هنوز تعدادی لگ با مقدار قابل توجه چه در نمودار  $acf$  و  $pacf$  دیده می‌شود. این موضوع نشان‌دهنده‌ی این است که هنوز توزیع باقی‌مانده‌ها تصادفی نیست و با وایت‌نویز فاصله دارد در نتیجه تمام الگوهای موجود در داده‌های ما توسط مدل کشف نشده است.

برای اطمینان از این موضوع آزمون لیونگ باکس را بر روی باقی‌مانده‌ها اجرا می‌کنیم. نتایج این آزمون به ترتیب زیر است:



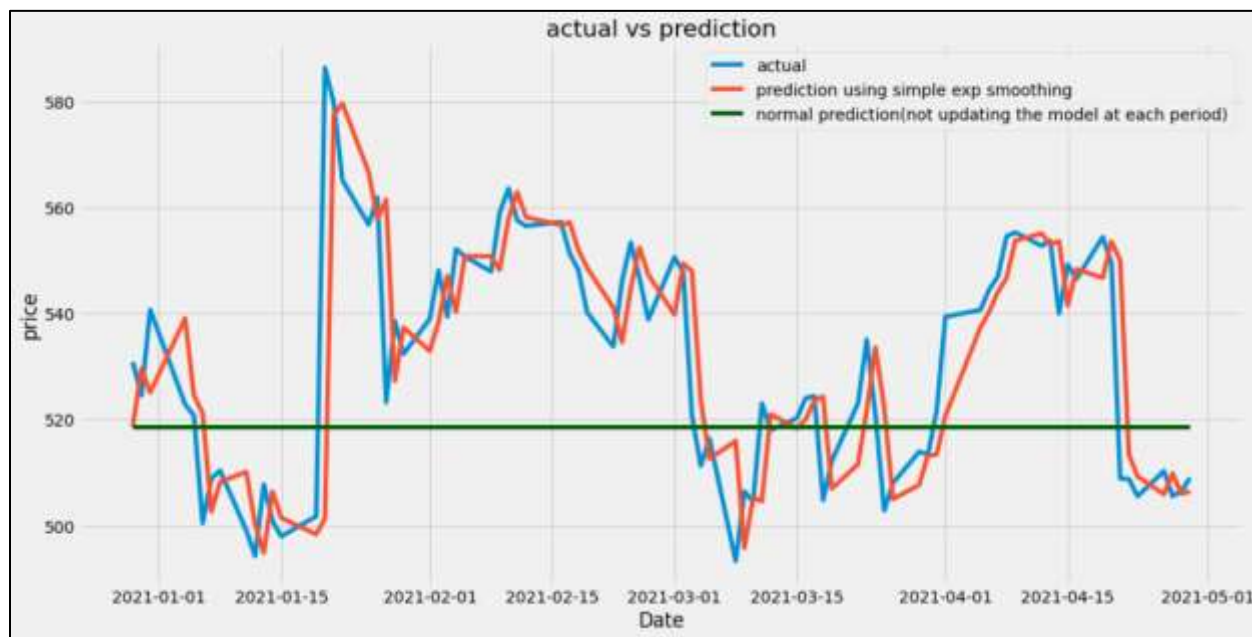
	lb_stat	lb_pvalue	bp_stat	bp_pvalue					
1	0.049345	0.824207	0.049149	0.824551	11	24.382267	0.011214	24.119609	0.012238
2	2.244913	0.325479	2.233085	0.327410	12	25.586356	0.012276	25.301370	0.013458
3	2.315222	0.509611	2.302928	0.511960	13	26.873072	0.012948	26.562523	0.014273
4	7.619350	0.106560	7.564904	0.108881	14	27.848150	0.014903	27.516937	0.016479
5	11.890964	0.036313	11.796914	0.037679	15	33.756972	0.003683	33.292713	0.004275
6	14.272591	0.026735	14.153305	0.027970	16	38.283402	0.001379	37.711228	0.001664
7	14.647647	0.040791	14.523890	0.042610	17	38.441499	0.002136	37.865346	0.002565
8	21.321387	0.006341	21.109237	0.006863	18	42.294589	0.001006	41.616368	0.001251
9	21.329868	0.011264	21.117593	0.012138	19	46.112667	0.000478	45.328248	0.000616
10	22.971521	0.010852	22.733154	0.011776	20	47.505523	0.000499	46.680517	0.000650

در ادامه مانند مدل‌های قبل نگاهی به توزیع رزیدوال‌ها می‌اندازیم تا ببینیم تا چه حد نرمال اند:



همان‌طور که مشاهده می‌شوند داده‌ها با کمی انحراف تقریباً توزیع نرمالی از خود نشان می‌دهند و وجود این انحراف‌ها نشان از عدم فیت شدن مناسب داده‌هایمان است. در ادامه پیش‌بینی‌های حاصله از این مدل را بررسی کرده‌ایم.

## پیش بینی های مدل هموارسازی نمایی ساده



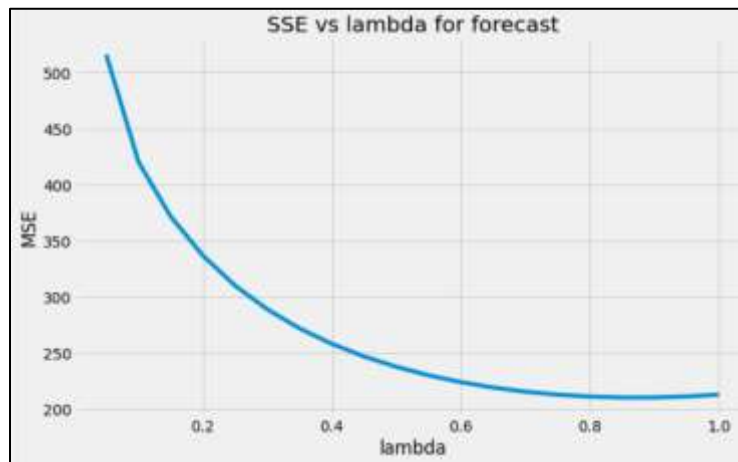
همان طور که در شکل مشاهده می کنیم و از قبل می دانیم پیش بینی های مدل هموارسازی نمایی برای تمام پریودهای آینده یک عدد یکسان و برابر با  $\bar{Y}_t$  است و اگر بدون بروزرسانی مدل تمام پریودهای مدنظر را پیش بینی کنیم ، مانند خط سبز رنگ نمودار بالا خواهد بود. اما بهتر است با هر بار پیش بینی مقدار واقعی پریود پیش بینی شده را در مدل لحاظ کرده و مجددا یک پریود آینده را پیش بینی کنیم که با خط قرمز رنگ در شکل بالا مشاهده میشود که تقریبا عملکرد مناسبی داشته است. و میانگین مربعات خطای پیش بینی نیز برابر با ۲۱۰ به دست آمده که در فایل کد نرم افزار نیز قابل مشاهده است.

قبل از بررسی مدل های دیگر باید به این نکته توجه کنیم که بهترین مدل انتخابی ما باید مدلی باشد که بهترین و دقیق ترین پیش بینی ها را ارائه دهد و تنها روی داده های آموزش بهترین عملکرد را نداشته باشد در نتیجه به منظور جلوگیری از انتخاب یک مدل اورفیت شده باید پیش بینی های حاصل را نیز مقایسه کنیم.

برای مقایسه دقت پیش بینی ها نیز میتوان از میانگین مربع خطای پیش بینی یک پریود آینده استفاده کرد. به همین منظور بازه ای از لاندا ها را بین ۰ و ۱ با گام ۰.۰۵ برمیگزینیم و هر بار مدل هموارسازی ساده با لاندا ی مربوطه را به داده های آموزشمان فیت کرده و در انتها میانگین مجموع مربعات خطای پیش بینی یک پریود بعدی را محاسبه میکنیم. خروجی این قطعه کد بدین شکل خواهد بود:

simple exponential model with landa = 0.05	MSE will be	515.5250111950622
simple exponential model with landa = 0.1	MSE will be	420.0820990263477
simple exponential model with landa = 0.15	MSE will be	371.3731597541392
simple exponential model with landa = 0.2	MSE will be	336.6228319316936
simple exponential model with landa = 0.25	MSE will be	309.86983271605186
simple exponential model with landa = 0.3	MSE will be	288.7618963838183
simple exponential model with landa = 0.35	MSE will be	271.82909878496383
simple exponential model with landa = 0.4	MSE will be	258.07734949829694
simple exponential model with landa = 0.45	MSE will be	246.8274782332964
simple exponential model with landa = 0.5	MSE will be	237.6078249115878
simple exponential model with landa = 0.55	MSE will be	230.08269168454007
simple exponential model with landa = 0.6	MSE will be	224.00680391500347
simple exponential model with landa = 0.65	MSE will be	219.1968671662775
simple exponential model with landa = 0.7	MSE will be	215.51385272273794
simple exponential model with landa = 0.75	MSE will be	212.85204934239167
simple exponential model with landa = 0.8	MSE will be	211.13255905207743
simple exponential model with landa = 0.85	MSE will be	210.2998900819452
simple exponential model with landa = 0.9	MSE will be	210.3208560228981
simple exponential model with landa = 0.95	MSE will be	211.18532288706658
simple exponential model with landa = 1.0	MSE will be	212.9085792450042

اگر نتایج خروجی بالا را به صورت نموداری نشان دهیم چنین شکلی خواهیم داشت.



که مشاهده می‌شود که مقدار لاندای انتخابی ما در واقع بهترین لاندای داده شده است و کمترین میزان خطای پیش‌بینی را ارائه می‌دهد و مشکل اورفیت در مدل‌مان وجود نخواهد داشت.

## مدل هموارسازی نمایی دوبل<sup>12</sup>

مدل هموارسازی نمایی دوبل در پیوست شماره‌ی ۲ به طور کامل توضیح داده شده است و در زیر نتایج استفاده از این مدل برای بررسی داده‌هایمان را آورده ایم.

در مدل هموارسازی نمایی ساده که در بالا تشریح کردیم و نتایج آن را ارائه دادیم، تنها عنصر سطح یا level در بررسی داده‌ها لحاظ می‌شود و اگر روند نیز در داده‌ها موجود باشد ممکن است مدل دچار تاخیر شده و نتواند این افزایش روند را به خوبی تشخیص دهد. در نتیجه در این موارد که داده‌ها دارای شیب افزایشی یا کاهشی هستند و مدل ساده در دنبال کردن این روند دچار مشکل میشود نیاز است تا از مدل‌های مرتبه‌ی ۲ یا دوبل استفاده کنیم.

<sup>12</sup> Double exponential smoothing

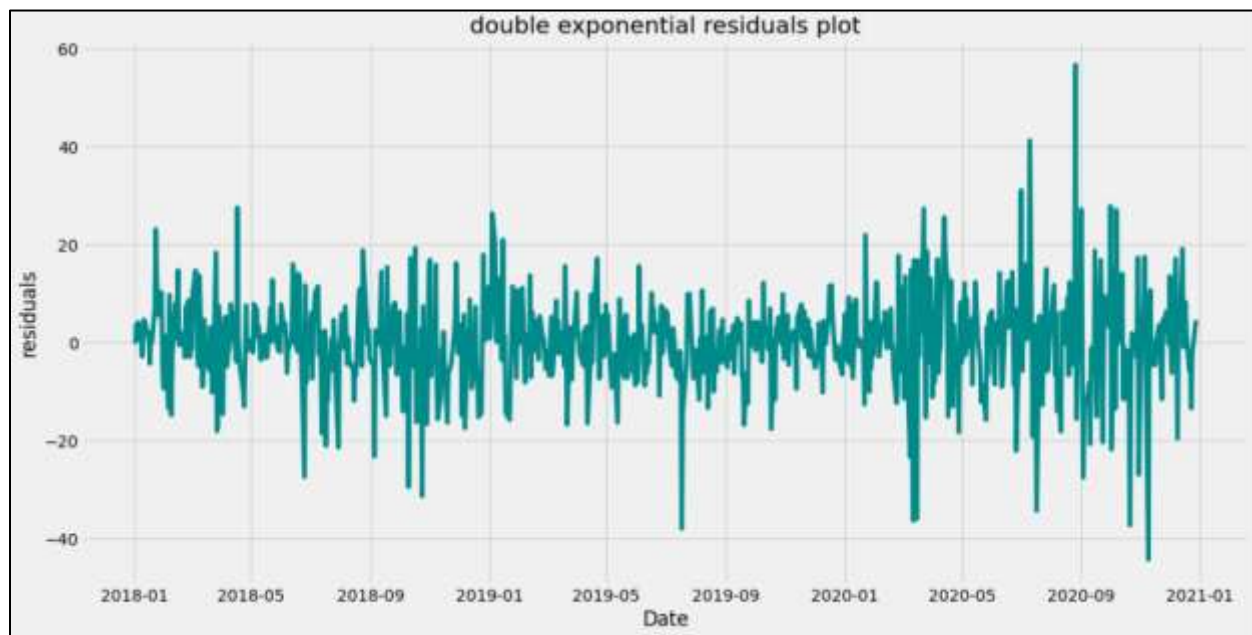
با توجه به نمودار زمانی رسم شده در ابتدای کار روند صعودی را در مشاهدات میتوانستیم مشاهده کنیم. پس از فیت کردن مدل دویل به داده‌های آموزشی پارامترهای مدل به صورت زیر خواهد بود:

ExponentialSmoothing Model Results			
Dep. Variable:	close	No. Observations:	753
Model:	ExponentialSmoothing	SSE	72747.238
Optimized:	True	AIC	3449.723
Trend:	Additive	BIC	3468.219
Seasonal:	None	AICC	3449.835
Seasonal Periods:	None	Date:	Tue, 29 Jun 2021
Box-Cox:	False	Time:	18:37:07
Box-Cox Coeff.:	None		
	coeff	code	optimized
smoothing_level	0.8941686	alpha	True
smoothing_trend	1.1208e-13	beta	True
initial_level	201.03135	l0	True
initial_trend	0.4217996	b0	True

همان‌طور که مشاهده می‌شود مقدار آلفا تقریباً با مدل ساده یکی است و نرخ هموارسازی ترند نیز بسیار کوچک و تقریباً صفر است که نشان می‌دهد روند صعود نمودار تقریباً ثابت است و با نرخ مشخصی بالا می‌رود. مقدارهای ابتدایی مدل نیز که شامل  $\beta_0$  و  $l_0$  هستند نیز به ترتیب ۲۰۱.۰۳۱۳۵ و ۰.۴۲۱۷۹۹۶ تعیین شده‌اند. مجموع مربعات خطای فیت نیز ۷۲۷۴۷ است که در مقایسه با مدل ساده کاهش اندکی داشته است. دیگر شاخص‌های ارزیابی مدل نیز شامل معیار آکایکه نیز در تصویر بالا قابل مشاهده است. نمودار داده‌های آموزشی و مقدار فیت شده را در زیر مشاهده می‌کنید که تقریباً مانند مدل ساده عملکرد مناسبی داشته است:

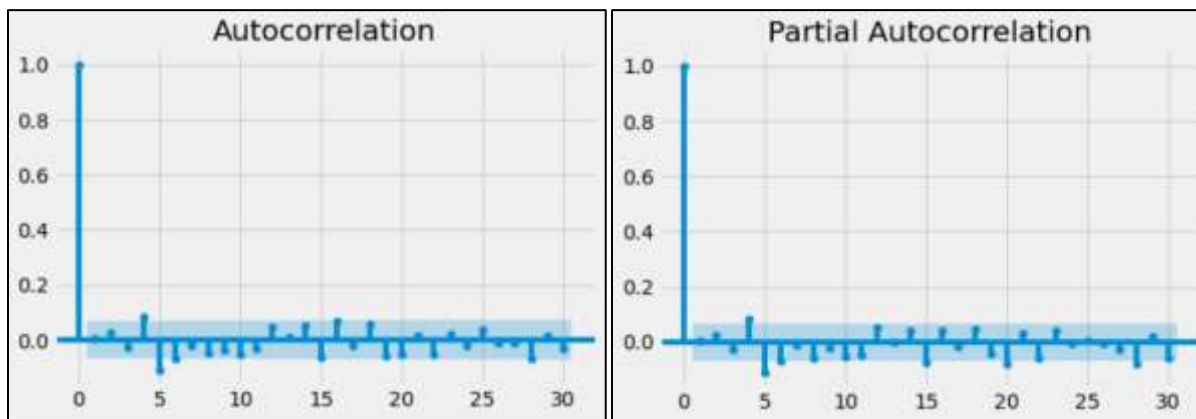


نمودار رزیدوال های شکل بالا را نیز در زیر مشاهده می فرمایید:



تقریباً شکلی مشابه با مدل ساده ارائه می کند و بهبود چندانی نداشته. گمان می توان ثابت نبودن واریانس و افزایش آن را پس از سال ۲۰۲۰ مشاهده کرد. علت این موضوع نیز شیوع پاندمی و ویروس کووید-۱۹ است که سبب اختلال در بازارهای جهانی و افزایش ناپایداری و نوسان در بازارهای مالی گردید.

