به نام خدا



گزارش پروژه درس هوش مصنوعی پیشرفته با عنوان

**پیش بینی شاخص بورس با استفاده از تلفیق تکنیک های یادگیری ماشین**

**نویسنده**

سید حسین فانی یزدی

**استاد**

محبوبه هوشمند

**تاریخ**

24/11/1399

فهرست

[**مقدمه:** 5](#_Toc64064086)

[**معرفی مقاله:** 6](#_Toc64064087)

[**پیاده سازی مقاله:** 8](#_Toc64064088)

[**دادههای ورودی:** 8](#_Toc64064089)

[**شبکه عصبی:** 9](#_Toc64064090)

[**طبقه بند SVR:** 13](#_Toc64064091)

[**پیاده سازی الگوریتم ها:** 14](#_Toc64064092)

[**نتایج:** 14](#_Toc64064093)

[**نتیجه گیری:** 22](#_Toc64064094)

[**منابع:** 22](#_Toc64064095)

فهرست تصاویر

[تصویر 1 - معماری کلی رویکرد تک مرحله ای برای پیش بینی n امین روز آینده 7](#_Toc64064257)

[تصویر 2 - معماری کلی رویکرد دو مرحله ای برای پیش بینیnامین روز آینده 7](#_Toc64064258)

[تصویر 3 - جزئیات رویکرد دو مرحله ای برای پیش بینی nامین روز آینده 8](#_Toc64064259)

[تصویر 4 - مدل شبکه عصبی (مدل پرسپترون با n نرون در لایه مخفی) 11](#_Toc64064260)

فهرست جدول (Machine\_Learning\_A-Z, بدون تاريخ)

[Table 1 - نتایج بدست آمده از مدل SVR در برابر ورودی ها متفاوت و مقدار Gamma متفاوت 14](#_Toc64064320)

[Table 2 - اطلاعات بدست آمده از اجرای الگوریتم های شبکه عصبی با داده¬های بدست آمده از 10 اندیکاتور و مقایسه با تنظیمات مختلف و دو نوع شبکه عصبی متفاوت 16](#_Toc64064321)

[Table 3 - تصاویر نمودارهای همگرایی شبکه¬ها عصبی در جدول 2 (نمودار خطای MSE نسبت به تعداد تکرار) (در شکل¬های ضریب 3، نمودار قرمز میزان خطای آموزش و نمودار آبی میزان خطای اعتبار سنجی می¬باشد.) 17](#_Toc64064322)

[Table 4 - اطلاعات بدست آمده از اجرای الگوریتم های شبکه عصبی با داده های بدست آمده از قیمت پایانی 60 روز گذشته و مقایسه با تنظیمات مختلف و دو نوع شبکه عصبی متفاوت 18](#_Toc64064323)

[Table 5 - تصاویر نمودارهای همگرایی شبکه ها عصبی در جدول 4 (نمودار خطای آموزش MSE نسبت به تعداد تکرار) (در شکل¬های ضریب 3، نمودار قرمز میزان خطای آموزش و نمودار آبی میزان خطای اعتبار سنجی می¬باشد.) 19](#_Toc64064324)

# **مقدمه:**

امروزه به دلیل گسترش فعالیت های اقتصادی بازارهای مالي و رونق سرمایه گذاری در بازارهای سرمایه به خصوص بورس اوراق بهادار توسط اشخاص حقيقي و حقوقي دسترسی به اطلاعات درست و به موقع و تحلیل دقیق و واقع بینانه آنها مهم ترین ابزار جهت اتخاذ تصمیم های درست و کسب منفعت مورد انتظار و استفاده بهینه و مطلوب از امکانات مالی می باشد. به نظر بعضی از کارشناسان سرمایه گذاری در سهام عرضه شده در بورس اوراق بهادار یکی از گزینه های پرسود در بازار سرمایه است. بورس اوراق بهادار، كه به معني یک بازار متشکل و رسمي سرمایه است كه در آن، خرید و فروش سهام شركتها یا اوراق قرضه دولتی یا موسسات معتبر خصوصي، تحت ضوابط و قوانين و مقررات خاصی انجام میشود؛ بنابراین دارای مشخصه مهم حمایت قانون از صاحبان پس اندازها و سرمایه های راكد و الزامات قانوني برای متقاضيان سرمایه به همين دليل بوده، و پيش بيني این بازار یکی از مهم ترین علاقه پژوهشگران و محققان مالي محسوب شده و یکي از مهم ترین اطلاعات در بازار بورس اوراق بهادار برای سرمایه گذاران اطلاعات قیمت سهام است. قيمت سهام به طور اساسی داینامیک غيرخطي، ناپاراماریک و آشوب گونه مي باشد؛ و نشان مي دهد كه سرمایه گذاران باید سری های زماني را به كار ببرند كه ناایستا و دارای ساختار آشوبگونه است. در حقیقت پراكندگي قيمت سهام تحت تأثير عوامل كلان اقتصادی مثل وقایع سياسی، سياست های شركت ها، شرایط اقتصادی، نرخ بهره، نرخ تورم، انتظارات سرمایه گذاران، سرمایه گذاران سنتی، انتخاب و فاکتورهای فيزیکي و رواني سرمایه گذاران ميباشد. بنابرین امروزه، پیش بینی قیمت سهام نه فقط خيلي چالش برانگيز، بلکه مورد علاقه زیاد سرمایه گذاران بوده و ازآنجا كه نوسان قيمت سهام یک روند غيرخطي دارد، روش های كلاسیک مناسب این كار نيستند؛ به دليل اینکه درجه خطایشان زیاد است در حالي كه باید این خطا كمينه شود.

پيش بيني قيمت سهام به عنوان یکي از چالش برانگيزترین كاربردهای پيش بيني سری های زماني در نظر گرفته شده است. پيش بيني قيمت سهام به عنوان یک مساله چالش برانگیز، فرآیند پيش بينی سری های زماني بوده، به دليل اینکه بازار سهام در اصل دیناميک، غيرخطی، پیچیده، ناپارامتریک و بی نظم است. علاوه براین، قيمت سهام از تعداد زیادی فاكتورهای اقتصاد كلان مثل رویدادهای سياسي، نرخ تورم، شرایط اقتصاد عمومي، انتظارات و رفتار سرمایه گذاران تأثير می پذیرد و همچنين متغيرهای مالي مثل حجم مبادلات، نسبت E/P و ... بر قيمت سهام هر شركتی تاثیر زیادی دارند و در واقع رفتار سرمایه گذاران را جهت میدهند. شبکه های عصبی برای حل مسائل پیچیده و غيرخطي مناسب تر از سری های زماني هستند، هرچند مشکلات خود رانیز دارند؛ مثلا در یادگيری دچار محدودیت اند چون داده های مربوط به قیمت سهام دارای اختلافات فاحش و ابعاد پیچیده ای هستند و لذا پيش بيني قیمت سهام کار بسیار مشکلی است.

