مجله‌ی بررسی‌های آمار رسمی ایران
سال ۳۲، شماره‌ی ۱، بهار و تابستان ۱۴۰۰، صص ۷۳ –۹۵

# پیش‌بینی قیمت سهام با استفاده از شبکه عصبی و جنگل تصادفی (مطالعه‌ی موردی: سهام بانک ملت)

**مریم محمدی*، حبیب جعفری و آزاد خانزادی**

دانشگاه زاری

**چکیده.** یکی از مسائل مهم در علم آمار پیش‌بینی مدل‌های غیرخطی است. در پژوهش حاضر با استفاده از مدل شبکه عصبی پرسپترون و مدل جنگل تصادفی به پیش‌بینی قیمت سهام بانک ملت طی ده سال بین سال‌های ۹۰ تا پایان ۹۹ پرداخته شده است. از معیار MAPE به‌عنوان معیار سنجش استفاده شده است. هر دو در حوزه یادگیری بانظارت توضیح داده می‌شود. از شاخص‌های تکنیکال مانند RSI، OBV، MACD، RW% و ... به عنوان متغیرهای مستقل استفاده شده است. یافته های تجربی مربوط به بررسی ده ساله به خوبی نشان می‌دهند که هر دو مدل به تنهایی قادر به پیش‌بینی قیمت سهام می‌باشند اما مدل شبکه عصبی عملکرد بهتری نسبت به جنگل تصادفی داشته است پس دارای قدرت بهتری در پیش‌بینی می‌باشد.

**واژه‌گان کلیدی**: شبکه عصبی، جنگل تصادفی، پیش‌بینی قیمت سهام، بانک ملت.

## ۱- مقدمه

مشخصه بازار مالی این است که یک سیستم پویا، پیچیده و غیر خطی، با داده‌های شدید، نویز، غیر ثابت، ماهیت بدون ساختار و با درجه بالایی از عدم قطعیت است[۱۰]. پیش‌بینی بازار سهام یک کار چالش‌برانگیز و مورد توجه محققین بوده است، زیرا بازار سهام در رفتار خود پر نوسان است[۲]. در اکثر اوقات پیش‌بینی قیمت سهام و اوراق بهادار مساله‌ای است که همیشه اقتصاددانان و سرمایه‌گذاران به دنبال بهینه‌سازی آن

* نویسنده عهده‌دار مکاتبات

دریافت: ۱۴۰۰/۱۱/۳، پذیرش: ۱۴۰۲/۶/۲۲.

بوده‌اند، تا در بهترین زمان و مکان مناسب سرمایه خود را افزایش دهند. که لازمه‌ی آن داشتن اطلاعات صحیح و اصولی از بازار بورس و تغییرات سهام است. اکثر افراد از پیش‌بینی برای تصمیم‌گیری و برنامه‌ریزی برای زندگی آینده استفاده می‌کنند. بنابراین، سرمایه‌گذار نیازمند ابزارهای قدرتمند و قابل اعتماد است تا از طریق آن به پیش‌بینی قیمت سهام بپردازد [۱۳]. پژوهش‌گران در داده‌های رگرسیونی از روش‌های یادگیری ماشین[۱] علاوه بر روش‌های آماری که بتواند مسائل غیرخطی را حل کنند بهره می‌برند. یادگیری ماشین یکی از پرکاربردترین شاخه‌های هوش مصنوعی[۲] می‌باشد. یادگیری ماشین شامل یادگیری بانظارت[۳]، بدون نظارت[۴]، نیمه نظارتی[۵] و تقویتی[۶] می‌باشد. هر دو مدل پژوهش در حوزه یادگیری بانظارت می‌باشند.

پیش‌بینی قیمت سهام وبملت مبتنی بر شبکه عصبی[۷] و جنگل تصادفی نشان می‌دهد که شبکه‌های عصبی قدرت پیش‌بینی قیمت‌ها را با وجود نوسانات در بازارها دارند.