نمودار های acf و pacf باقی مانده های مدل را در زیر مشاهده می کنید:



همان طور که انتظار می رود نمودار acf و pacf نیز نشان از عدم وجود همبستگی ندارد و به نظر نمی رسد که باقی مانده های مدل مانع از نوین باشند. این اتفاق نشان دهنده این است که با توجه به بهینه بودن مدل و عملکرد مناسب آن بر روی داده های آموزش گمان می رود هموارسازی نمایی ما در پیدا کردن تمام الگوهای موجود در داده ها ناتوان است.

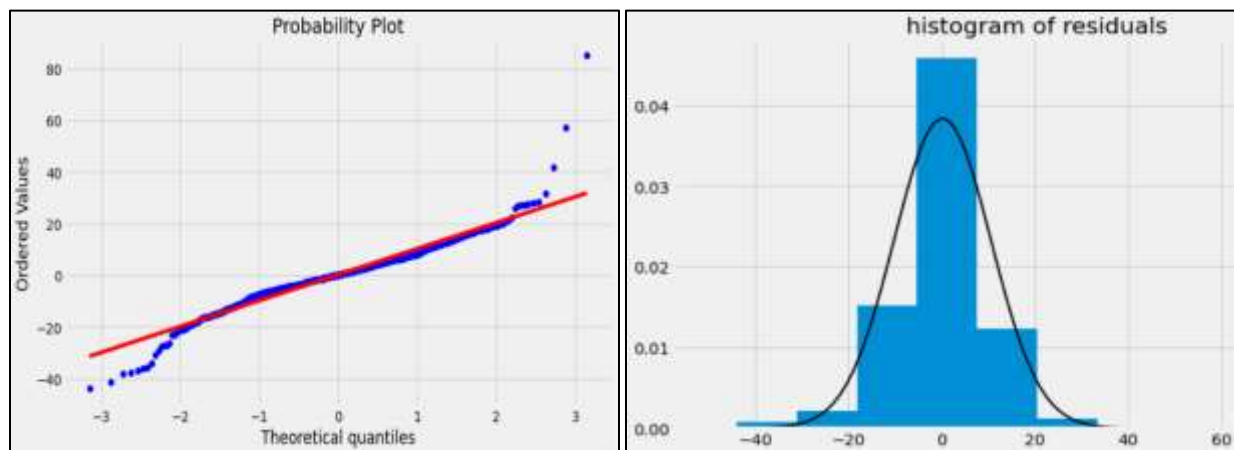
برای اطمینان نتایج تست لیونگ باکس مرتبط با باقی مانده های مدل را در زیر مشاهده می کنید:



	lb_stat	lb_pvalue	bp_stat	bp_pvalue					
1	0.005665	0.940003	0.005645	0.940111	11	27.284858	0.004161	27.027827	0.004551
2	0.523678	0.769635	0.521185	0.770595	12	29.255966	0.003609	28.966005	0.003986
3	1.102657	0.776433	1.096709	0.777869	13	29.332218	0.005871	29.040892	0.006459
4	6.672253	0.154256	6.626427	0.156997	14	31.435976	0.004814	31.104483	0.005360
5	16.656311	0.005200	16.527087	0.005490	15	35.000828	0.002458	34.597017	0.002805
6	20.462097	0.002291	20.296541	0.002452	16	38.992495	0.001090	38.502944	0.001282
7	20.871531	0.003965	20.701577	0.004238	17	39.337891	0.001603	38.840509	0.001881
8	22.762265	0.003683	22.569749	0.003963	18	42.222773	0.001029	41.656539	0.001235
9	23.939556	0.004398	23.731586	0.004746	19	45.216582	0.000639	44.575324	0.000786
10	26.315303	0.003338	26.073313	0.003643	20	47.540206	0.000493	46.837946	0.000618

با توجه به نتایج کاملاً واضح است که مدل چهار همبستگی سریالی است و مدل هموارسازی نمایی دابل ما خوب عمل نکرده است و هنوز الگوی کشف نشده در باقی مانده ها داریم.

دو نمودار توزیع احتمال و هیستوگرام را نیز در زیر مشاهده می کنید که رفتاری مشابه را ارائه می کند و در چارک های انتهایی به دلیل نوسان های ایجاد شده در بازار همپوشانی خوبی را نمیتوان مشاهده نمود.

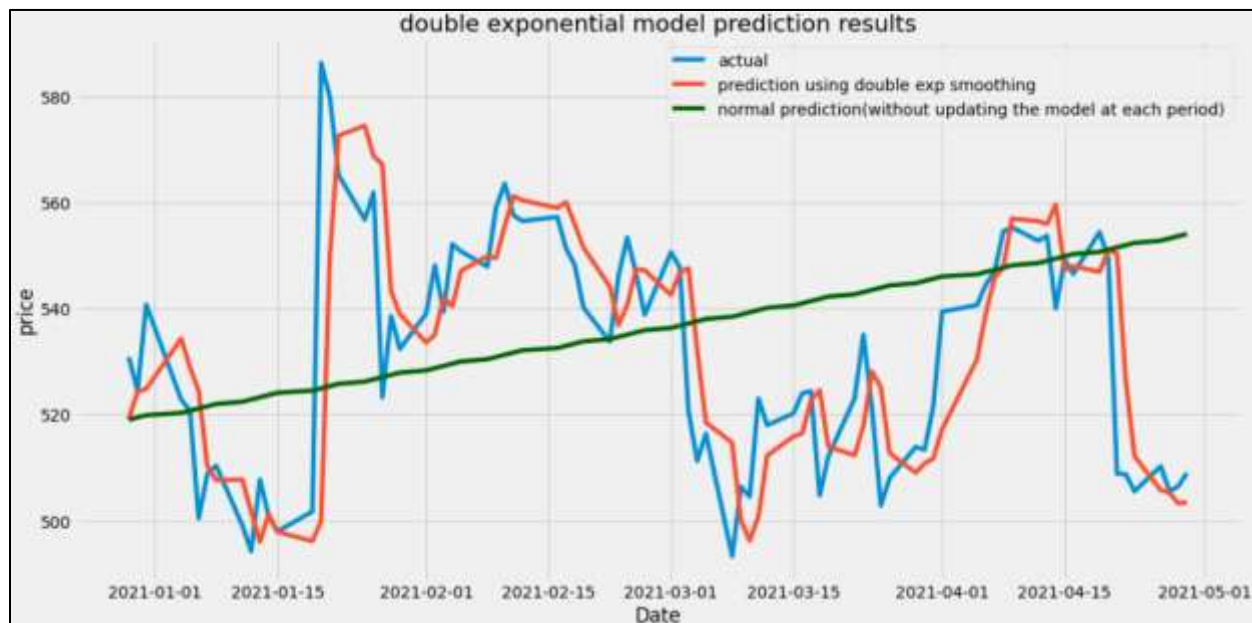


که نشان دهنده این هست هنوز الگوهایی در مدل وجود دارند و باقی مانده های ما هنوز وایت نویز نیستند. برای اطمینان نیز میتوانیم آزمون لینوگ باکس یا باکس پیرس را بر روی باقی مانده ها انجام دهیم که آماره آزمون و  $p$ -value این آزمون را تا لگ ۱۵ ام مشاهده می کنید که باز هم نیز نشان دهنده وجود همبستگی در باقی مانده های مدل مان است.

لازم به ذکر است، زمانی که باقی مانده های مدل ما هم چنان دارای الگو هستند و نوسانات شدیدی نیز دارند این احتمال وجود دارد که کاندید مناسبی برای مدل های آرچ و گارچ باشند و به کمک آن ها بتوان این نوسانات را مدل کرده و ریسک آینده بازار مالی را پیش بینی نمود. در بخش های جلوتر صحبت خواهیم کرد که آیا باقی مانده های مدل مان کاندیدای مناسبی برای مدل گارچ خواهد بود یا نه.

## پیش بینی های حاصل از مدل هموارسازی نمایی دوپل

در نمودار زیر نیز وضعیت مشابه مدل هموارسازی ساده را مشاهده می کنیم. به دلیل نوسانات و ناپایداری شدید داده های مالی تفاوت چندانی را با لحاظ کردن عنصر ترند در مدل نمی بینیم و بهبود در پیش بینی های حاصله بسیار اندک است. نکته ای که در نمودار زیر با مدل ساده تفاوت دارد ، پیش بینی های بدون بروزرسانی آن است که بر خلاف مدل ساده که یک خط صاف بود دارای یک روند خطی است که آن نیز به علت وجود عنصر ترند در مدل می باشد.



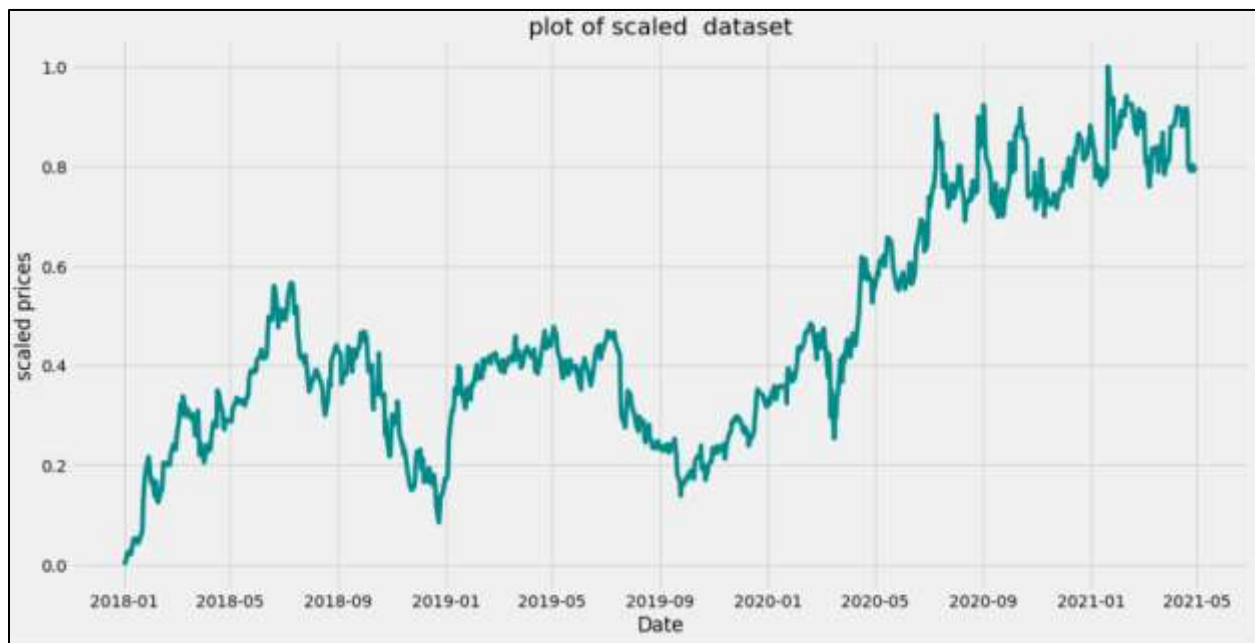
در انتهای این بخش لازم به بیان است که داده های مالی به دلیل ناپایداری بسیار زیاد و روند غیر قابل پیش بینی ای که دارند معمولاً توسط مدل های آرمایی و هموارسازی نمایی مورد استفاده قرار نمی گیرند. و همان طور که بیان شد معمولاً از الگوریتم های بسیار پیچیده تر برای آنالیز معاملات بازار سهام استفاده می شود. به عنوان مثال یکی از آنالیز حساسیت یکی از این موارد است ، بدین شکل که با توجه به اتفاقات و رویدادهای رخ داده نوسانات بازار پیش بینی می شود. هم چنین مدل های ترکیبی نیز کاربرد خوبی دارند به عنوان مثال مدل های آرما-گارچ از این دست هستند. مدل های یادگیری ماشین عصبی و بخصوص شبکه های نیز عملکرد مناسبی در پیش بینی داده های مالی دارند و میتوانند روابط بسیار پیچیده موجود در داده ها را کشف کرده و آن ها را مدل کنند. در ادامه ی پروژه به همین موضوع می پردازیم و به کمک یک مدل شبکه عصبی بازگشتی داده ها را مدل می کنیم.

## مدل شبکه عصبی بازگشتی<sup>۱۳</sup>

شبکه عصبی یکی از الگوریتم‌های یادگیری ماشین است که انواع مختلفی دارد و کاربردهای بسیار گسترده‌ای در صنعت و علم امروزی دارد. در این پروژه ما از نوع خاصی از شبکه‌های عصبی به نام شبکه عصبی بازگشتی استفاده می‌کنیم که کاربرد بسیار گسترده‌ای در پیش‌بینی داده‌ها مالی دارد و به طور عمومی برای داده‌هایی که به صورت یک توالی ارائه می‌شوند کاربرد دارد. توضیحات کامل و مفصل این شبکه عصبی به همراه تصاویر مربوطه در پیوست ۱ آورده شده است. در زیر نتایج حاصله از مدل شبکه عصبی را مشاهده خواهیم کرد.

قبل از هر چیزی باید داده‌های خود را پیش‌پردازش کنیم. ابتدا نیاز است که داده‌هایمان را فشرده کنیم. علت این موضوع نیز آموزش راحت‌تر شبکه عصبی است زیرا مقادیر بزرگ داده سبب کند شدن و سخت شدن یادگیری توسط شبکه عصبی می‌شود. به همین منظور تمام داده‌ها در این قسمت بین بازه ۰ تا یک فشرده شده اند و در انتها و پس از فیت کردن مدل و پیش‌بینی، معکوس تبدیل پیاده‌شده بر روی داده‌ها را اعمال می‌کنیم تا به داده‌های واقعی برسیم.