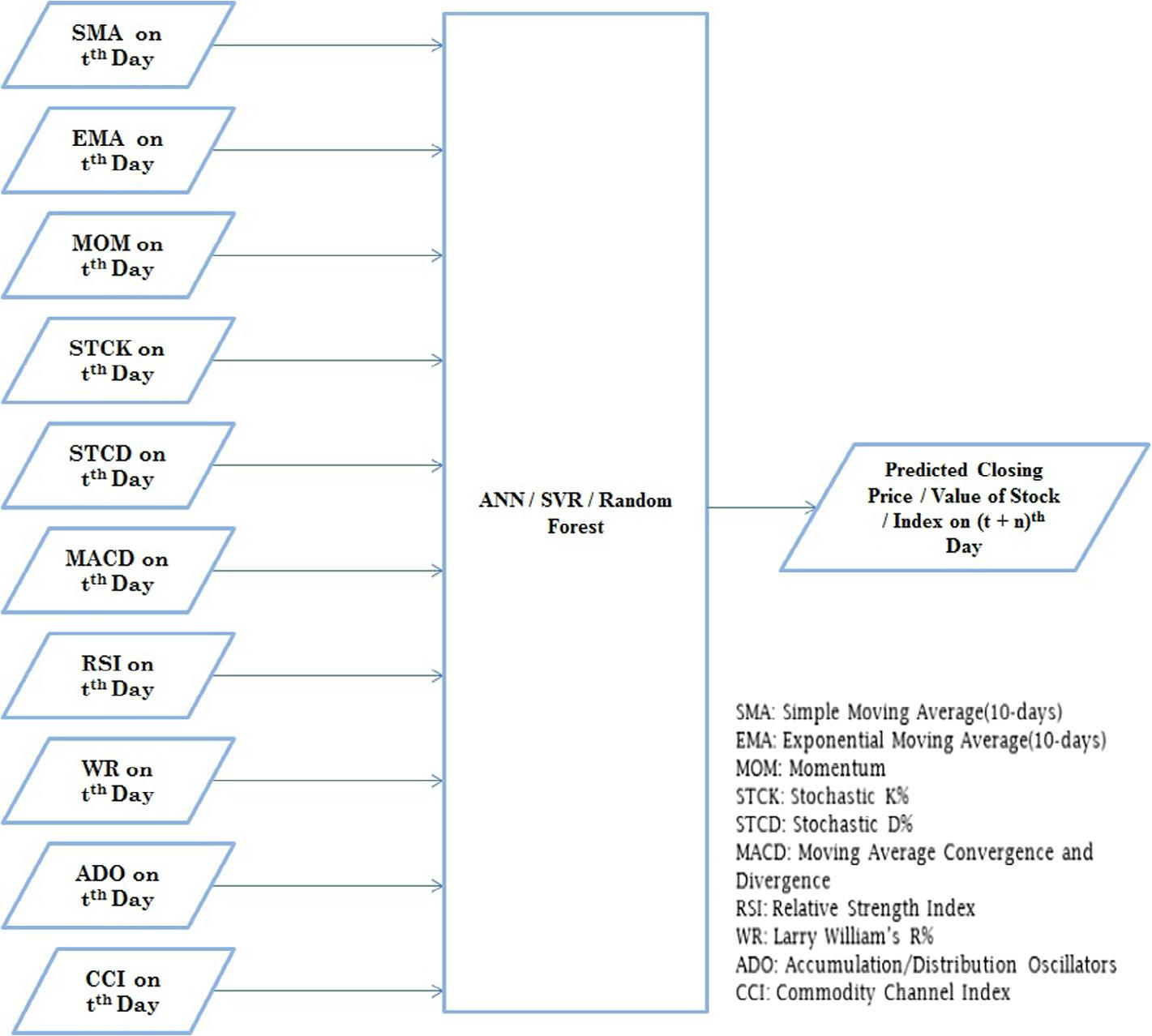
# **معرفی مقاله:**

در این مقاله سعی شده تا با استفاده از تکنیک های مختلف یادگیری ماشین، مدلی را بسازد که بتواند شاخص بازار سرمایه را در روزهای پیشرو پیش بینی کند. در نظر داشته باشید که محقق برای رسیدن به چنین مدلی ابتدا راه حلی را معرفی می­کند و در مرحله بعدی سعی دارد تا آنرا بهبود ببخشد. سپس با انجام آزمایشاتی نتایج هر دو مدل رو باهم مقایسه می­کند تا بتواند نتیجه گیری مناسبی داشته باشد.

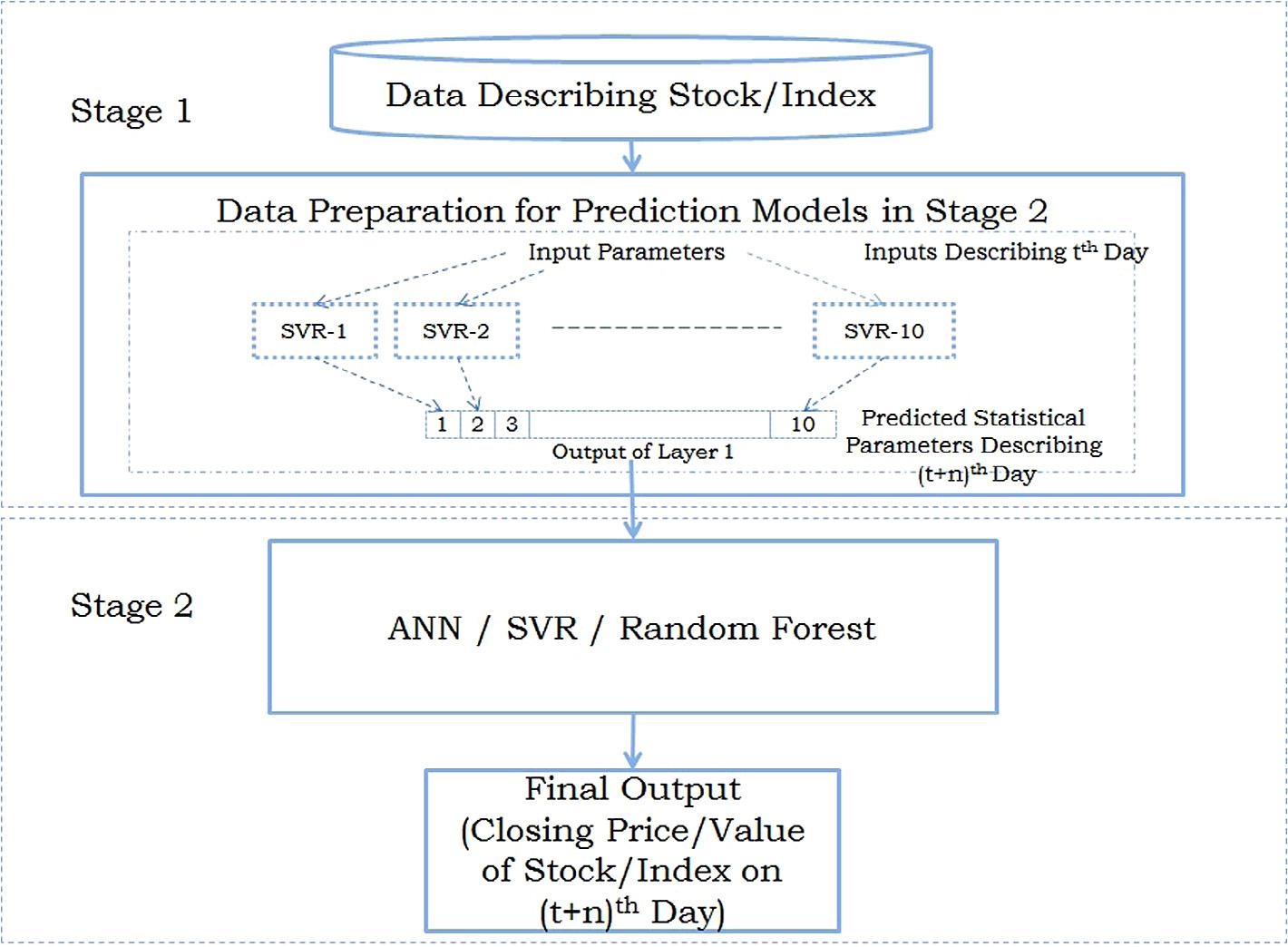
محقق با معرفی طبقه بند SVR و Random Forest و شبکه عصبی، ابتدا مدلی تک فاز را ارائه می­دهد که با توجه به ورودی خود می­تواند روزهای پیشرو را پیش بینی کند. در این مدل تک فاز می­توان از هر یک از تکنیک ها به صورت جداگانه استفاده کرد. سپس مدلی دو فاز و ترکیبی را پیشنهاد میدهد که در فاز اول ابتدا داده­های ورودی توسط تکنیک SVR پرداز می­شوند و پس از آن به فاز دوم می­روند. در این مدل، فاز دوم همانند مدل تک فاز می­باشد با این تفاوت که ورودی ها را از فاز اول دریافت میکند.