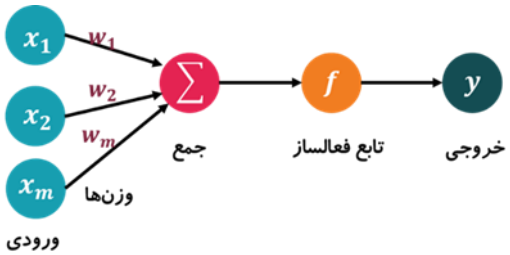
پیش‌بینی قیمت شاخص سهام به دلیل نوسانات و ویژگی‌های نویز ، همواره یکی از چالش‌برانگیزترین کارها برای افرادی است که در حوزه مالی و سایر مقیاس‌های مرتبط با آن کار می‌کنند [۳]. یک سوال باز در جامعه مدرن این است که چگونه می‌توان دقت پیش‌بینی قیمت شاخص سهام را بهبود بخشید. تعریف سری‌های زمانی را می‌توان به عنوان یک دنباله زمانی خلاصه کرد که شامل داده‌های مشاهده شده از رفتار یا فعالیت دوره‌ای اختیاری در برخی زمینه‌ها مانند آمار، علوم اجتماعی، مالی، مهندسی و اقتصاد هستند. قیمت شاخص سهام نوعی سری زمانی مالی است که دارای نسبت سیگنال به نویز پایین [۱۰] و توزیع سنگین [۵] است.

این نوع ویژگی‌های پیچیده، پیش‌بینی روند قیمت شاخص را بسیار دشوار می‌کند. هدف پیش‌بینی سری‌های زمانی ساخت مدل‌هایی برای شبیه‌سازی مقادیر آینده با توجه به مقادیر گذشته آن‌ها است [۷]. بنابراین، افراد با استفاده از روش‌های یادگیری عمیق مانند شبکه‌های عصبی مصنوعی(ANN) شروع به تجزیه و تحلیل روابط غیرخطی می‌کنند [۹].

در یادگیری ماشین، ابتدا سیستم مجموعه داده را دریافت کرده وسپس الگوریتم توسط داده مشخص می‌شود و در نهایت آموزش می‌بینند. در این پژوهش، به مطالعه برخی الگوریتم‌های یادگیری بانظارت پرداخته می‌شود. در این نوع الگوریتم دو نوع از متغیرهای مستقل و وابسته وجود دارد که متغیرهای مستقل یا ورودی ها $x_1.x_2.x_3.\ldots.x_p$ یک یا چند متغیر هستند و بر اساس مقادیر آنها باید متغیر وابسته یا خروجی Y را به کمک الگوریتم‌ها پیش‌بینی کنیم. در این پژوهش به معرفی دو الگوریتم شبکه عصبی و جنگل تصادفی می‌پردازیم که هر دو از پرکاربردترین الگوریتم‌های یادگیری با نظارت می‌باشند.

## ۲- شبکه‌های عصبی پرسپترون چند لایه[۸]

پرسپترون چندلایه نوعی شبکه عصبی مصنوعی است که از تعداد محدودی لایه پیوسته تشکیل شده است. یک MLP حداقل دارای سه لایه شامل لایه ورودی، لایه پنهان، و لایه خروجی در بسیاری از موارد، لایه‌های پنهان بیشتری در ساختار MLP وجود دارد که می‌تواند با راه‌حل‌های تقریبی برای بسیاری از مسائل پیچیده مانند تقریب برازش برخورد کند [۷]. یک MLP را می توان به عنوان یک گراف جهت‌دار در نظر گرفت که مجموعه‌ای از بردارهای ورودی را به مجموعه‌ای از بردارهای خروجی نگاشت می‌کند که از چندین لایه گره متصل به لایه بعدی تشکیل شده است. این اتصالات عموماً سیناپس‌ها نامیده می‌شوند [۱۲].




**شکل ۱-** ساختار ریاضی نرون

ورودی با x، وزن‌ها با w، تابع مجموع با ∑، تابع فعال سازی[۹] با f و خروجی با yنشان داده شده است.

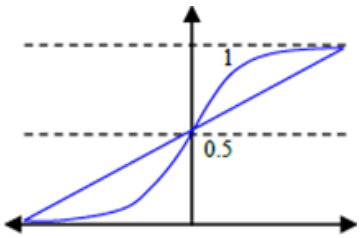
تابع مجموع، جمع وزن‌دار ورودی‌ها را با رابطه زیر محاسبه می‌کند:

$$Net = (w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots) = \sum_{i=1}^{m} w_i x_i \quad (۱)$$

علاوه بر گره‌های ورودی، هر گره یک نورون با تابع فعال‌سازی غیرخطی است [۷].