در زیر نمودار داده‌های فشرده شده را می‌بینیم که دقیقاً همان روند مشاهده می‌شود اما مقادیر قیمت فشرده شده اند:



بر خلاف مدل‌های قبلی، شبکه‌های عصبی نیاز دارند تا داده‌هایمان شامل یک مقدار  $X$  و  $Y$  باشند. بدین معنی که باید توالی از داده‌ها را به عنوان مقدار  $X$  و توالی دیگری را به عنوان  $Y$  در نظر بگیریم. بدین منظور مجموعه داده آموزشی خود را باید تغییر شکل دهیم و با ابعاد متفاوتی وارد شبکه کنیم. در این پروژه ۳۰ داده گذشته را به عنوان  $X$  و داده بعدی پس از این ۳۰ روز را به عنوان  $Y$  در نظر گرفته ایم. یعنی اولین زوج مرتب داده‌های آموزشی جدیدمان شامل قیمت‌های ۳۰ روز گذشته به عنوان  $X$  و قیمت سهام در روز ۳۱ ام به عنوان  $Y$  است. به همین ترتیب برای زوج‌های بعدی یک واحد به جلو می‌رویم و قیمت سهام بین روزهای دوم تا ۳۱

<sup>13</sup> Recurrent neural network



ام به عنوان  $x$  و قیمت سهام در روز ۳۲ ام به عنوان  $y$  لحاظ می‌شود. به عنوان مثال اولین زوج مرتب داده‌های آموزشی مان به شکل زیر است:

```
given the array :  
[0.001033041 0.01183585 0.02315259 0.02849948 0.02138757  
0.02971941 0.04197056 0.05232691 0.05310559 0.0426454 0.0499909  
0.05032834 0.06880887 0.12775452 0.15633187 0.17813482 0.19085316  
0.216783 0.20175455 0.17969211 0.16611726 0.1722428 0.13805899  
0.16780437 0.16479349 0.12726139 0.12562616 0.14763673 0.14846725]  
predict this y :  
[[0.16853113]]
```

بعد از آنکه داده‌ها آماده شد حال نیاز است شبکه عصبی مورد نظر را برای فیت کردن داده‌هایمان طراحی کنیم.

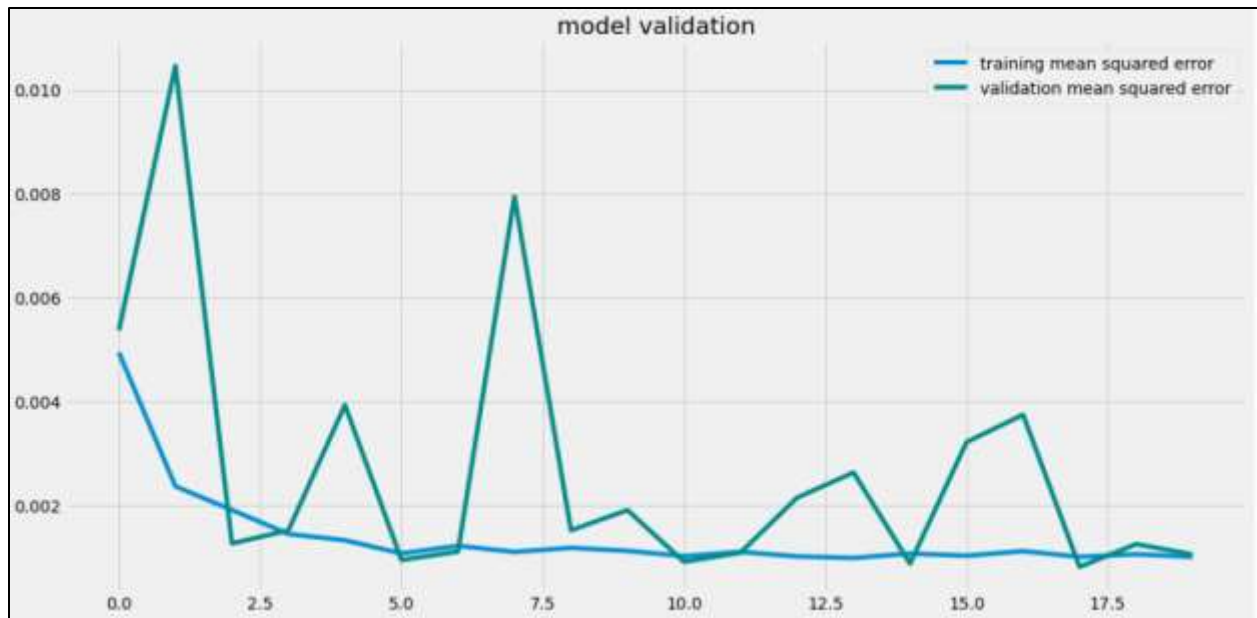
### طراحی شبکه عصبی

هیچ روش قطعی و مسلمی برای تشخیص اینکه دقیقاً چند لایه و در هر لایه چند نورون استفاده شود وجود ندارد. در پروژه پیش رو با توجه مطالعات بر روی مقالات و پروژه‌های مشابه دو لایه LSTM که هر کدام به ترتیب شامل ۵۰ نورون هستند و همچنین یک لایه فولی کانکتد<sup>14</sup> با ۲۵ نورون و در نهایت یک نورون که خروجی ما را مشخص می‌کند، مدل ما را تشکیل می‌دهند. تجربه نشان داده است دو لایه LSTM برای تشخیص روابط پیچیده کافی هستند و بدین منظور نیز در پروژه پیش رو این قاعده نیز رعایت شده است. در اولین لایه LSTM نیز از تابع فعال سازی relu استفاده کردیم که توضیحات آن در پیوست آورده شده است. به منظور بروزرسانی شبکه و وزن‌های موجود از الگوریتم adam که برای داده‌های پیچیده مناسب عمل می‌کند استفاده کرده ایم. در نهایت پس از طراحی شبکه داده‌هایمان را فیت می‌کنیم. لازم به ذکر است که برای این مدل ۱۰ مرتبه شبکه عصبی موجود را با داده‌های آموزشی مان اجرا کردیم. هر چند می‌توانستیم از تعداد دفعات بیشتری استفاده کنیم اما به منظور جلوگیری از اورفیت شدن به این تعداد بسنده کردیم.

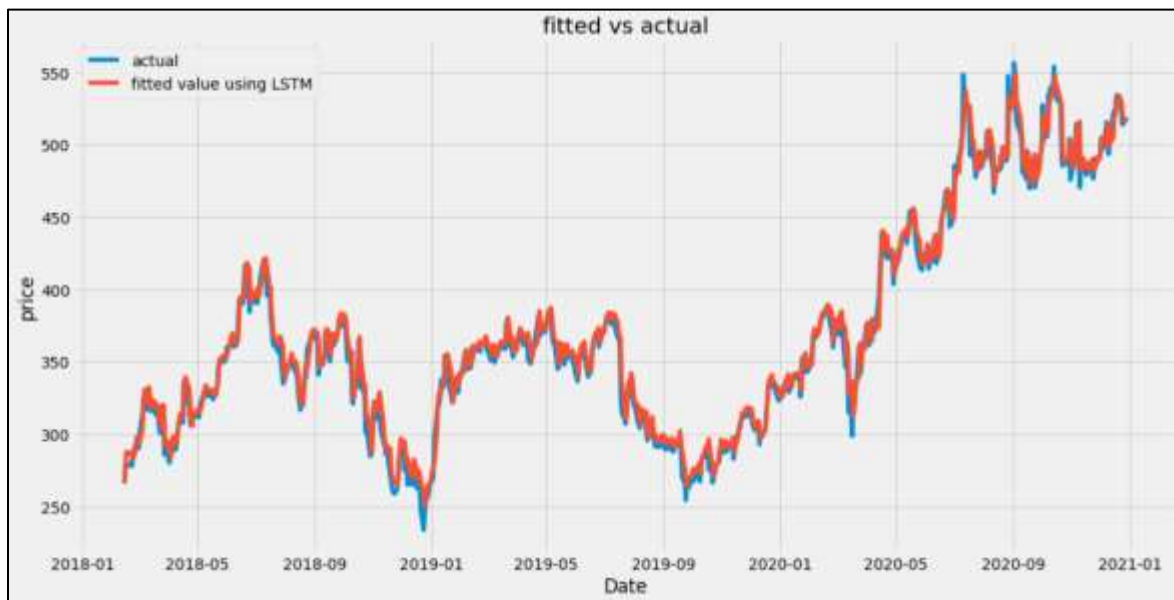
نکته ای که در شبکه‌های عصبی و به طور کلی مدل‌های یادگیری ماشین باید مد نظر داشته باشیم، عدم رخ دادن اورفیت و یا برعکس آن آندرفیت در مدل است. در شبکه‌های عصبی برای بررسی این موضوع و هنگام فیت شدن مدل به داده‌ها دیتاستی را به طور مستقل از داده‌های آموزشی تعریف کرده و نام آن را نیز validation dataset می‌نامیم. پس از هر بار ورود داده‌های آموزشی به شبکه و بروزرسانی مدل، داده‌های دیتاست validation را نیز پیش‌بینی می‌کنیم و مقدار دقت آن را می‌سنجیم. به کمک این روش می‌توانیم وضعیت مدل را در طی فیت شدن دنبال کرده و اگر در جایی روند افزایشی در میزان خطای ارزیابی مدل دیده شود باید نتیجه گرفت مدل دچار اورفیت شده.

در شکل زیر مقادیر training loss و validation loss را مشاهده می‌کنید. برای اینکه مدل اورفیت نباشد باید روند نزولی را در هر دوی این متغیرها مشاهده کنیم بدین معنی که با افزایش دقت یادگیری، دقت پیش‌بینی نیز افزایش بیابد:

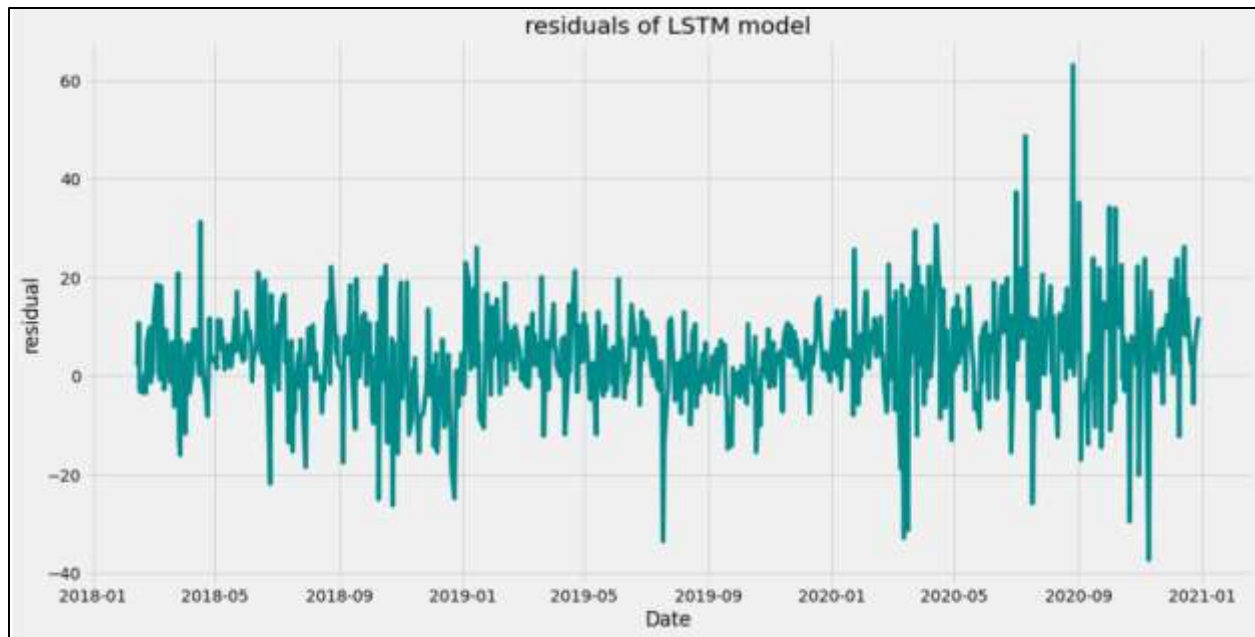
<sup>14</sup> Dense layer or fully connected layer



همان طور که مشاهده می شود همزمان با بالا رفتن دقت یادگیری مدل (خط آبی) دقت پیش بینی نیز افزایش پیدا کرده (خط قرمز) که نشان دهنده این است مدل به دست آمده مشکلی ندارد و می تواند برای پیش بینی داده های آینده مورد استفاده قرار بگیرد. در مدل های یادگیری ماشین معمولاً مقدار باقی مانده و وایت نویز بودنشان ملاک انتخاب مدل نیست ولی از آنجایی که بحث مطالعه ما مدل های بکار رفته در سری زمانی هست در ادامه باقی مانده های مدل شبکه عصبی بالا را نیز مورد بررسی قرار دادیم.

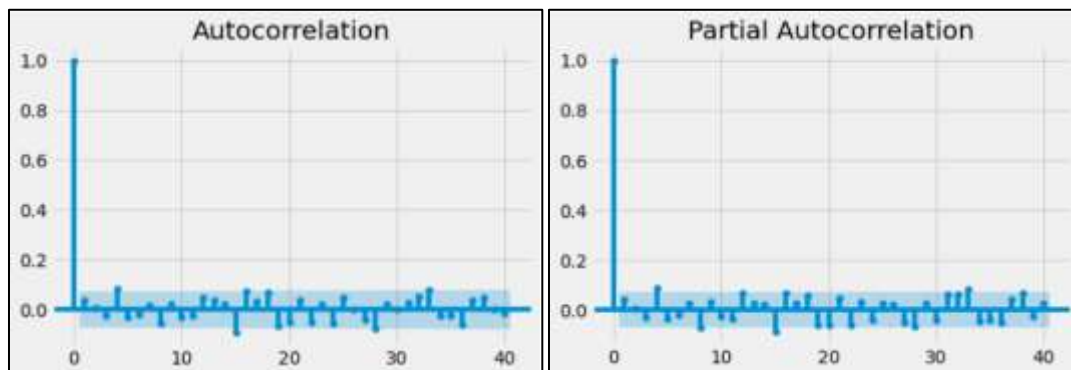


همان طور که مشاهده می شود ، شبکه عصبی به طور مناسبی آموزش دیده است و داده های آموزشی را تا حد خوبی یاد گرفته است. در ادامه باقی مانده های مدل را بررسی می کنیم تا از مناسب بودن مدل حاصله اطمینان حاصل کنیم . مانند گذشته نمودارهای لازم را رسم و بررسی می کنیم:



نمودار رزیدوال‌های مدل را در بالا می‌بینیم که تقریباً رفتاری مشابه مدل‌های قبلی دارد و پس از شروع سال ۲۰۲۰ و گسترش ویروس کرونا و بالا رفتن نوسانات بازار، این موضوع روی باقی‌مانده‌های مدل نیز قابل مشاهده است. میانگین فرایند نیز تقریباً ثابت مانده و ایستا به نظر می‌رسد. حال باید بررسی کنیم که آیا باقی‌مانده‌ها وایت نویز هستند و یا مانند مدل هموارسازی دارای وابستگی می‌باشند.