در هر یک از این دو مدل پیشنهادی، داده های بازار سرمایه توسط 10 اندیکاتور پردازش میشوند و سپس به مدل­ها داده می­شوند. باید توجه داشت که در بعضی اندیکاتور­ها، نیاز است تا تنظیماتی را در نظر بگیریم. به عنوان مثال برای اندیکاتور SMA باید تعین کنیم که چه تعداد از روزهای­گذشته را پردازش کنیم. به همین دلیل با اینکه در مقاله بعضی از تنظیمات بیان شده اند، اما باز هم ممکن است در پیاده سازی برخی تنظیمات و محاسبات متفاوت باشند. قابل توجه است که در یکی از دو مورد از اندیکاتور ها محاسبات و فرمول پیشنهادی در مقاله اشتباه بوده که یقینا در نتایج تاثیر خواهد داشت. با توجه به همه این احتمالات، میتوان گفت که ممکن است داده­های ورودی در پیاده سازی و داده­های محقق متفاوت باشند.

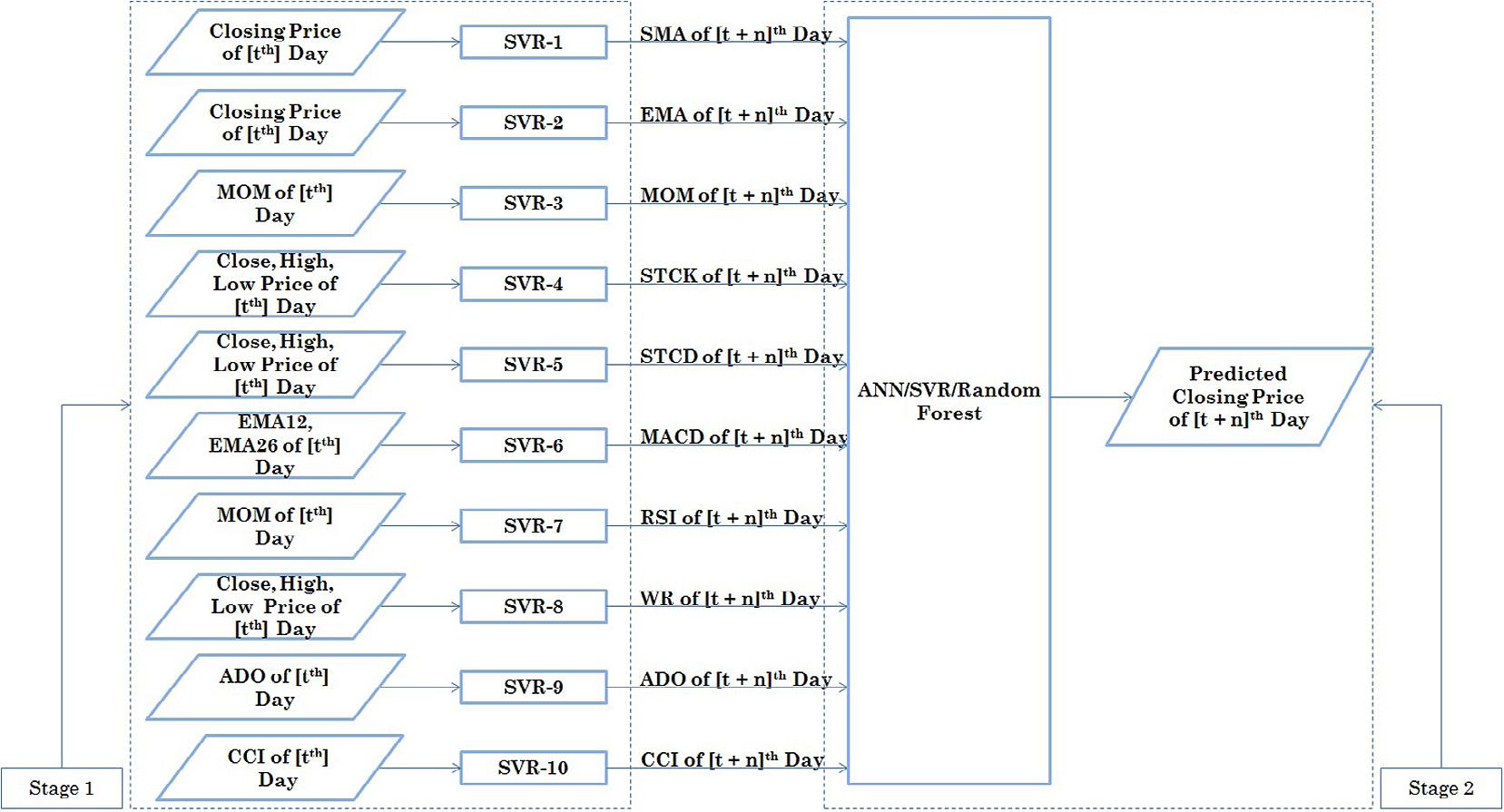
محقق بعد از معرفی مدل­ها و معرفی اندیکاتور­ و داده­های آموزشی، مدل­های پیشنهادی را آزمایش میکند و نتایج را میسنجد و مقایسه می­کند. قابل ذکر است که توضیحات مقاله در مورد پیاده سازی بسیار ناچیز بوده و در بعضی موارد، اطلاعات و راه­های مورد استفاده غلط می­باشد.



تصویر 1 - معماری کلی رویکرد تک مرحله ای برای پیش بینی n امین روز آینده



تصویر 2 - معماری کلی رویکرد دو مرحله ای برای پیش بینیnامین روز آینده



تصویر 3 - جزئیات رویکرد دو مرحله ای برای پیش بینی nامین روز آینده

# **پیاده سازی مقاله:**

## **داده­های ورودی:**

برای ارائه داده­ها به شبکه عصبی و یا تکنیک­های دیگر، دو راه پیش رو داریم. اول استفاده از 10 اندیکاتور همانند چیزی که در مقاله معرفی شده بود. راهی دیگر اینگونه است که میتوان فقط قیمت پایانی n روز قبل (60 روز) را به عنوان ورودی به شبکه بدهیم تا شبکه بر اساس روند و یادگیری خود بتواند روز بعدی را پیش بینی کند. بدیهی است که در روش دوم تعداد نرون های لایه ورودی برابر n خواهند بود. عدد 60 معرفی شده در این بخش با آزمون و خطا بدست امده. لازم به ذکر است که روش دوم از اطمینان زیادی برخوردار نیست و ممکن است در نوسانات بزار عمل پیش بینی به درستی عمل نکند.