بدین منظور ابتدا نمودارهای acf و pacf را رسم کرده و مشاهده می‌کنیم:

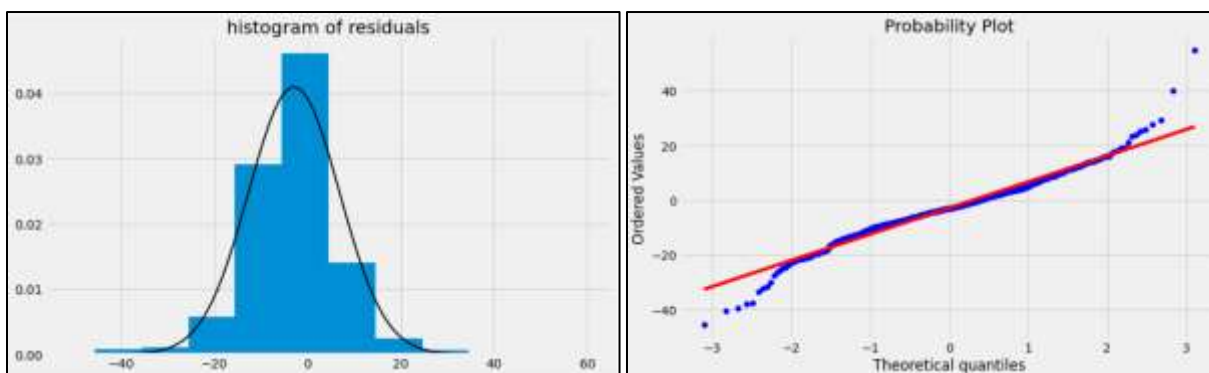


همان طور که مشاهده می‌شود تقریباً و با تقریب خوبی غیر از ۲ یا ۳ مورد در محدوده هستند و این نشان‌دهنده این است که مدل ما عملکرد مناسبی داشته است و تقریباً الگوی موجود در داده‌ها را تشخیص داده است و خطاهای حاصل در پیش‌بینی مدل تقریباً رندم و تصادفی هستند. برای بررسی دقیق‌تر آزمون لیونگ باکس را بر روی باقی‌مانده‌ها اجرا می‌کنیم که نتایج آن را در زیر مشاهده می‌کنید:

	lb_stat	lb_pvalue	bp_stat	bp_pvalue
1	1.145888	0.284412	1.141147	0.285410
2	1.185360	0.552844	1.180401	0.554216
3	1.696086	0.637804	1.687604	0.639693
4	7.034840	0.134059	6.982175	0.136833
5	7.682124	0.174646	7.623210	0.178260
6	7.926190	0.243561	7.864582	0.248193
7	8.188794	0.318241	8.123927	0.321793
8	10.425990	0.236390	10.330265	0.242606
9	10.748032	0.293376	10.647421	0.300652
10	11.295016	0.335000	11.185351	0.343261
11	11.737368	0.383705	11.619772	0.392891
12	13.481439	0.335039	13.330164	0.345507
13	14.607560	0.332488	14.432986	0.344073
14	15.010188	0.377459	14.826728	0.390101
15	20.927042	0.139177	20.604843	0.149937
16	25.011290	0.069627	24.587688	0.077431
17	25.672078	0.080646	25.231159	0.089654
18	29.037584	0.047922	28.503823	0.054791
19	32.435547	0.027900	31.803362	0.032885
20	34.323100	0.024020	33.633638	0.028712

همان طور که مشاهده می‌شود تا لگ ۱۷ مقدار p-value بیش‌تر از 0.05 است که به معنای عدم وجود ارتباط و همبستگی میان این ۱۷ لگ می‌باشد. در نتیجه می‌توان با تقریب خوبی این باقی‌مانده‌ها را وایت نویز دانست هر چند کماکان مشکل ناپایداری واریانس را بعد از سال ۲۰۲۰ مشاهده می‌کنیم که طبیعی است و برای مدل کردن این ناپایداری‌ها نیاز داریم تا از مدل‌های دیگری مانند آرچ و گارچ استفاده کنیم که این مدل‌ها در مبحث تحلیل ریسک در بازارهای مالی کاربرد بسیاری دارند.

در ادامه نیاز داریم توزیع داده‌هایمان را بررسی کنیم تا ببینیم تا چه اندازه به توزیع دلخواه ما یعنی توزیع نرمال نزدیک اند. مانند گذشته نیز از دو نمودار توزیع احتمال نرمال و هیستوگرام برای این منظور استفاده می‌کنیم:



نتایج حاصله تقریباً مانند مدل‌های هموارسازی نمایی بوده و توزیع باقی‌مانده‌های مدل تقریباً نرمال است.

### پیش‌بینی‌های مدل شبکه عصبی بازگشتی

پس از بررسی وضعیت فیت شدن داده‌ها نیاز داریم که پیش‌بینی‌هایمان را به دست آوریم. بدین منظور نیاز داریم که آرایه‌های ۳۰ تایی را به مدل به عنوان ورودی دهیم و با توجه به الگوی کشف شده توسط شبکه عصبی خروجی مد نظر را بگیریم. توجه داشته باشیم که خروجی شبکه عصبی‌مان عددی بین ۰ و یک هست و نیاز داریم تبدیل انجام شده در ابتدای مدل را معکوس کنیم. پس از هر پیش‌بینی که انجام می‌دهیم نیاز داریم که اولین عضو آرایه ورودی را حذف و مقدار واقعی داده پیش‌بینی شده را به آرایه اضافه

می‌کنیم و مجدد این روند تکرار شده تا پیش‌بینی هایمان حاصل شود. در زیر مقادیر پیش‌بینی شده مدل را در برابر مقدار واقعی قیمت سهام مشاهده می‌کنید:



همان طور که مشاهده می‌شود مدل پیش‌بینی های بسیار خوبی ارائه کرده و روند قیمتی را بسیار خوب پیش‌بینی کرده است. هم چنین لگ موجود در پیش‌بینی نسبت به روش‌های هموارسازی نمایی نیز کم‌تر است که نشان از این دارد، مدل شبکه عصبی واکنش سریع تری به تغییرات قیمت نشان می‌دهد. میانگین مربعات خطای پیش‌بینی نیز 202 به دست آمده که نسبت به میانگین فرایندمان مقدار مناسبی است.

## مدل کردن ناپایداری داده‌ها به کمک آرچ<sup>۱۵</sup> و گارچ

تغییر در واریانس یا ناپایداری داده‌ها در طول زمان همان‌طور که در قسمت‌های پیشین بحث شد، سبب بروز مشکلات متعدد در مدل‌های کلاسیک مانند آریما می‌شود. مدل‌های آرچ و گارچ به منظور مدل کردن این ناپایداری‌ها در داده‌های وابسته به زمان توسعه پیدا کردند.

یکی از مشکلات مدل‌های تک‌متغیره مثل مدل‌های آریمایی این است که تغییرات ایجاد شده در واریانس در طول زمان را مدل نمی‌کنند. البته لازم به اشاره است که بعضی داده‌ها که تغییرات واریانس اندکی دارند را میتوان با اعمال تبدیل‌های لگاریتمی پایدار کرد اما در مدل‌های مالی که نمونه آن را در همین پروژه مشاهده کردیم، تغییرات واریانس به صورت پیوسته در طول زمان رخ می‌دهد و نمیتوان با اعمال این تبدیل‌ها واریانس را پایدار نمود. این تغییرات واریانس در مدل‌های مالی ناپایداری<sup>۱۶</sup> نامیده می‌شود.

در سری‌های زمانی، وقتی واریانس به شکل سیستماتیکی مثل تغییر روند یا ترند افزایش پیدا کند، به آن هتروسکدستیسیته<sup>۱۷</sup> می‌گویند. این واژه اصطلاحی در آمار است که به معنای واریانس متغیر در طول سری زمانی می‌باشد. همان‌طور که بیان کردیم اگر تغییر واریانس به طور سیستماتیک رخ دهد و بتوان همبستگی‌ای را میان واریانس داده‌ها در بازه‌های زمانی متفاوت برقرار نمود در نتیجه میتوان از مدل‌هایی برای پیش‌بینی واریانس در آینده استفاده کرد. یک دسته از این مدل‌ها که بسیار مشهور هستند مدل‌های آرچ و گارچ نام دارند.

## مدل آرچ

به مدلی گفته می‌شود که فرض بر این دارد که واریانس عبارات خطا یک تابع از اندازه این عبارات در دوره‌های زمانی قبل است: معمولاً واریانس با مربع خطاهای دوره‌های قبلی مرتبط است. چنین مدلی آرچ نامیده می‌شود. مدل گارچ دارای یک پارامتر لگ است که نشان‌دهنده تعداد باقی‌مانده‌های<sup>۱۸</sup> در نظر گرفته شده از پریودهای گذشته در مدل است. معمولاً پیشنهاد می‌شود مدل آرچ بر روی باقی‌مانده‌های مدل پیاده شده استفاده شود. به عنوان مثال باقی‌مانده‌های مدل آریما.

مشخصات مدل آرچ را در زیر مشاهده می‌کنیم:

اگر  $\varepsilon_t$  نشان‌دهنده خطا در پریود  $t$  باشد و فرض شود  $\varepsilon_t = \sigma_t z_t$  در حالی که  $z_t \sim N(0,1)$ ، سری  $\sigma_t^2$  به صورت زیر مدل می‌شود:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_p \varepsilon_{t-p}^2 = \alpha_0 + \sum_{i=1}^q \alpha_i \varepsilon_{t-i}^2$$

که در آن  $\alpha_0 > 0, i > 0, \alpha_i \geq 0$

<sup>15</sup> ARCH or Autoregressive Conditional Heteroskedasticity method

<sup>16</sup> volatility

<sup>17</sup> heteroskedasticity

<sup>18</sup> Residual errors



## مدل گارچ

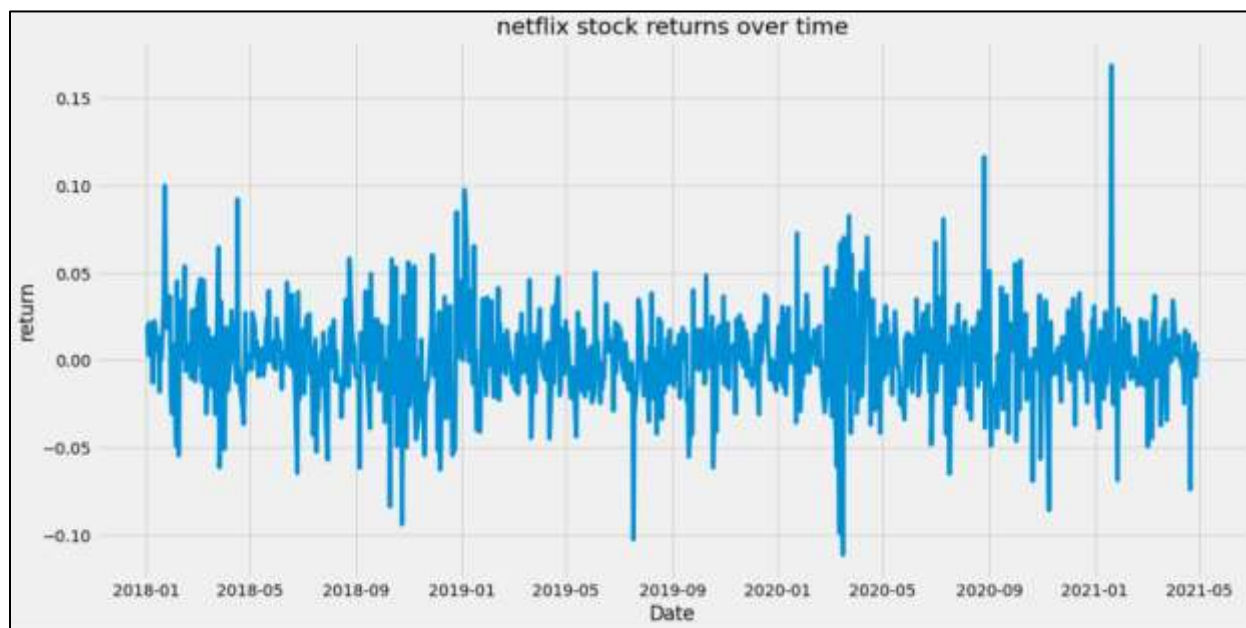
اگر مدل آرما را برای واریانس خطاها فرض بگیریم، مدل گارچ را خواهیم داشت. در این حالت مدل  $\text{garch}(p,q)$  که در آن  $p$  مرتبه  $\sigma^2$  در مدل  $\text{garch}$  و  $q$  مرتبه  $\varepsilon^2$  در این مدل نشان می‌دهد و به صورت زیر نشان داده می‌شود:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_p \varepsilon_{t-p}^2 + \beta_1 \sigma_{t-1}^2 + \dots + \beta_p \sigma_{t-p}^2 = \alpha_0 + \sum_{i=1}^q \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^p \beta_i \sigma_{t-i}^2$$

اگر در مدل گارچ  $p$  برابر با ۰ قرار داده شود به همان مدل  $\text{arch}$  با مرتبه  $q$  می‌رسیم. نکته ای که قبل از پیاده سازی این مدل‌ها بر روی داده‌های خود باید توجه کنیم این است که فرض ایستایی داده‌ها باید برقرار باشد و روند صعودی، نزولی یا فصلی در آن مشاهده نشود.

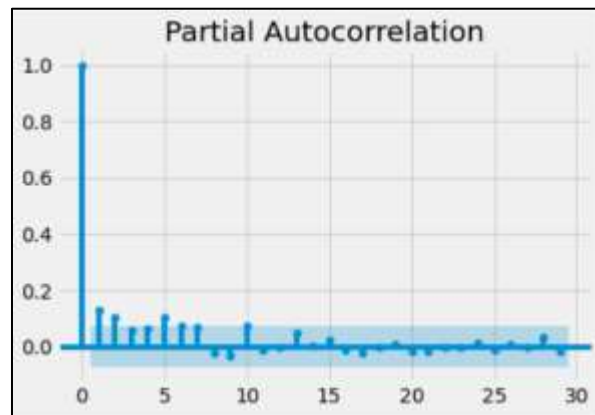
## فیت کردن مدل گارچ

مدل‌های آرچ و یا گارچ برای پیش‌بینی واریانس و ناپایداری مدل استفاده می‌شوند و بهتر است بر روی داده‌هایی پیاده شوند که میانگین ثابت و دارای نوسانات زیاد باشند. در صورتی که باقی‌مانده‌های مدل‌مان دارای نوسان زیادی باشند می‌توان مدل‌های آرچ و گارچ را بر روی آن پیاده کرد تا پیش‌بینی مناسبی از ناپایداری در آینده به دست آوریم. در این بخش از پروژه مدل گارچ را بر روی مقدار بازگشت<sup>۱۹</sup> یا افزایش قیمت سهام نسبت به روز گذشته پیاده می‌کنیم. علت این موضوع نیز این است که نرخ بازگشت به خوبی تغییرات در داده‌ها و نوسانات رخ داده را نشان می‌دهد و کاندیدای مناسبی برای مدل گارچ می‌باشد. در شکل زیر نمودار بازگشت سهام را مشاهده می‌کنید:

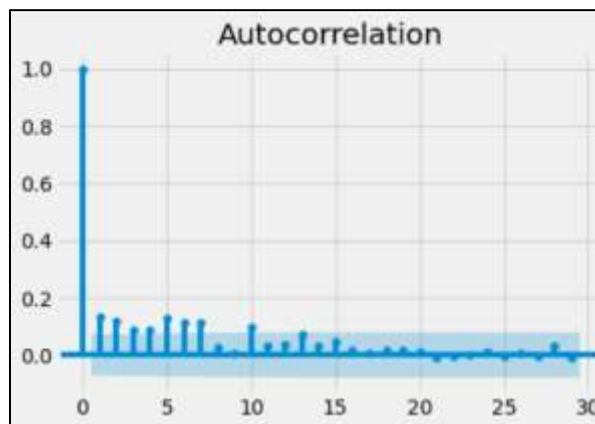


<sup>19</sup> return

همان طور که مشاهده می‌کنید به وضوح قابل مشاهده است که در برخی بازه‌ها واریانس کم است که به معنی پایداری روند حرکت قیمت‌ها است و در برخی بازه‌ها به خصوص بعد از سال ۲۰۲۰ این واریانس در برخی نقاط بسیار زیاد است که نشان از ناپایداری مدل دارد. در مدل گارچ پیشنهادی که در زیر پیاده می‌کنیم سعی می‌کنیم تا ناپایداری‌های داده‌هایمان را در آینده پیش بینی کنیم. مانند قسمت‌های قبل داده‌هایمان را به دو دسته train و test تقسیم می‌کنیم. برای تشخیص مرتبه‌ی  $p$  در مدل‌مان نیاز داریم تا  $pacf$  مربع داده‌هایمان را رسم کنیم. علت این موضوع نیز ساختار مدل گارچ است که توضیحات آن را در بالا ارائه کردیم. پس از رسم نمودار  $pacf$  چنین شکلی حاصل خواهد شد:



همان طور که می‌بینیم الگوی واضحی وجود ندارد و بعد از دو لگ اول و دوم که خارج از حدود کنترلی هستند، لگ‌های بعدی با کمی چشم‌پوشی در حدود مشخصه قرار دارند پس به نظر می‌رسد که مقدار ۱ یا ۲ برای پارامتر  $p$  در مدل کافی باشند. نمودار  $acf$  مربع داده‌ها را نیز در زیر مشاهده می‌کنیم که نشان‌دهنده وجود همبستگی سریالی در داده‌ها است.



با توجه به بررسی‌های انجام شده در نرم‌افزار و مقایسه معیار  $aic$  و  $log-likelihood$  و همچنین میزان تاثیرگذاری پارامترها در مدل با توجه به  $p$ -value داده شده، مدل  $garch(1,1)$  بهترین عملکرد را در برازش داده‌ها نشان می‌دهد. در تصویر زیر خلاصه پارامترهای حاصله از فیت شدن مدل  $garch(1,1)$  را به داده‌هایمان مشاهده می‌کنید:



Constant Mean - GARCH Model Results

Dep. Variable:	close	R-squared:	0.000
Mean Model:	Constant Mean	Adj. R-squared:	0.000
Vol Model:	GARCH	Log-Likelihood:	1676.72
Distribution:	Normal	AIC:	-3345.45
Method:	Maximum Likelihood	BIC:	-3326.95
		No. Observations:	753
Date:	Mon, Jul 05 2021	Df Residuals:	752
Time:	20:41:07	Df Model:	1

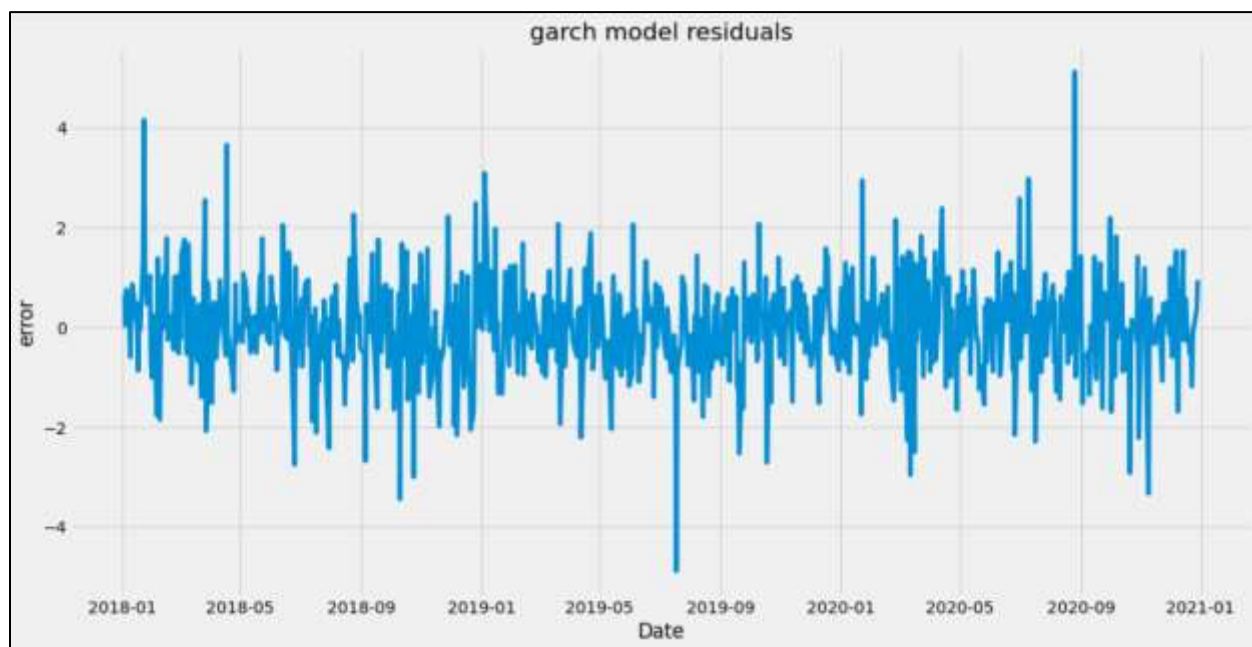
Mean Model

	coef	std err	t	P> t	95.0% Conf. Int.
mu	1.8623e-03	8.775e-04	2.122	3.380e-02	[1.426e-04, 3.582e-03]

Volatility Model

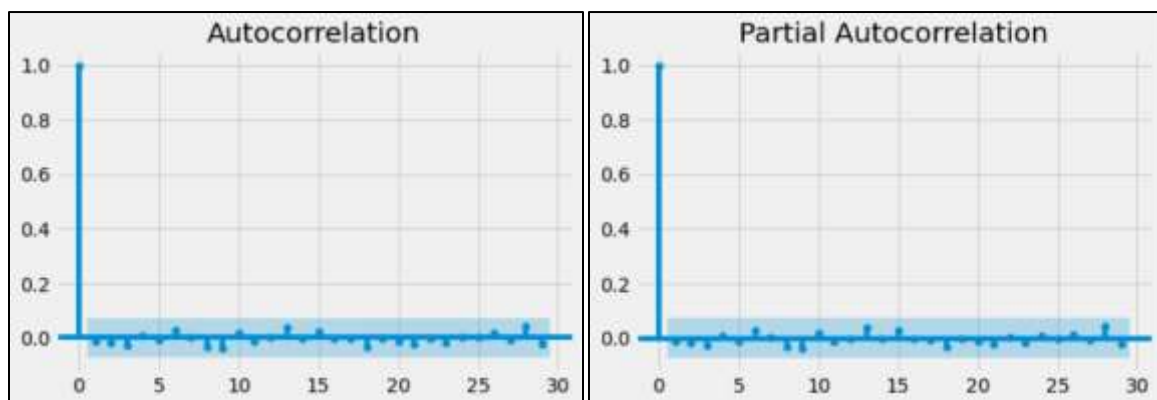
	coef	std err	t	P> t	95.0% Conf. Int.
omega	6.8092e-05	1.426e-05	4.776	1.792e-06	[4.015e-05, 9.604e-05]
alpha[1]	0.0904	3.056e-02	2.958	3.101e-03	[3.049e-02, 0.150]
beta[1]	0.8196	3.292e-02	24.898	7.877e-137	[0.755, 0.884]

همان طور که مشاهده می‌شود هر ۳ پارامتر مدلمان قابل توجه هستند. برای مدل کردن میانگین فرایند نیز از آنجا که میانگین فرایند تقریباً ثابت است از مدل ثابت<sup>20</sup> استفاده کردیم. در ادامه نیاز داریم تا باقی‌مانده های مدل گارچمان را نیز بررسی کنیم که در زیر نمودار باقی‌مانده‌های مدل را مشاهده می‌کنیم:



<sup>20</sup> Constant

همان طور که به نظر می‌رسد باقی‌مانده‌ها تقریباً توزیع مناسبی دارند و واریانس نیز در ثابت به نظر می‌رسد. در نتیجه از روی شکل بالا میتوان تا حد خوبی وایت نویز بودن باقی‌مانده‌ها را نتیجه گرفت اما برای بررسی بیشتر نمودارهای acf و pacf مربع باقی‌مانده‌ها را رسم کردیم. علت اینکه مربع باقی‌مانده‌ها را در مدل گارچ بررسی میکنیم نیز به دلیل ساختار مدل آن است که در بالا تشریح کردیم.

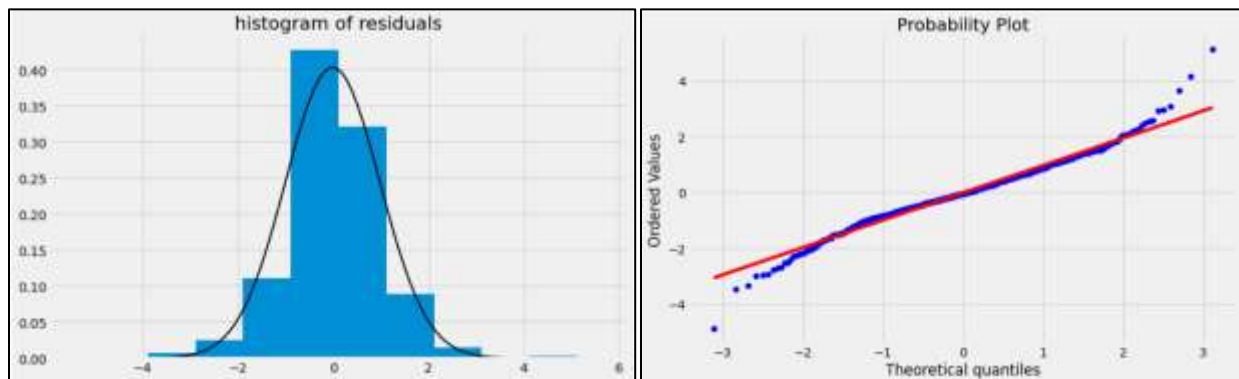


همان طور که مشاهده می‌شود ، نمودارهای بالا نیز گواه عدم وجود همبستگی در باقی‌مانده‌های مدل گارچمان هستند. برای بررسی بیشتر نیز میتوانیم از آزمون لیونگ باکس نیز بهره ببریم. با توجه به شکل‌های بالا باید انتظار مقادیر p-value بالای ۰.۰۵ را در تمام لگ‌ها داشته باشیم. نتایج این آزمون به ترتیب زیر می‌باشد:

	lb_stat	lb_pvalue			
			16	5.422944	0.993217
1	0.089164	0.765242	17	5.429618	0.996271
2	0.338895	0.844131	18	6.360562	0.994502
3	0.894277	0.826809	19	6.362259	0.996915
4	0.958525	0.916018	20	6.457166	0.998122
5	1.030785	0.960045	21	6.854146	0.998411
6	1.682806	0.946441	22	6.855217	0.999136
7	1.694705	0.974794	23	7.076043	0.999400
8	2.589744	0.957415	24	7.080089	0.999680
9	3.698184	0.930132	25	7.080090	0.999834
10	3.978198	0.948325	26	7.278277	0.999889
11	4.081291	0.967478	27	7.311909	0.999941
12	4.081304	0.981924	28	8.800810	0.999799
13	5.026737	0.974600	29	9.149508	0.999837
14	5.027133	0.985431	30	9.698585	0.999836
15	5.412351	0.988022			

همان طور که مشاهده می‌شود مقادیر p-value بالا و نزدیک به ۱ هستند و این موضوع نشان دهنده عدم وجود همبستگی را در باقی‌مانده‌های مدل اثبات می‌کند.

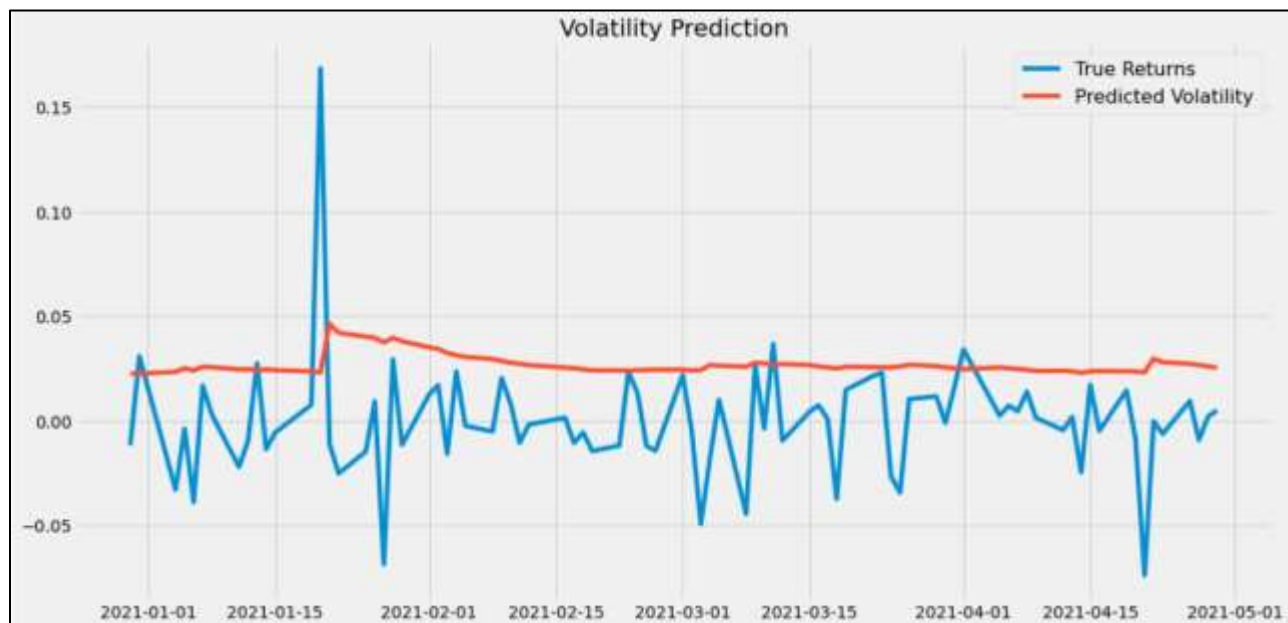
دیگر موردی که باید بررسی شود توزیع نرمال باقی مانده های مدل است که به کمک هیستوگرام و نمودار توزیع احتمال بررسی کرده ایم. نتایج آن را در زیر مشاهده می کنیم:



با توجه به نمودارهای بالا نیز میتوان نتیجه گرفت که باقی مانده تقریباً همپوشانی مناسبی با توزیع نرمال دارند و بدین صورت میتوان  $garch(1,1)$  برازش بسیار خوبی بر روی داده های ما داشته و میتوانیم از آن برای پیش بینی ناپایداری داده هایمان در آینده استفاده کنیم.

### پیش بینی ناپایداری تغییرات قیمت به کمک گارچ

در شکل زیر مقدار نرخ بازگشت در ۸۴ پریود بعدی و هم چنین میزان تخمین مدل گارچ از ناپایداری را مشاهده می کنیم که میتوان دید عملکرد مناسبی داشته و نوسان ایجاد شده در ژانویه ۲۰۲۱ را مدل کرده است و در ادامه نیز با توجه به آنکه واریانس ثابت است روند ثابتی را از پیش بینی های مدل مشاهده می کنیم.



در انتها نیز لازم با تاکید است که مدل گارچ برای پیش بینی مقدار قیمت مورد استفاده قرار نمی گیرد و بیشتر در مباحث ارزیابی ریسک<sup>۲۱</sup> کاربرد دارد تا بتوان تخمینی از وضعیت ناپایداری یا بی ثباتی بازار اعلام کرد. در نتیجه اگر مدل پیش بینی روزی ناپایدار را داشته باشد بهتر است در آن روز سرمایه گذاری ای انجام نشود زیرا احتمالاً ریسک بالایی خواهد داشت.