هر دو روش های فوق را می­توان روی داده­های بازار­های مختلف منجمله ایران استفاده کرد اما در نظر داشته باشد که برای بعضی اندیکاتورها به داده­های بجز قیمت پایانی نیاز است. در ادامه، 10 اندیکاتور را به اختصار معرفی می­کنیم. لازم به ذکر است که برای محاسبه اندیکاتور ها از دو کتابخانه معتبر ta و finta استفاده شده است که محاسبات خود را بر اساس استاندارد­های جهانی انجام داده اند اما برای کامل بودن کار بعضی اندیکاتورها را نیز خودمان نیز پیاده کرده­ایم و آنهارا در توابعی به صورت جامع نوشتیم تا کاربر به راحتی از آنها استفاده کند.

indicatorName = []

Stock\_data\_pd = readStockData()

*# (use = 1) -> selfwrite*

*# (use = 2) -> ta library*

*# (use = 3) -> finta library*

SMA(Stock\_data\_pd, 10, *use* = 3)

EMA(Stock\_data\_pd, 50, *use* = 3)

RSI(Stock\_data\_pd, 50, *use* = 3)

CCI(Stock\_data\_pd, 20, 0.015, *use* = 3)

MACD(Stock\_data\_pd, *period\_fast* = 12,  *period\_slow* = 26,  *period\_sign* = 9, *use* = 3)

WilliamsR(Stock\_data\_pd, *period* = 14, *use* = 3)

Momentum(Stock\_data\_pd, *period* = 10, *use* = 3)

Stochastics(Stock\_data\_pd, 14, 3, *use* = 3)

*# AccDistIndicator(Stock\_data\_pd, use = 3)*

saveStockData\_toCSV(Stock\_data\_pd, 'With\_Indicators')

ابتدا داده­های بازار را توسط تابعی مشخص و با کمک کتابخانه panda میخوانیم و سپس با استفاده از توابع ساخته شده برای اندیکاتور­ها انهارا به داده های خود اضافه می­کنیم. تنظیمات استفاده شده، با توجه به استاندارد­های جهانی انتخاب شده­اند اما باید بدانیم که این اعداد براساس تجربه بدست می­آیند و باید متناسب با نمودار انتخاب شود.

بعد از آنکه داده­های خود را بدست آوردیم (چه از روش اول و چه از روش دوم)، باید داده هارا نرمالسازی کنیم تا بازه تغییرات داده­ها باهم اختلاف زیادی نداشته باشد. بدین منظور از روش MinMaxScaling استفاده می­کنیم تا داده­ها در بازه 0 تا 1 نرمال شوند.

بعد از آن داده­ها به 3 دسته آموزش، تست و پیش بینی تقسیم می­شوند. دلیل آن اینست که داده­ها تست و پیش بینی باید دیده نشده و جدا از داده­های آموزش باشند. بدین ترتیب 70 درصد کل داده­ها را برای آموزش، 20 درصد برای تست و 10 درصد برای پیش­بینی در نظر می­گیریم. البته می­توانستیم بجای داده­های پیش­بینی از داده­ها تست هم استفاده کنیم اما بدلیل وجود داده کافی برای بخش آموزش، تصیمیم بر این تقسیم بندی گرفته شد.

## **شبکه عصبی:**

از آنجا که محقق در ارائه اطلاعات در مورد نوع، نحوه و چکونگی پیاده سازی کوتاهی کرده است و فقط به ارائه نتایج و توضیحات کوتاه بسنده کرده است، بناچار مجبور بوده­ایم تا با استفاده از دانش خود اقدام به پیاده سازی کنیم. در پیاده سازی شبکه عصبی، از دو نوع شبکه با ساختار متفاوت استفاده کرده­ایم. ساختار اول، شبکه عصبی پرسپتورن که ساختاری با اتصال کامل و متراکم را دارد و ساختار دوم، شبکه عصبی بازگشتی و دارای حافظه می­باشد که با توجه به پیچیدگی مناسب و حافظه خود در پیش بینی سری­های زمانی، در مواردی کمی بهتر از شبکه عصبی پرسپترون می­باشد.

در این پیاده سازی که تماما با زبان Python می­باشد با توجه به نیاز برای توسعه شبکه عصبی از کتابخانه­های Keras و Tensorflow استفاده شده است. در ادامه ساختار این دو شبکه را بررسی می­کنیم. ابتدا شبکه پرسپترون:

layerName = 'Dense'

*# Designing the Neural*

model = Sequential()

model.add(Dense(50, *input\_dim* = 10, *activation* = 'sigmoid', *name*='layer\_1'))

model.add(Dense(1 ,  *activation*='relu', *name*='layer\_2'))

*# Compiling and Get Summary*

model.compile(*optimizer* = 'adam' , *loss* = 'mse')

model.summary()

در پیاده سازی این شبکه با توجه به نوع داده­های ورودی تعداد 10 نرون در لایه ورودی می­باشد که می­توان براساس نوع داده ورودی آنرا تغییر داد و 50 نرون در لایه میانی که با فعال ساز سیگمویید عمل می­کند و 1 نرون در لایه خروجی که قیمت پایانی روز بعد را برای ما پیش بینی می­کند و دارای فعال ساز خطی می­باشد. برای محاسبه خطا از معیار میانگین مربعات خطا (MSE) و برای بهینه سازی از گرادیان نزولی تطبیقی (adam) استفاده شده است. در مراحل قبلی با انواع داده­های ورودی بکار گرفته شده آشنا شده­اید. در قسمت بعدی، پیاده سازی شبکه عصبی بازگشتی و دارای حافظه را بررسی می­کنیم.

layerName = 'LSTM\_Dense'

*# Designing the Neural*

model = Sequential()

model.add(LSTM(50, *return\_sequences* = True , *input\_shape* = (X\_Train.shape[1] , 1) , *name*='layer\_1'))

model.add(LSTM(100, *return\_sequences* = False , *name*='layer\_2'))

model.add(Dense(50 , *name*='layer\_3', *activation*='relu'))

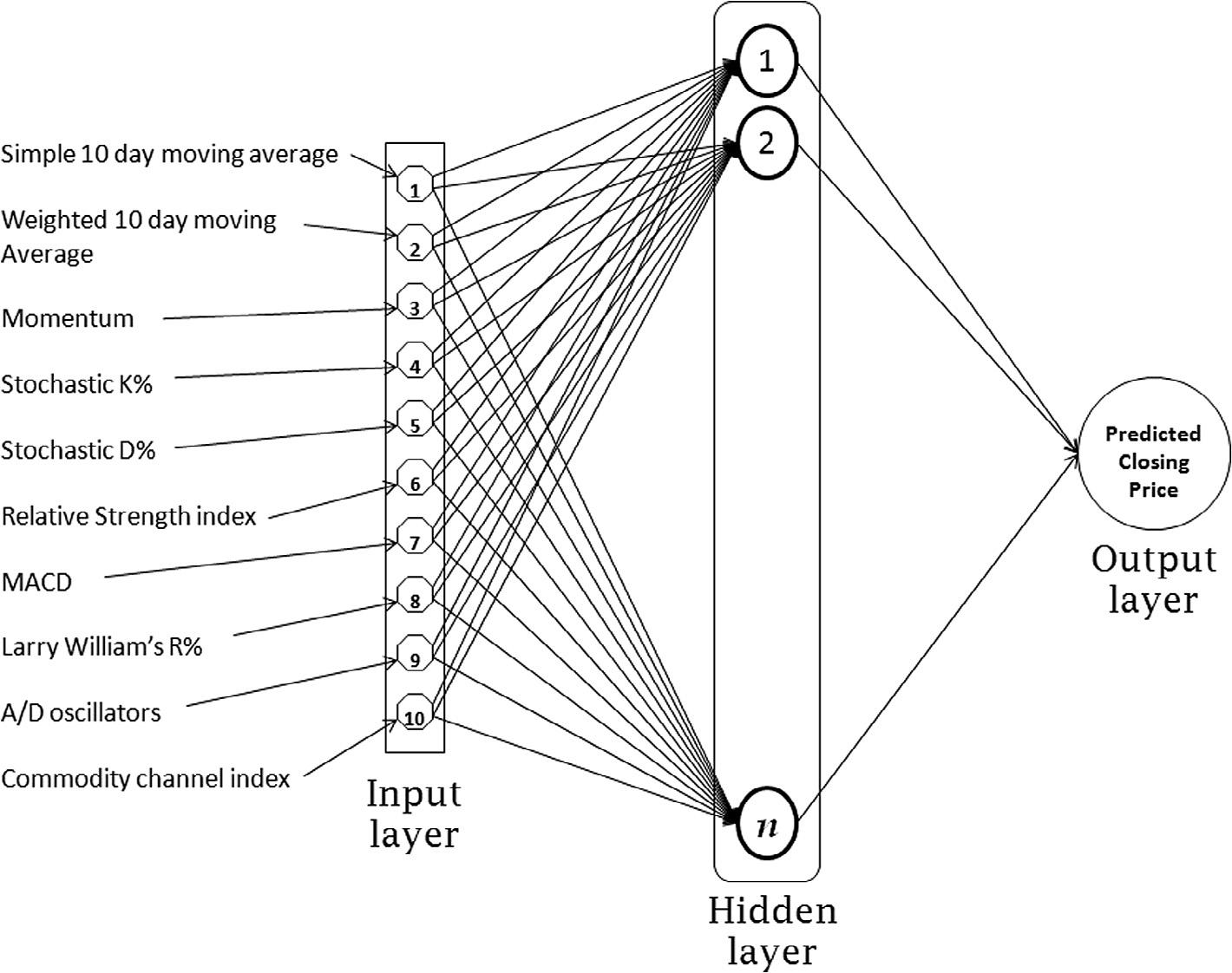
model.add(Dense(1 ,  *name*='layer\_4', *activation*='relu'))

*# Compiling and Get Summary*

model.compile(*optimizer* = 'adam' , *loss* = 'mse')

model.summary()

در این پیاده سازی از دو لایه LSTM و دولایه Dense استفاده شده است که در لایه اول LSTM با 50 نرون عمل بازگشتی انجام می­شود و در لایه بعدی با 100 نرون عمل بازگشتی فقط بین نرون ها صورت می­گیرد. در دو لایه بعدی Dense، یکی با 50 نرون و دیگری با 1 نرون به عنوان لایه خروجی، از تابع فعالساز خطی و همچنین برای محاسبه خطا از معیار میانگین مربعات خطا (MSE) و برای بهینه سازی از گرادیان نزولی تطبیقی (adam) استفاده شده است.



تصویر 4 - مدل شبکه عصبی (مدل پرسپترون با n نرون در لایه مخفی)

تعیین تعداد نرون­ها و تعدادلایه­های میانی، بر اساس تجربه و آزمون و خطاست. در هر دو شبکه عصبی این ساختار نتیجه بهتری را رقم میزد. شاید با تغییرات در ساختار بتوان به راه حل­های بهتری نیز رسید. علاوه بر این، تعیین تعداد تکرار در شبکه عصبی امری مهم است که این مهم نیز با بررسی و آزمایشات بسیار بدست می­اید. البته راه­های مثل نمودار اعتبار سنجی نیز می­تواند به پیدا کردن تکرار مناسب کمک کند.

*# Trian Setting*

epochs            = 10

verbose           = 1

shuffle           = True

validation\_split  = 0.0

*# Training action*

model.fit(  *x*                 = X\_Train,

*y*                 = Y\_Train,

*batch\_size*        = 1,

*verbose*           = verbose,

*epochs*            = epochs,

*shuffle*           = shuffle,

*validation\_split*  = validation\_split,

*callbacks*         = [logger]

        )

در هر دو شبکه عصبی عمل آموزش با دستورات یکسان اجرا می­شود. تعداد تکرار­ها، نوع لاگ کردن، بهم ریختن ترتیب داده­ها و تقسیم داده­های آموزش برای داشتن داده­های اعتبار سنجی در این بخش مشخص می­شود. دستور بالا برای آموزش شبکه استفاده می­شود.

test\_Error\_MSE = model.evaluate(*x*        = X\_Test,

*y*        = Y\_Test,

*verbose*  = 2)

در دستور بالا، بعد از عمل آموزش، شبکه با داده­ها تست، لرزیابی می­شود و خطا با معیار MSE محاسبه شده باز می­گردد.

Pure\_Predictions = model.predict(X\_Predict)

در نهایت نیز، بعد از تست می­توانیم عمل پیش­بینی را با کمک دستور بالا انجام دهیم.

بعد از بدست آوردن نتایج از بخش تست و پیش­بینی، می­توانیم میزان خطا را برحسب معیار­های متفاوت بیان کنیم. در اینجا از سه معیار MSE، RMSE و MAE استفاده می­کنیم. خطا در بخش تست و ارزیابی بدینگونه است.

print(*f*"---------------TEST---------------")

test\_Error\_RMSE  = np.sqrt(test\_Error\_MSE)

print(*f*"The MSE for Test data set is: {test\_Error\_MSE} , RMSE Value: {test\_Error\_RMSE}")

print(*f*"The MSE for Test data set is: {test\_Error\_MSE*:.4e*} , RMSE Value: {test\_Error\_RMSE*:.4e*}")

print(*f*"------------------------------")

خطا در بخش پیش­بینی نیز اینگونه محاسبه می­شود.

print(*f*"---------------PREDICTION---------------")

predict\_Error\_mse = mean\_squared\_error(Y\_Predict, Predictions)

predict\_Error\_mae = mean\_absolute\_error(Y\_Predict, Predictions)

print(*f*"Prediction MSE value: {predict\_Error\_mse} and RMSE value: {np.sqrt(predict\_Error\_mse)} and MAE value: {predict\_Error\_mae}")

print(*f*"Prediction MSE value: {predict\_Error\_mse*:.4e*} and RMSE value: {np.sqrt(predict\_Error\_mse)*:.4e*} and MAE value: {predict\_Error\_mae*:.4e*}")

print(*f*"------------------------------")

## **طبقه بند SVR:**

این طبقه بند از دسته طبقه بند SVM می­باشد که در تلاش است تا الگوی مناسب را از داده­ها پیدا کند. برای پیاده­ سازی این بخش از کتابخانه sklearn استفاده شده است. برای محاسبه خطا در این بخش از همین کتابخانه کمک گرفته شده است.