## نتیجه گیری

به طور خلاصه در این پروژه برای پیش بینی مقدار قیمت سهام از ۳ مدل استفاده کردیم و از مدل آرچ و گارچ نیز برای مدل کردن تغییرات قیمت و نوسان های موجود بهره بردیم. اما کدام مدل و روش بهترین روش برای پیش بینی قیمت سهام است؟ ساده ترین روش برای پاسخ به این سوال نگاه کردن به مقدار میانگین مجموع مربعات خطای مدل هاست که به ترتیب زیر در این پروژه حاصل شدند:

مدل  $arima(0,1,5)$  : ۲۰۵

مدل  $simple\ exponential\ smoothing$  : ۲۱۰

مدل  $double\ exponential\ smoothing$  : ۲۱۵

مدل شبکه عصبی LSTM : ۲۰۲

با توجه به نتایج حاصل شده به نظر می رسد مدل شبکه عصبی نتایج دقیق تری را حاصل کرده است و میتوان آن را به عنوان مدل برتر دانست.

در انتها لازم به بیان است که نمیتوان هیچ حکم مسلمی درباره اینکه مدل شبکه عصبی همواره بهتر عمل می کند نداریم و در شرایط متفاوت و داده های متفاوت ممکن است نتایج دیگری حاصل شود. همان طور که در بخش مرور ادبیات مطرح کردیم در برخی موارد مدل های کلاسیک مانند آریما و در برخی دیگر مدل های مدرن شبکه عصبی نتایج بهتری دارند. هم چنین مدل های ترکیبی نیز در برخی موارد نتایج بسیار ارزشمندی حاصل می کنند. به عنوان مثال در داده هایی که نوسانات بسیار شدید داریم و واریانس این نوسانات به طور دائم در حال تغییر است ، مدل  $arma-garch$  یا  $arima-garch$  گزینه های خوبی برای بررسی چنین مدل هایی هستند.

<sup>21</sup> Risk assesment

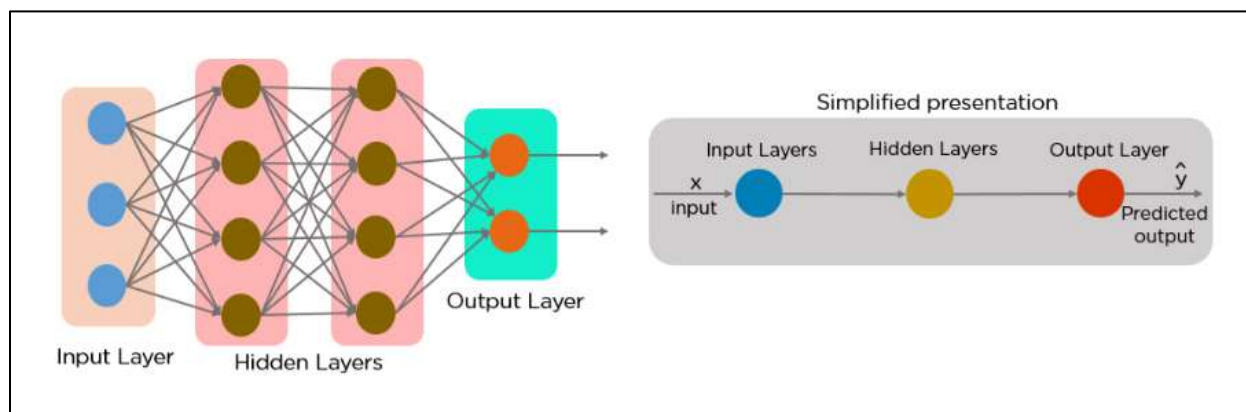
## پیوست ۱

### شبکه عصبی بازگشتی<sup>۲۲</sup>

قوی ترین الگوریتمی که برای تحلیل داده‌های خود استفاده می‌کنیم ، شبکه‌های عصبی بازگشتی هستند که یکی از الگوریتم های یادگیری ماشین می‌باشد. شبکه‌های عصبی بازگشتی کاربردهای بسیاری در زمینه‌های مختلفی همچون ترجمه متون ، کامل کردن جملات ( natural language process ) ، لیبل گذاری بر روی تصاویر ، تحلیل سری‌های زمانی و بسیاری کاربردهای دیگر می‌باشد. این نوع شبکه عصبی بسیار قدرتمند بوده و می‌تواند پیش بینی‌هایی با دقت بسیار بالا ارائه کند. قبل از اینکه به استفاده از این الگوریتم برای تحلیل داده‌های خود بپردازیم نیاز است تا توضیح مختصری راجع به این الگوریتم بدهیم. این توضیحات را در قالب چند سوال که در ادامه آورده شده اند مشاهده می‌کنید:

### شبکه عصبی چیست؟

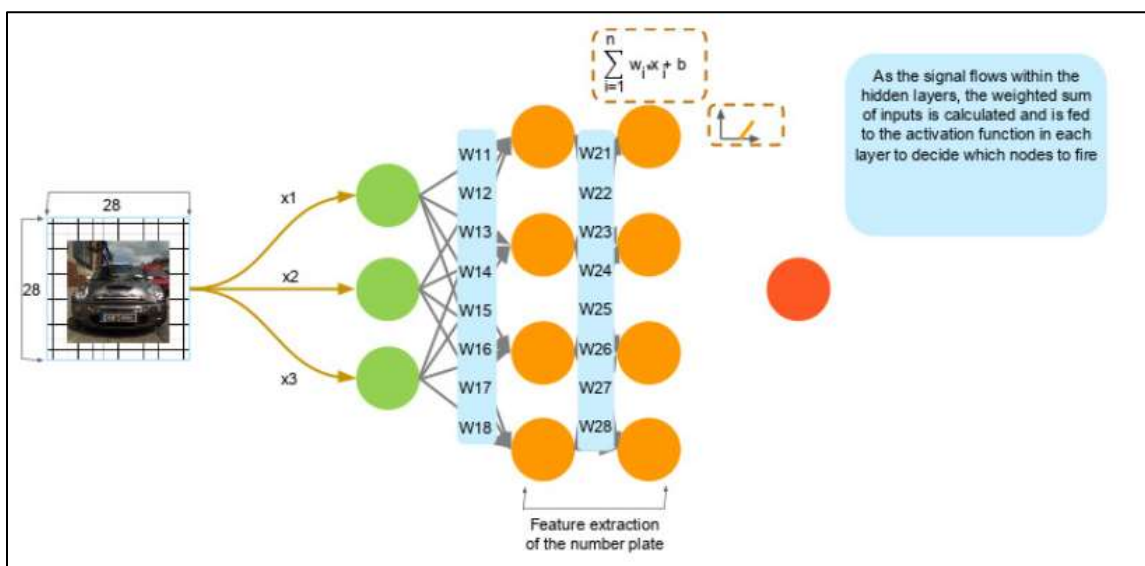
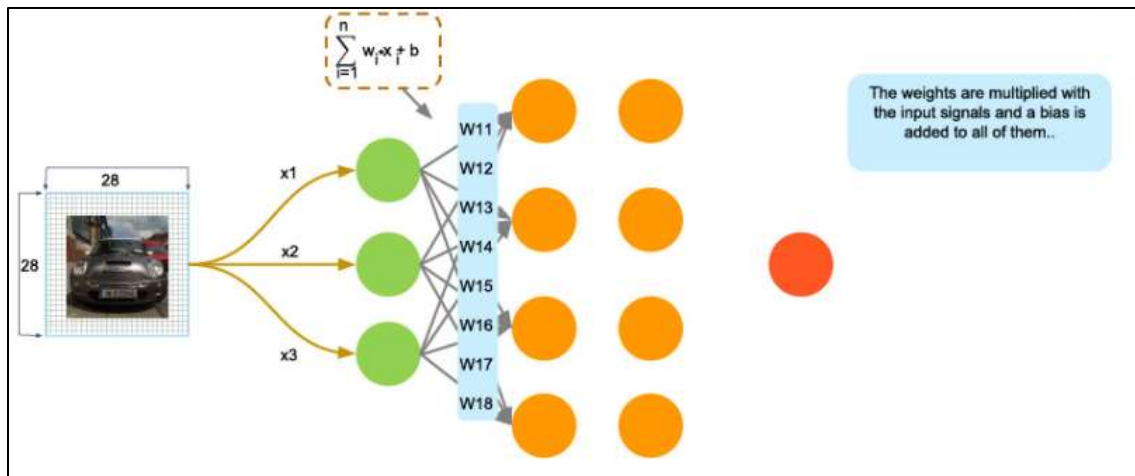
شبکه عصبی از تعدادی لایه متصل به یکدیگر تشکیل شده است و شیوه کار آن از مغز انسان الهام گرفته شده است. اولین لایه شبکه عصبی لایه ورودی است. لایه یا لایه های بعدی که به لایه‌های پنهان نیز مشهور است عمل محاسبات و استخراج ویژگی داده‌های ورودی را بر عهده دارد و در نهایت لایه خروجی جواب نهایی را مشخص می‌کند. در زیر شمایی کلی از شبکه عصبی مشاهده می‌کنید:



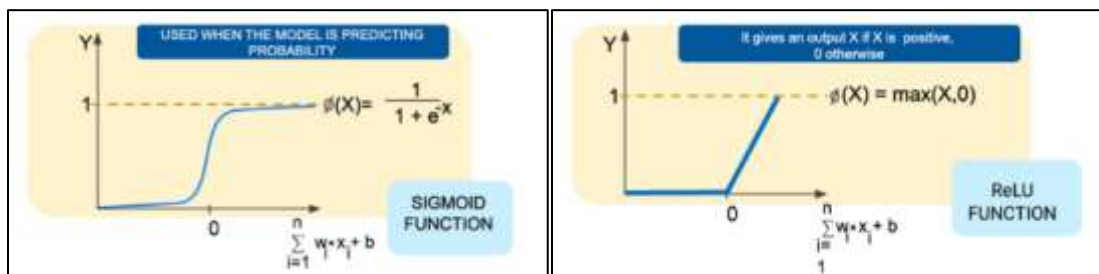
به عنوان مثال میتوان شبکه عصبی ای را مثال زد که تصاویر یک دوربین کنترل سرعت ورودی‌های آن ، و شماره پلاک خودرو خروجی آن است. همان طور که در شکل زیر مشاهده می‌کنید تصویر مدنظر شامل ۲۸ در ۲۸ پیکسل است. هر نورون ورودی شبکه عصبی مطابق با یک پیکسل از این تصویر است. هر نورون عددی بین ۰ تا یک را میگیرد که ۰ به معنای سفید و ۱ به معنای سیاه است. این اطلاعات در وزن رابطهای موجود که به صورت رندوم تعیین شده است ضرب شده و سپس با جمع با عدد ثابتی که bias نام دارد وارد لایه مخفی می‌شود. در واقع جمع وزنی داده‌های ورودی به عنوان ورودی برای یک تابع فعال‌سازی<sup>۲۳</sup> در نظر گرفته شده تا تصمیم گرفته شود کدام پیکسل‌ها برای استخراج ویژگی‌ها استفاده شوند.

<sup>۲۲</sup> Recurrent neural network (RNN)

<sup>۲۳</sup> Activation function



تابع های فعال سازی مختلفی برای شبکه عصبی توسعه داده شده است که تابع سیگموئید<sup>۲۴</sup>، تابع آستانه<sup>۲۵</sup> و تابع واحد خطی<sup>۲۶</sup> اصلاح شده از مهم ترین آنان است که دوتای آن ها را در زیر مشاهده می نمایید:



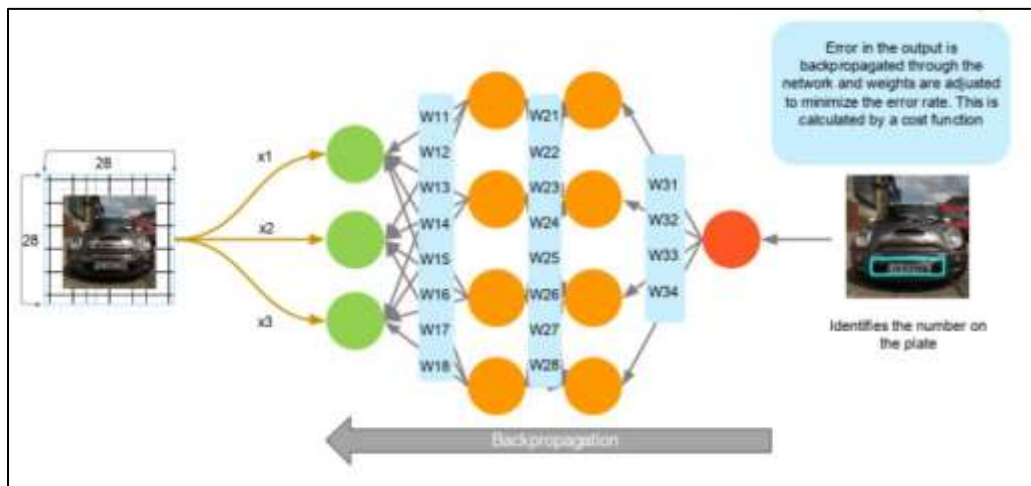
<sup>24</sup> Sigmoid function

<sup>25</sup> Threshold function

<sup>26</sup> Rectified linear unit (relu)



حال پس از آنکه به کمک لایه‌های مخفی و توابع فعال‌سازی جواب نهایی حاصل شد خطای حاصله محاسبه شده و به کمک الگوریتم‌های مشخصی وزن‌های موجود در شبکه عصبی را اصلاح می‌کنیم تا جواب‌های دقیق‌تری حاصل شود که به این عمل back-propagation می‌گویند.



با هر تکرار و آموزش شبکه عصبی این وزن‌ها آپدیت شده تا به دقت مدنظر برسیم. اما مطالب مطرح شده خلاصه و کلیتی از شبکه‌های عصبی بود. شبکه‌های عصبی مختلفی با نحوه کار مختلفی وجود دارد که پایه‌ای‌ترین مدل آن را در بالا توضیح دادیم و **Feed-forward Neural Network** نام دارد. داده‌ها در این شبکه در یک جهت حرکت می‌کنند. مزیت این شبکه عصبی سریع بودن آن است هرچند آموزش آن کمی زمان می‌برد.

### چرا شبکه عصبی بازگشتی؟

شبکه‌های عصبی بازگشتی به دلیل بعضی از ناکارآمدی‌ها در شبکه‌های عصبی معمولی یا feed forward توسعه یافتند. این کاستی‌ها عبارتند از:

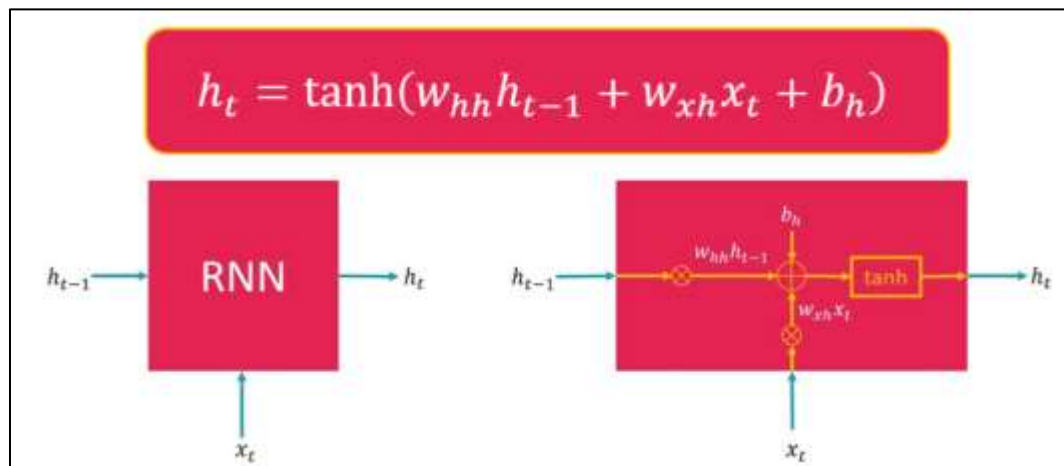
- عدم کارایی برای داده‌های سری زمانی و توالی‌های عددی
- در نظر نگرفتن خروجی‌های گذشته در تحلیل داده‌های آینده
- عدم به یادآوری ورودی‌های گذشته

### انواع شبکه‌های عصبی بازگشتی

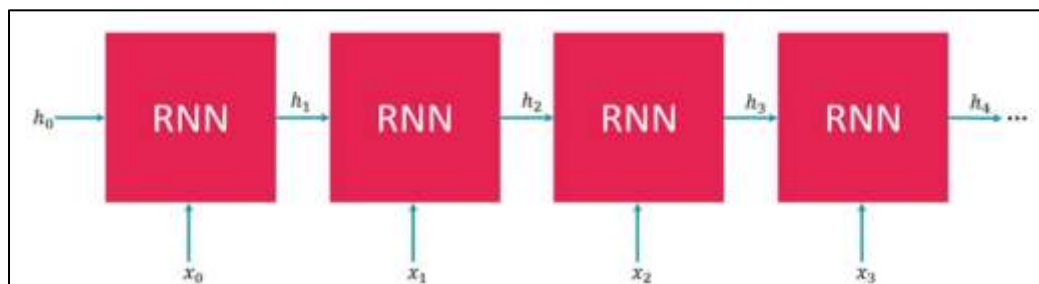
شبکه‌های عصبی بازگشتی خود نیز بر حسب تعداد ورودی و خروجی به ۴ دسته یک به یک، یک به چند، چند به یک و چند به چند تقسیم می‌شوند.

### شیوه کار شبکه عصبی بازگشتی

شبکه عصبی بازگشتی شامل دو ورودی حافظه و ورودی اصلی است. به ورودی حافظه hidden state گفته می‌شود و معمولاً با نماد  $h$  نشان داده می‌شود. ورودی اصلی هم با  $x$ . هر یک از دو ورودی به یک لایه MLP متصل شده است. دو لایه  $w_{xh}$  و  $w_{hh}$  را به ترتیب برای ورودی  $h$  و  $x$  در نظر می‌گیریم. حالا باید  $h_{t-1}$  و  $x_{t-1}$  به عنوان ورودی وارد RNN شده و در دو ماتریس وزن  $w_{xh}$  و  $w_{hh}$  ضرب شوند. طبق رابطه نشان داده شده در شکل زیر، پس از ضرب‌ها، جمع انجام می‌شود و با یک بایاس جمع می‌شود. در نهایت، خروجی از یک تابع تحریک غیرخطی مانند  $\tanh$  عبور داده می‌شود. خروجی تابع تحریک، همان  $h_t$  است. در رابطه و شکل زیر ساختار RNN نشان داده شده است. در شکل سمت راست، جزئیات شبکه بازگشتی نشان داده شده است.



خروجی بالا معادل Hidden state بعدی ( $h_t$ ) و خروجی شبکه RNN در زمان  $t$  است. حالا از اتصال زنجیره‌ای این یک شبکه به هم، شکل زیر به وجود می‌آید.



شبکه های عصبی بازگشتی برای پیش بینی توالی های عددی ، پیش بینی بازار سهام ، ترجمه متون و تولید متون کارایی بسیار دارد اما مشکلی که در این شبکه ها وجود دارد آموزش سخت آن است. علت این موضوع نیز مشکلی رایج در این شبکه به نام گریادیان ناپدیدار<sup>۲۸</sup> است. این موضوع سبب آن می‌شود که لایه های ابتدایی شبکه تغییر خاصی نکنند و این موضوع سبب می‌شود که شبکه داده های قدیمی تر را فراموش کند. این مشکل حافظه کوتاه مدت نیز نام دارد.

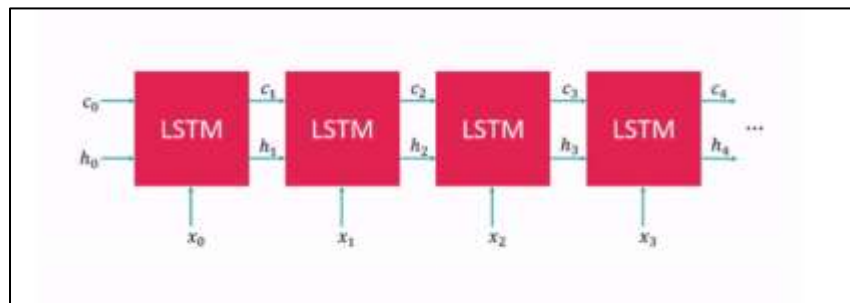
برای حل مشکل حافظه شبکه های عصبی بازگشتی ، مدل دیگر شبکه عصبی که LSTM نام دارد توسعه یافت که در این پروژه ما از این شبکه استفاده می‌کنیم. شبکه های LSTM در پیش بینی داده های سری زمانی با حجم بالا کاربرد گسترده ای در صنعت دارند



که پیش بینی بازار مهم ترین کاربرد آن است. در زیر به شرح کامل این شبکه عصبی می پردازیم و در نهایت آن را بر روی داده های خود پیاده سازی می کنیم.

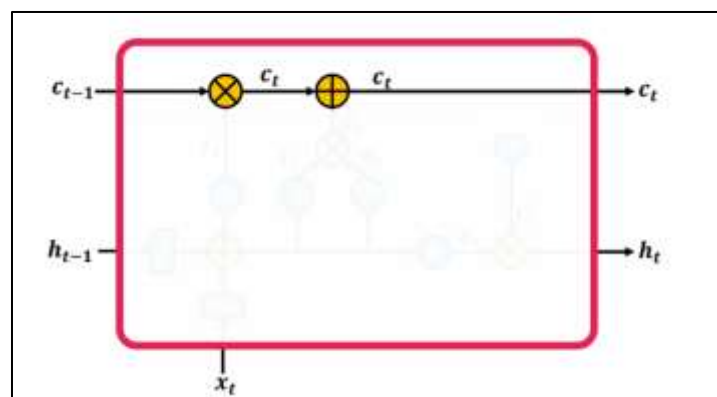
### شبکه عصبی LSTM

شبکه عصبی LSTM هم مانند شبکه RNN به صورت زنجیره ای پشت سرهم قرار می گیرد. شکل زیر این ساختار زنجیره ای را نشان می دهد..



### معماری شبکه LSTM

شبکه عصبی RNN یک ورودی و خروجی داشت. درواقع یک مسیر بین ورودی و خروجی شبکه RNN شکل می گیرد. اما شبکه LSTM متفاوت است. این شبکه دو ورودی و خروجی دارد که در شکل بالا می توانید مشاهده کنید. بین این ورودی و خروجی ها، یکی از ورودی ها مستقیم به خروجی متصل شده است. به شکل زیر نگاه کنید؛ ورودی  $C_{t-1}$  مستقیماً به خروجی  $C_t$  متصل شده است. این اتصال همین طور ساده از اول تا آخر دنباله ادامه دارد.  $C$  مخفف Cell State هست و یک مولفه کلیدی در LSTM است. به  $Cell State$ ، حافظه بلندمدت یا Long Term Memory هم گفته می شود.



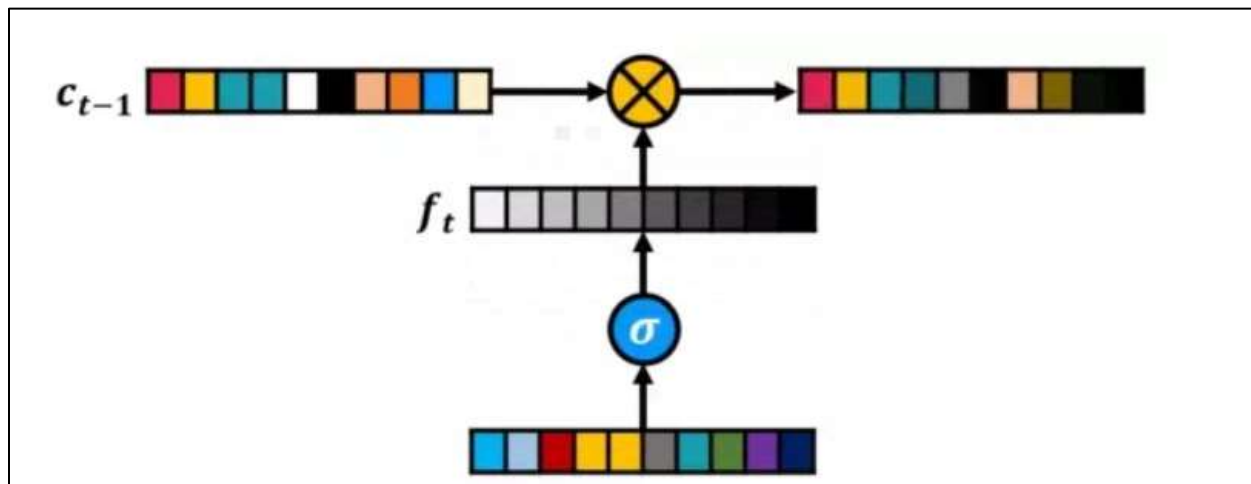
این خط ورودی همواره همه بلوک های LSTM در استپ های زمانی مختلف را به هم متصل می کند. در گذر زمان، در این حافظه بلندمدت ( $C$ ) اطلاعاتی ذخیره یا از آن حذف می شود. این حافظه بلندمدت دو خاصیت مهم و جالب دارد:

- می توانیم اطلاعات در آن را پاک کنیم. یعنی، همان فراموش کردن. ( علامت ضربدر در شکل بالا به معنای فراموش کردن است)

- می‌توانیم به آن اطلاعاتی اضافه کنیم. یعنی، همان به‌خاطر سپردن. (علامت به‌علاوه به معنای به‌خاطر سپردن)

## فراموشی در شبکه LSTM

به علامت ضربدر روی خط حافظه بلندمدت در شکل زیر نگاه کنید. این عملگر دو ورودی دارد که یکی همان  $C_{t-1}$  است. دومی هم یک ورودی است که در ادامه توضیح می‌دهم از کجا آمده است. ولی طبق شکل زیر ورودی قبل از اعمال از یک تابع سیگموئید می‌گذرد. این تابع سیگموئید باعث می‌شود که خروجی عددی بین ۰ تا ۱ شود. توجه داشته باشید که هر دو ورودی از جنس بردار با طول یکسان هستند که به‌صورت درایه به درایه ضرب می‌شوند و به عنوان خروجی یک بردار ایجاد می‌کنند.



حال پس از ضرب این دو بردار نتایج زیر ممکن هست حاصل شود:

- هر درایه‌ای از  $f_t$  اگر ۰ باشد، یعنی به درایه متناظرش در  $C_{t-1}$  اجازه نمی‌دهد به خروجی برود. یعنی می‌گوید بریز دور
- هر درایه‌ای از  $f_t$  اگر ۱ باشد، یعنی تمام ورودی  $C_{t-1}$  به خروجی می‌رود و دست‌نخورده باقی می‌ماند.
- اگر هم عددی بین ۰ تا ۱ باشد، طبیعتاً تا حدی روی ورودی تاثیر می‌گذارد. پس این گونه می‌توانیم اطلاعاتی را از داخل  $C_{t-1}$  حذف کنیم

## اما ورودی $f_t$ چگونه تشکیل می‌شود؟

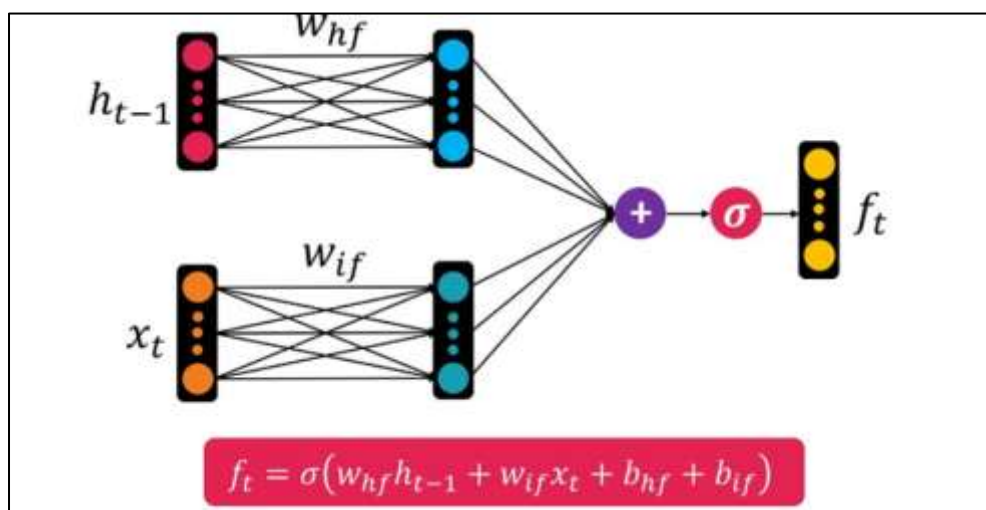
مطابق شکل زیر، این ورودی را یک شبکه عصبی کوچک با دو ورودی  $x_t$  و  $h_{t-1}$  تشکیل می‌دهد. این شبکه عصبی کوچک، وظیفه فراموشی بخشی از اطلاعات موجود در حافظه بلندمدت را دارد. به این شبکه عصبی گیت فراموشی (forget gate) گفته می‌شود.



ساختار گیت فراموشی به ساختار داخلی شبکه عصبی بازگشتی خیلی شبیه است. این شبکه دو ورودی  $x_t$  و  $h_{t-1}$  دارد. این دو ورودی با هم ترکیب می‌شوند و سپس از یک لایه سیگموئید می‌گذرند. لایه سیگموئید عددی بین ۰ تا ۱ ایجاد می‌کند که در بردار ورودی

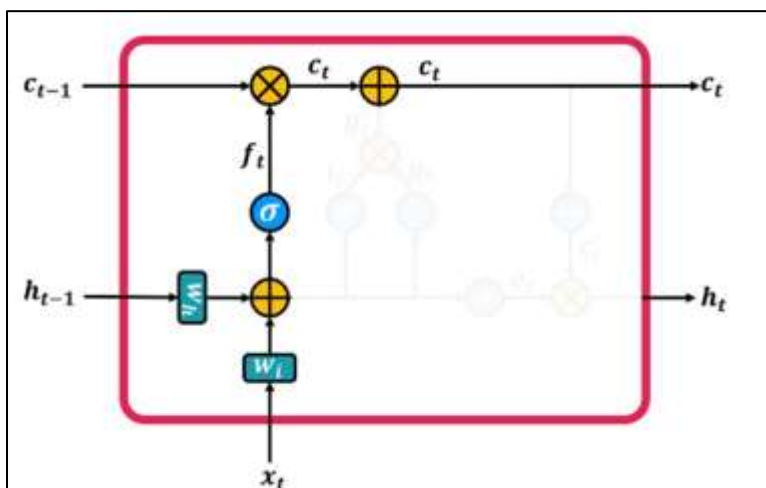
$C_{t-1}$  ضرب درایه به درایه می‌شود. این بردار به اندازه  $C_{t-1}$  است و تعیین می‌کند که هر درایه  $C_{t-1}$  باید در چه عددی از ۰ تا ۱ ضرب شود. هر درایه از  $f_t$  اگر به عدد ۱ نزدیک باشد، معنایش اینست که این درایه از  $C_{t-1}$  باید حفظ شود. اگر به ۰ نزدیک باشد، یعنی این درایه از  $C_{t-1}$  را دور بریز.