در قطعه کد­های زیر، بعد از فراخوانی کتابخانه­ها، ابتدا مدل ساخته می­شود. سپس آموزش داده شده و پس از آن با داده­ها تست و پیش­بینی جداگانه ارزیابی می­شود. بعد از بدست آوردن نتایج، میزان خطا را برحسب معیار­های متفاوت بیان می­کنیم. در اینجا از سه معیار MSE، RMSE و MAE استفاده می­کنیم.

*# Libraries*

from sklearn.svm import SVR from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

svr  = SVR(*kernel* = 'rbf', *gamma* = 2)

svr.fit(X\_Train, Y\_Train)

Predict\_Y\_Test      = svr.predict(X\_Test)

Predict\_Y\_Predict   = svr.predict(X\_Predict)

print(*f*"---------------TESTING---------------")

test\_Error\_mse = mean\_squared\_error(Y\_Test, Predict\_Y\_Test)

test\_Error\_mae = mean\_absolute\_error(Y\_Test, Predict\_Y\_Test)

print(*f*"Prediction MSE value: {test\_Error\_mse} and RMSE value: {np.sqrt(test\_Error\_mse)} and MAE value: {test\_Error\_mae}")

print(*f*"Prediction MSE value: {test\_Error\_mse*:.4e*} and RMSE value: {np.sqrt(test\_Error\_mse)*:.4e*} and MAE value: {test\_Error\_mae*:.4e*}")

print(*f*"---------------PREDICTION---------------")

predict\_Error\_mse = mean\_squared\_error(Y\_Predict, Predict\_Y\_Predict)

predict\_Error\_mae = mean\_absolute\_error(Y\_Predict, Predict\_Y\_Predict)

print(*f*"Prediction MSE value: {predict\_Error\_mse} and RMSE value: {np.sqrt(predict\_Error\_mse)} and MAE value: {predict\_Error\_mae}")

print(*f*"Prediction MSE value: {predict\_Error\_mse*:.4e*} and RMSE value: {np.sqrt(predict\_Error\_mse)*:.4e*} and MAE value: {predict\_Error\_mae*:.4e*}")

print(*f*"------------------------------")

# **پیاده سازی الگوریتم ها:**

در پیاده سازی مقاله با توجه به گسترده بوده انتخاب­ها برای پیاده سازی، تصمیم بر این شده است که 6 الگوریتم متفاوت در 6 فایل پایتون توسعه یابد که هر کدام از نظر ویژگی با دیگری متفاوت است. در اینجا سه نوع مدل متفاوت (دو نوع شبکه عصبی متفاوت و یک مدل SVR) و دو روش برای تولد داده­ها ورودی توسعه یافته است که جمعا 6 روش را پیشنهاد می­دهد. فایل­ها در کنار گزارش ارائه خواهند شد.

با توجه با گنگ بودن روش دو مرحله­ای پیشنهادی در مقاله، در این گزارش سعی شده تا با تمرکز بر مدل تک مرحله­ای و بهینه سازی آن نتایج را بهبود ببخشیم.

# **نتایج:**

این نتایج بر اساس پیش بینی 1 روز آینده جمع آوری شده اند یعنی بعد از آموزش کافیست داده­های ورودی لازم برای امروز را جمع اوری کرده تا شکبه قیمت پایانی فردا را پیش­بینی کند. در ابتدا نتایج مربوط به مدل SVR را بررسی می­کنیم.

Table 1 - نتایج بدست آمده از مدل SVR در برابر ورودی ها متفاوت و مقدار Gamma متفاوت

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| نوع داده­ها | Gamma | MSE TEST | RMSE TEST | MAE TEST | MSE Prediction | RMSE Prediction | MAE Prediction |
| با اندیکاتور | 0.5 | 2.8841e-03 | 5.3703e-02 | 4.2817e-02 | 1.1592e-03 | 3.4047e-02 | 2.7068e-02 |
| با اندیکاتور | 1 | 3.0591e-03 | 5.5309e-02 | 4.3878e-02 | 8.0025e-04 | 2.8289e-02 | 2.1437e-02 |
| با اندیکاتور | 2 | 3.7673e-03 | 6.1379e-02 | 4.9078e-02 | 9.6457e-04 | 3.1058e-02 | 2.4871e-02 |
| با اندیکاتور | 3 | 5.0277e-03 | 7.0906e-02 | 5.6752e-02 | 1.4181e-03 | 3.7657e-02 | 3.0057e-02 |
| با اندیکاتور | 4 | 6.1920e-03 | 7.8690e-02 | 6.3070e-02 | 1.7355e-03 | 4.1660e-02 | 3.2596e-02 |
| با اندیکاتور | 5 | 7.5600e-03 | 8.6948e-02 | 7.0611e-02 | 2.1350e-03 | 4.6206e-02 | 3.5881e-02 |
| با اندیکاتور | 10 | 1.6791e-02 | 1.2958e-01 | 1.1395e-01 | 6.1885e-03 | 7.8667e-02 | 6.9134e-02 |
| پایانی60 روز گذشته | 0.5 | 2.5670e-03 | 5.0666e-02 | 4.0352e-02 | 1.2910e-03 | 3.5931e-02 | 3.0113e-02 |
| پایانی60 روز گذشته | 1 | 2.9031e-03 | 5.3880e-02 | 4.2616e-02 | 6.6236e-04 | 2.5736e-02 | 2.2110e-02 |
| پایانی60 روز گذشته | 2 | 3.9242e-03 | 6.2643e-02 | 4.8878e-02 | 6.4072e-04 | 2.5312e-02 | 2.0832e-02 |
| پایانی60 روز گذشته | 3 | 6.3975e-03 | 7.9984e-02 | 6.6304e-02 | 2.1387e-03 | 4.6246e-02 | 4.0580e-02 |
| پایانی60 روز گذشته | 4 | 9.9583e-03 | 9.9791e-02 | 8.8095e-02 | 5.1116e-03 | 7.1495e-02 | 6.4689e-02 |
| پایانی60 روز گذشته | 5 | 1.2899e-02 | 1.1358e-01 | 1.0190e-01 | 7.1448e-03 | 8.4527e-02 | 7.6596e-02 |
| پایانی60 روز گذشته | 10 | 7.6596e-02 | 1.6407e-01 | 1.5012e-01 | 1.6216e-02 | 1.2734e-01 | 1.1706e-01 |

با توجه به نتایج بدست آمده از جدول می­توان عنوان کرد که مقدار gamma کوچکتر نتیجه بهتری را به ارمقان می­آورد. با این حال در نوع داده­های ورودی دوم (قیمت پایانی 60 روز گذشته) کمترین مقدار خطا در gamma برابر 2 می­باشد. اما کمترین خطا در نوع دیگر داده متغییر می­باشد. علاوه بر این می­توان بیان کرد که نتایج در نوع داده دوم (قیمت پایانی 60 روز گذشته) بسیار رضایت بخش­تر نسبت به نوع داده اول (اندیکاتورها) می­باشد.