دو بردار ورودی  $x_t$  و  $h_{t-1}$  با بهره‌گیری از شبکه عصبی mlp با یکدیگر ترکیب می‌شوند که ساختار آن را در شکل زیر مشاهده می‌نمایید:



در شکل بالا، دو لایه کاملاً متصل به یکدیگر را با وزن‌های  $w_{hf}$  و  $w_{if}$  داریم. این دو لایه به ترتیب برای ورودی  $h_{t-1}$  و  $x_t$  هستند. اندیس hf مخفف عبارت hidden و forget هست. اندیس if مخفف عبارت input و forget هست. قابل ذکر است که دومین اندیس همواره به نام گیت اشاره می‌کند.

با توجه به توضیحات بالا فرایند فراموشی را میتوان به طور خلاصه در شکل زیر مشاهده کرد:



به خاطر سپردن در شبکه عصبی

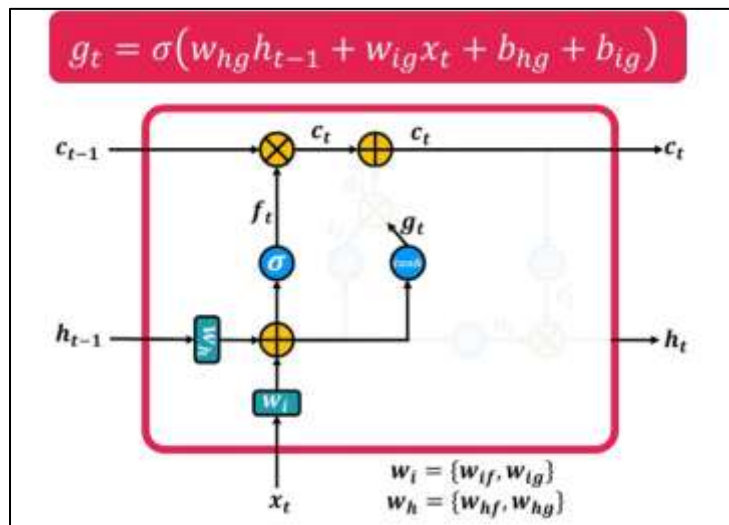
بعد از گذر از عملگر ضربدر ، به عملگر به علاوه روی مسیر  $C_t$  به  $C_{t-1}$  می‌رسیم. اینجا قرار است با ورودی  $C$  چیزی جمع شود. یعنی اطلاعات جدیدی به آن اضافه خواهد شد و به خاطر سپردن اتفاق می‌افتد. دو ورودی داریم که یکی  $C_t$  است. ورودی دوم را در ادامه توضیح می‌دهیم اما هر چه هست، برداری هم‌اندازه ورودی  $C$  است.

### می‌خواهیم چه اطلاعاتی از حال حاضر ( $t$ ) را در حافظه بلندمدت ذخیره کنیم؟

به شکل زیر نگاه کنید. این اطلاعات با استفاده از یک شبکه عصبی محاسبه می‌شود. این شبکه عصبی همانند گیت ورودی شامل دو ورودی  $x_t$  و  $h_{t-1}$  است. باز هم این ورودی‌ها باید از دو لایه متصل به هم ( $w_{hg}$  و  $w_{ig}$ ) عبور کنند و بعد با هم جمع شوند. حالا این ورودی از یک تابع تانژانت هایپربولیک باید عبور داده شود. خروجی  $g_t$  بین  $-1$  تا  $1$  خواهد بود. چرا بین  $-1$  تا  $1$  نیاز داریم؟ ساده هست، برای اینکه ممکن است بخواهیم اثر تعدادی از درایه‌ها یا مولفه‌های موجود در  $C$  را کاهش دهیم. یعنی با مقادیر بین  $-1$  تا  $1$  می‌توانیم اثر بعضی مولفه‌ها را زیاد یا کم کنیم. ( شکل )

تا اینجا فهمیدیم که هم گیت فراموشی و هم این شبکه جدید هردو ساختار مشابهی تا قبل از اعمال تابع تحریک دارند. یعنی هردو شبکه، دو ورودی دارند. هریک از این دو ورودی، به صورت جداگانه به یک شبکه چندلایه متصل می‌شوند و در نهایت هم با هم جمع می‌شوند. اما یک مشکل وجود دارد؛ ما در خروجی  $g_t$  یک مقداری اطلاعات داریم. اما شاید این ورودی آنقدر ارزش نداشته باشد که بخواهیم  $C_{t-1}$  را آپدیت کنیم. بله، می‌دانیم که این ورودی هم شامل اطلاعاتی است، اما شاید آنقدر ارزش نداشته باشد که بخواهیم به حافظه بلندمدت بسپاریم! فرض کنید الان در لحظه  $t$  در موقعیت کلمه " را " در جمله زیر هستیم:

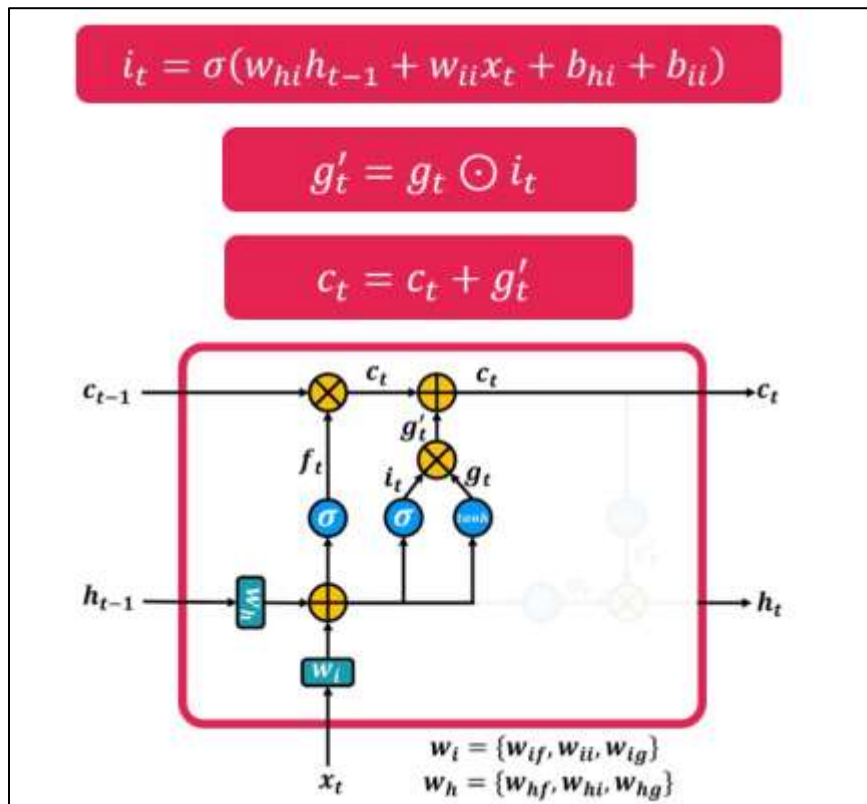
دو تیم رئال مادرید و بارسلونا برای جلوگیری از همه‌گیری کرونا، درخواست تعویق بازی را دارند.



به خاطر سپردن این کلمه برای تشخیص کلمات بعدی بسیار کم است پس بهتر است همین جا در آن را دور بریزیم و به حافظه بلندمدت انتقال ندهیم. برای تشخیص مهم بودن یا نبودن این کلمه کافی است نگاهی به گیت فراموشی بیندازیم. اگر از یک گیتی مشابه با ساختار گیت فراموشی استفاده کنیم و آنرا بر سر راه خروجی  $g_t$  قرار دهیم، می‌توانیم به راحتی تنظیم کنیم که این خروجی چقدر ارزش دارد. پس باز هم یک گیت یا دریچه مشابه شکل ۱۱ نیاز داریم. به این گیت جدید، گیت ورودی گفته می‌شود.

### گیت ورودی شبکه عصبی LSTM

گیت ورودی، ارزیاب میزان ارزش اطلاعات موجود در  $g_t$  است. به همین خاطر نامش را گیت ورودی گذاشته‌اند. مشابه گیت فراموشی، ممکن است مقادیر موجود در بردار  $i_t$  نزدیک به صفر باشد، بنابراین اثر  $g_t$  را کم می‌کند. برعکس، ممکن است مقادیر بردار  $i_t$  نزدیک به ۱ باشد، در این صورت  $g_t$  می‌رود تا در حافظه بلندمدت ذخیره شود. گیت ورودی، دو ورودی  $x_t$  و  $h_{t-1}$  را وارد دو لایه به هم متصل می‌کند و سپس اینها را با هم جمع می‌کند و در نهایت از تابع سیگموید عبور می‌دهد. در شکل زیر رابطه گیت ورودی، رابطه نحوه تشکیل خروجی نهایی و البته مقدار بروزرسانی شده  $C_t$  را نیز مشاهده می‌کنید.

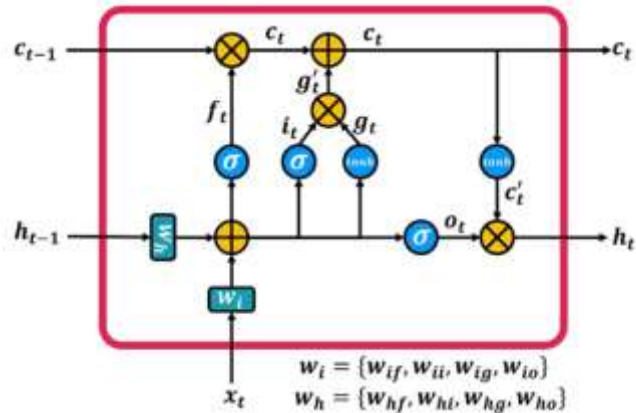


### ساخت خروجی $h_t$ در شبکه LSTM

برای ساختن خروجی شبکه بازگشتی LSTM ( $h_t$ ) خروجی  $C_t$  را از یک تانژانت هایپربولیک عبور می‌دهیم، سپس آماده می‌شویم که آنرا به خروجی متصل کنیم که در شکل زیر نشان داده شده است. اما برای پایان کار نیاز به یک گیت خروجی نیز هست که تعیین می‌کند چقدر از حافظه بلندمدت باید به خروجی منتقل شود. این گیت دو ورودی  $x_t$  و  $h_{t-1}$  را وارد دو لایه کاملاً متصل به هم می‌کند و سپس آن‌ها را با هم جمع می‌کند و در نهایت از تابع سیگموید عبور می‌دهد. در نهایت خروجی به دست آمده از گیت خروجی و خروجی حاصل شده از حافظه بلند مدت که دو بردار با طول یکسان هستند درایه به درایه در هم ضرب می‌شوند و خروجی شبکه در پرپود  $t$  حاصل می‌شود که شکل کامل و تمام ارتباطات موجود در یک سلول LSTM را در شکل زیر مشاهده می‌کنید.

$$o_t = \sigma(w_{ho}h_{t-1} + w_{io}x_t + b_{ho} + b_{io})$$

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



## پیوست ۲

### مدل هموارسازی نمایی دابل

در این مدل پیش‌بینی‌های ارائه شده علاوه بر مقدار مسطح شده<sup>۲۹</sup> از عنصر روند نیز استفاده می‌کنیم بدین شکل که پیش‌بینی ارائه شده در این مدل را به شکل زیر می‌توان بیان کرد:

$$\text{Forecast} = \text{smoothed value} + \text{correction using trend information}$$

اگر  $level_n$  را مقدار مسطح شده داده در پریود  $n$  و  $trend_n$  را مقدار رشد مورد انتظار در پریود بعدی تعریف کنیم، پیش‌بینی یک دوره آینده را میتوان به شکل زیر نوشت:

$$\hat{X}_{n+1} = level_n + trend_n$$

$$Level_n = \alpha \cdot \text{new information} + (1 - \alpha)(\text{old level} + \text{amount of trend})$$

$$level_n = \alpha x_n + (1 - \alpha)(level_{n-1} + trend_{n-1})$$

$$trend_n = \beta \cdot \text{new trend} + (1 - \beta) \cdot \text{old trend}$$

$$trend_n = \beta(level_n - level_{n-1}) + (1 - \beta)trend_{n-1}$$

مقادیر ابتدایی برای شروع مدل معمولاً برابر است  $level_1 = x_1$  و  $trend_1 = x_2 - x_1$

---

<sup>29</sup> Level



- [1] I. N. Tansel, S. Y. Yang, G. Venkataraman, A. Sasirathsiri, W. Y. Bao, and N. Mahendrakar, "Modeling time series data by using neural networks and genetic algorithms," in *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems: Proceedings of the Intelligent Engineering Systems Through Artificial Neural Networks*, C. H. Dagli, A. L. Buczak, J. Ghosh, M. J. Embrechts, and O. Erosy, Eds., vol. 9, pp. 1055–1060, ASME Press, New York, NY, USA, 1999.
- [2] C. K. Lee, Y. Sehwan, and J. Jongdae, "Neural network model versus SARIMA model in forecasting Korean stock price index (KOSPI)," *Issues in Information System*, vol. 8, no. 2, pp. 372–378, 2007.
- [3] N. Merh, V. P. Saxena, and K. R. Pardasani, "A comparison between hybrid approaches of ANN and ARIMA for Indian stock trend forecasting," *Journal of Business Intelligence*, vol. 3, no. 2, pp. 23–43, 2010.
- [4] J. Sterba and K. Hilovska, "The implementation of hybrid ARIMA neural network prediction model for aggregate water consumption prediction," *Aplimat—Journal of Applied Mathematics*, vol. 3, no. 3, pp. 123–131, 2010.
- [5] A. G. Lahane, "Financial forecasting: comparison of ARIMA, FFNN and SVR," 2008, <http://www.it.iitb.ac.in/~ashishl/files/MTechProjectPresentation.pdf>.
- [6] J. T. Yao, C. L. Tan, and H. L. Poh, "Neural networks for technical analysis: a study on KLCI," *International Journal of Theoretical and Applied Finance*, vol. 2, no. 2, pp. 221–241, 1999.
- [7] J. V. Hansen, J. B. McDonald, and R. D. Nelson, "Time series prediction with genetic-algorithm designed neural networks: an empirical comparison with modern statistical models," *Computational Intelligence*, vol. 15, no. 3, pp. 171–184, 1999.
- [8] V. R. Prybutok, J. Yi, and D. Mitchell, "Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations," *European Journal of Operational Research*, vol. 122, no. 1, pp. 31–40, 2000.
- [9] [wikipedia](#)
- [10] Stephanie Glen. "ADF — Augmented Dickey Fuller Test" From StatisticsHowTo.com
- [11] Ayodele Ariyo Adebisi, Aderemi Oluyinka Adewumi, Charles Korede Ayo, "Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction", *Journal of Applied Mathematics*, vol. 2014, Article ID 614342, 7 pages, 2014.
- [12] <https://civilica.com/doc/826436/>
- [13] <https://civilica.com/doc/1119544/>