در مرحله بعدی نتایج بدست آمده از شبکه عصبی را بررسی می­کنیم. ابتدا در جدول شماره 2، داده­های نوع اول یعنی اندیکاتور ها را به شبکه ها وارد می­کنیم. نمودار همگرایی و خطای آموزش (MSE) نسبت به تعداد تکرار نتایج این شبکه در جدول شماره 3 نمایش داده می­شود. در ادامه، در جدول شماره 4، داده­های نوع دوم (قیمت پایانی 60 روز گذشته) را استفاده می­کنیم و شبکه را با آن آموزش می­دهیم. و در نهایت در جدول شماره 5، نمودار همگرایی و خطای آموزش (MSE) نسبت به تعداد تکرار نتایج این آزمایش را بررسی و مشاهده خواهیم کرد. شایان ذکر است که در بعضی از نمودار ها با توجه به حضور داده اعتبار سنجی نمودار دارای دو خط روند خواهد بود که خط روند قرمز نمایش دهنده خطای آموزش (MSE) و خط روند آبی نمایش دهنده خطای اعتبار­سنجی (MSE) می­باشد.

Table 2 - اطلاعات بدست آمده از اجرای الگوریتم های شبکه عصبی با داده¬های بدست آمده از 10 اندیکاتور و مقایسه با تنظیمات مختلف و دو نوع شبکه عصبی متفاوت

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | Net | Input Data | Input Dim | Epoch | Shuffle | Validation split (0-1) | MSE Train | MSE TEST | RMSE TEST | MSE Prediction | RMSE Prediction | MAE Prediction |
| 1 | Only Dense | 1712 | 10 | 10 | False | 0.0 | 0.2436 | 0.6989 | 0.836 | 0.6676 | 0.8171 | 0.8155 |
| 2 | Only Dense | 1712 | 10 | 10 | True | 0.0 | 4.7219e-04 | 2.0740e-04 | 1.4401e-02 | 1.2224e-04 | 1.1056e-02 | 8.9696e-03 |
| 3 | Only Dense | 1369 (Val 343) | 10 | 10 | True | 0.2 | 4.1641e-04 (6.8945e-04) | 4.3658e-04 | 2.0895e-02 | 3.9148e-04 | 1.9786e-02 | 1.7603e-02 |
| 4 | Dense & LSTM | 1712 | 10 | 10 | False | 0.0 | 0.2436 | 0.6989 | 0.836 | 0.6676 | 0.8171 | 0.8155 |
| 5 | Dense & LSTM | 1712 | 10 | 10 | True | 0.0 | 9.4328e-04 | 9.9499e-04 | 3.1543e-02 | 6.7216e-04 | 2.5926e-02 | 2.3169e-02 |
| 6 | Dense & LSTM | 1369 (Val 343) | 10 | 10 | True | 0.2 | 7.1682e-04 (0.0020) | 4.8713e-04 | 2.2071e-02 | 4.8981e-04 | 2.2132e-02 | 1.7788e-02 |
| 7 | Only Dense | 1712 | 10 | 50 | False | 0.0 | 0.2436 | 0.6989 | 0.836 | 0.6676 | 0.8171 | 0.8155 |
| 8 | Only Dense | 1712 | 10 | 50 | True | 0.0 | 3.6827e-04 | 2.8378e-04 | 1.6846e-02 | 2.0187e-04 | 1.4208e-02 | 1.1671e-02 |
| 9 | Only Dense | 1369 (Val 343) | 10 | 50 | True | 0.2 | 3.0650e-04 (6.3284e-04) | 3.9314e-04 | 1.9828e-02 | 3.4731e-04 | 1.8636e-02 | 1.6011e-02 |
| 10 | Dense & LSTM | 1712 | 10 | 50 | False | 0.0 | 0.2436 | 0.6989 | 0.836 | 0.6676 | 0.8171 | 0.8155 |
| 11 | Dense & LSTM | 1712 | 10 | 50 | True | 0.0 | 3.5539e-04 | 4.0943e-04 | 2.0234e-02 | 1.9477e-04 | 1.3956e-02 | 1.1479e-02 |
| 12 | Dense & LSTM | 1369 (Val 343) | 10 | 50 | True | 0.2 | 2.2359e-04 (2.2359e-04) | 6.2253e-04 | 6.2253e-04 | 3.8422e-04 | 1.9601e-02 | 1.6784e-02 |

با توجه به نتایج جدول شماره 2 می­توان اظهار کرد، نتایج در تعداد تکرار پایین در شبکه پرسپترون (Only Dense) نتیجه بهتری دارد اما در صورتی که تعداد تکرار بیشتر شود، شبکه عصبی بازگشتی و دارای حافظه (Dense & LSTM) نتایج بهتری می­دهد. البته باید ذکر کرد که اگر داده­ها به ترتیب به شبکه وارد شود خطای شبکه بسیار زیاد خواهد بود. با توجه به اینکه ویزگی­های مورد استفاده شبکه برای هر قیمت پایانی قبلا به صورت مجزا توسط اندیکاتور­ها استخراج شده است، می­توان با بهم ریختن داده­ها قدرت یادگیری شبکه را افزایش و خطای آنرا کاهش دهیم. در نظر داشته باشید، درصورتی که داده­ها به تریب و مرتب به شبکه وارد شوند، ممکن است شبکه آنهارا حفظ کند و یا نتواند آنهارا به درستی یاد بگیرد. برای دوری از این مشکل، ترتیب داده­ها را بهم خواهند ریخت.

Table 3 - تصاویر نمودارهای همگرایی شبکه¬ها عصبی در جدول 2 (نمودار خطای MSE نسبت به تعداد تکرار) (در شکل¬های ضریب 3، نمودار قرمز میزان خطای آموزش و نمودار آبی میزان خطای اعتبار سنجی می¬باشد.)

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 2 | 1 | شماره در جدول 2 |
|  |  |  | تصویر همگرایی شبکه (روند نمودار با ضریب 0.6 ملایم شده است) |
| 6 | 5 | 4 | شماره در جدول 2 |
|  |  |  | تصویر همگرایی شبکه (روند نمودار با ضریب 0.6 ملایم شده است) |
| 9 | 8 | 7 | شماره در جدول 2 |
|  |  |  | تصویر همگرایی شبکه (روند نمودار با ضریب 0.6 ملایم شده است) |
| 12 | 11 | 10 | شماره در جدول 2 |
|  |  |  | تصویر همگرایی شبکه (روند نمودار با ضریب 0.6 ملایم شده است) |

در ادامه نتایج داده­های نوع دوم (قیمت پایانی 60 روز گذشته) را بررسی می­کنیم.

Table 4 - اطلاعات بدست آمده از اجرای الگوریتم های شبکه عصبی با داده های بدست آمده از قیمت پایانی 60 روز گذشته و مقایسه با تنظیمات مختلف و دو نوع شبکه عصبی متفاوت

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | Net | Input Data | Input Dim | Epoch | Shuffle | Validation split (0-1) | MSE Train | MSE TEST | RMSE TEST | MSE Prediction | RMSE Prediction | MAE Prediction |
| 1 | Only Dense | 1705 | 60 | 10 | False | 0.0 | 0.2452 | 0.6995 | 0.8364 | 0.6676 | 0.8171 | 0.8155 |
| 2 | Only Dense | 1705 | 60 | 10 | True | 0.0 | 7.8285e-04 | 5.0915e-04 | 2.2564e-02 | 2.7219e-04 | 1.6498e-02 | 1.3633e-02 |
| 3 | Only Dense | 1364 (Val 341) | 60 | 10 | True | 0.2 | 7.6015e-04 (7.9220e-04) | 4.4485e-04 | 2.1091e-02 | 2.2595e-04 | 1.5032e-02 | 1.2002e-02 |
| 4 | Dense & LSTM | 1705 | 60 | 10 | False | 0.0 | 0.2452 | 0.6995 | 0.8364 | 0.6676 | 0.8171 | 0.8155 |
| 5 | Dense & LSTM | 1705 | 60 | 10 | True | 0.0 | 3.1054e-04 | 3.5990e-04 | 1.8971e-02 | 2.7321e-04 | 1.6529e-02 | 1.4474e-02 |
| 6 | Dense & LSTM | 1364 (Val 341) | 60 | 10 | True | 0.2 | 2.9200e-04 | 2.2947e-04 | 1.5148e-02 | 1.9573e-04 | 1.3990e-02 | 1.1860e-02 |
| 7 | Only Dense | 1705 | 60 | 50 | False | 0.0 | 0.2452 | 0.6995 | 0.8364 | 0.6676 | 0.8171 | 0.8155 |
| 8 | Only Dense | 1705 | 60 | 50 | True | 0.0 | 4.4717e-04 | 3.4953e-04 | 3.4953e-04 | 1.4797e-04 | 1.2164e-02 | 9.8938e-03 |
| 9 | Only Dense | 1364 (Val 341) | 60 | 50 | True | 0.2 | 4.2287e-04 (6.7614e-04) | 1.2375e-03 | 3.5179e-02 | 8.7406e-04 | 2.9565e-02 | 2.7193e-02 |
| 10 | Dense & LSTM | 1705 | 60 | 50 | False | 0.0 | 0.2452 | 0.6995 | 0.8364 | 0.6676 | 0.8171 | 0.8155 |
| 11 | Dense & LSTM | 1705 | 60 | 50 | True | 0.0 | 2.3175e-04 | 3.9412e-04 | 1.9853e-02 | 3.2265e-04 | 1.7962e-02 | 1.6005e-02 |
| 12 | Dense & LSTM | 1364 (Val 341) | 60 | 50 | True | 0.2 | 2.0905e-04 (2.9087e-04) | 1.7912e-04 | 1.3384e-02 | 1.2543e-04 | 1.1199e-02 | 8.9713e-03 |

Table 5 - تصاویر نمودارهای همگرایی شبکه ها عصبی در جدول 4 (نمودار خطای آموزش MSE نسبت به تعداد تکرار) (در شکل¬های ضریب 3، نمودار قرمز میزان خطای آموزش و نمودار آبی میزان خطای اعتبار سنجی می¬باشد.)

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 2 | 1 | شماره در جدول 4 |
|  |  |  | تصویر همگرایی شبکه (روند نمودار با ضریب 0.6 ملایم شده است) |
| 6 | 5 | 4 | شماره در جدول 4 |
|  |  |  | تصویر همگرایی شبکه (روند نمودار با ضریب 0.6 ملایم شده است) |
| 9 | 8 | 7 | شماره در جدول 4 |
|  |  |  | تصویر همگرایی شبکه (روند نمودار با ضریب 0.6 ملایم شده است) |
| 12 | 11 | 10 | شماره در جدول 4 |
|  |  |  | تصویر همگرایی شبکه (روند نمودار با ضریب 0.6 ملایم شده است) |

با مشاهده نتایج جدول شماره 4، در میابیم که شبکه عصبی بازگشتی و دارای حافظه (Dense & LSTM) نتایج بهتری را در هر دو نوع تعداد تکرار رقم زده و نسبت به شبکه پرسپترون (Only Dense) بهتر عمل کرده است. البته باید عنوان کرد که نتایج شبکه عصبی بازگشتی و دارای حافظه (Dense & LSTM) در این نوع داده نسبت به جدول شماره 2، که با داده اندیکاتور­ها آموزش دیده است بهتر می­باشد.

# **نتیجه گیری:**

از آنجا عمل پیش­بینی بازار سرمایه با توجه به نوسانات و متغیرات این بازار بسیار سخت می­باشد و در اکثر مواقع با خطا مواجه است، استفاده از این نوع مدل­ها بصورت جداگانه و تمام اختیار پیپنهاد نمی­شود. در مقالات و پیشنهادات بسیاری، مدل­های مختلفی ارائه و پیشنهاد شده­اند اما همه آنها در بزرگترین و حساس­ترین نقطه بازار که رفتار جامعه و بازخورد های سیاسی و تاثیرات اقتصادی است ضعف دارند و این بخش مهم ترین قسمت برای پیش­بینی روند بازار است.

با اینحال می­توان بیان کرد که نتایج هر دو شبکه عصبی از مدل SVR بهتر بوده و پیش­بینی دقیق­تری را به ما می­دهد. در بین شبکه­های عصبی، به طور کلی شبکه عصبی بازگشتی و دارای حافظه (Dense & LSTM) که توسط ما پیشنهاد شد نسبت به شبکه پرسپترون (Only Dense) که در مقاله پیشنهاد شده بود نتایج بهتری را رقم زد. از بین مدل­های ورودی و داده­ها نیز مدل دوم (قیمت پایانی 60 روز گذشته) که توسط ما معرفی شده بود، عملکرد و تاثیر بهتری بر مدل های استفاده شده داشت و نتایج بهتری را در خروجی ارائه داد.

در نهایت پیشنهاد می­شود، برای اینکه بتوانید در بازار سرمایه پیش­بینی درستی را بدست آورید، بهتر است به سراغ زبان mlq و مشاور خبره (expert advisor) بروید و نتایج را در محیط استاندارد بررسی و مشاهده کنید.

# **منابع:**

A.R.J. Fredo, G. K. (2015). Automated segmentation and analysis of corpus callosum in autistic MR brain. *Journal of Medical and Biological Engineering*.

F. Fumero, S. A.-H. (2011). RIM-ONE: An open retinal image database for optic. *in 24th International Symposium on Computer-Based Medical Systems (CBMS).*

*http://medimrg.webs.ull.es/research/retinal-imaging/rim-one/* . (n.d.). Retrieved from medimrg.

K. Manickavasagam, S. S. (2014). Development of systems for classification of different plasmodium species in. *Journal of Advanced Microscopy Research*.

*Machine\_Learning\_A-Z*. (n.d.). Retrieved from https://github.com/srafay/Machine\_Learning\_A-Z.

N.S.M. Raja, S. S. (2015). Improved PSO based multi-level thresholding for cancer infected breast thermal images. *Procedia Computer Science.*

N.S.M. Raja, V. R. (2014). Otsu based optimal multilevel image thresholding using firefly algorithm. *Modelling and Simulation in Engineering*.

S. Ramakrishnan, N. R. (2014). Analysis of vasculature detection in human retinal images using bacterial foraging optimization based multi thresholding. *International Journal of Swarm Intelligence and Evolutionary Computation,*.

S.Hore, S. C.-N. (2016). An Integrated Interactive Technique for Image Segmentation using Stack based Seeded Region Growing and Thresholding. *International Journal of Electrical and Computer Engineering*.

V.S.Lakshmi, S. T. (2016). Chaotic cuckoo search and Kapur/Tsallis approach in segmentation of t.cruzi from blood smear images. *International Journal of Computer Science and Information Security (IJCSIS)*.