رابطه بین سطح فعال شدن و خروجی با استفاده از تابع تبدیل بیان می‌شود که دارای انواع مختلفی از جمله تانژانت هیپربولیک، سیگموئید و ...می‌باشد. یکی از پرکاربردترین توابع در پیش‌بینی، تابع لجستیک است:

$$F(x) = \frac{1}{1 + e^{-x}} \quad (۲)$$



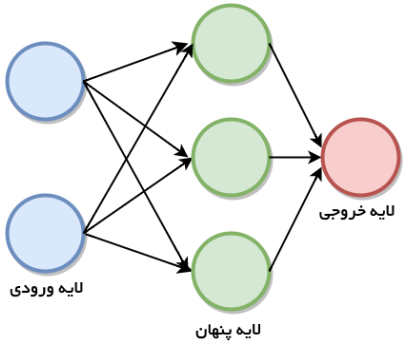

**شکل۲-** نمودار سیگوئید

مقدار تابع لجستیک در بازه ۰ و ۱ متغیر است. مقادیر ورودی در این تابع انتقال در محدوده ∞− تا ∞+ است.

## ۲-۱- شبکه‌های پیشخور

در شبکه‌های پیش‌خور[۱۰]، مسیر پاسخ همیشه رو به جلو پردازش می‌شود و به نرون‌های لایه[۱۱]های قبل باز نمی‌گردد. در این نوع شبکه‌ها سیگنال‌ها اجازه دارند از مسیر یکطرفه

ورودی تا خروجی عبور کنند. در بدن انسان نیز، پیام‌های عصبی به صورت یکطرفه حرکت می‌کنند: از دنریت به بدنه سلول و سپس به آکسون. ساده‌ترین این شبکه‌ها، شبکه‌های پرسپترون[۱۲] هستند.




**شکل ۳-** شمایی از شبکه پیشخور

در این نوع شبکه‌ها لایه اول به عنوان لایه ورودی، لایه‌های میانی به عنوان لایه پنهان، لایه آخر نیز به عنوان لایه خروجی نامیده می‌شوند.

## ۲-۲- الگوریتم پس انتشار خطا

یک روش یادگیری نظارت شده به نام الگوریتم پس انتشار اغلب برای آموزش MLP استفاده می‌شود. MLP تعمیم پرسپترون است که بر این ضعف غلبه می‌کند که پرسپترون نمی‌تواند داده‌های خطی جدانشدنی را تشخیص دهد. لایه‌های MLP کاملاً به هم متصل هستند، به این معنی که هر نورون هر لایه با تمام نورون‌های لایه قبلی مرتبط است و این اتصال نشان‌دهنده جمع و جمع وزن‌ها است [۷].

شبکه عصبی به دنبال یادگیری از طریق تغییرات در وزن‌ها و اریبی می‌باشد. اصل یادگیری در شبکه عصبی نیز بر پایه تکرار است. داده‌ها چندین بار به شبکه وارد می‌شود و شبکه با کم و زیاد کردن اوزان می‌تواند بهترین مدل را تخمین بزند. در هر تکرار دو مرحله

وجود دارد. در مرحله اول حرکت رو به جلو است، بردار ورودی به شبکه پرسپترون چندلایه وارد می‌شود و از طریق لایه‌های میانی تأثیرات آن به لایه‌های خروجی انتشار می‌یابد.
بردار خروجی تشکیل یافته در لایه خروجی، پاسخ واقعی شبکه پرسپترون چندلایه را تشکیل می‌دهد. در این مسیر پارامترهای شبکه ثابت و بدون تغییر در نظرگرفته می‌شود. مسیر دوم موسوم به مسیر برگشت است. در این مسیر برخلاف مسیر رفت پارامترهای شبکه پرسپترون چندلایه تغییر و تنظیم می‌گردد.

$$e_j(n) = y_j(n) - \hat{y_j}(n) \quad (۳)$$

اگر اثر خطا برای نرون j ام برابر با $\frac{1}{2}e_1^2(\mathrm{n})$ تعریف کنیم، انرژی خطای کل(برای تمام نرون‌های لایه خروجی) عبارت است از:

$$E(n) = \frac{1}{2}\sum_{j=1}^{c} e_j^2(n)$$

که c تعداد سلول‌های خروجی است.
الگوریتم پس انتشار خطا، تصحیح وزن‌ها $\Delta W_\pi(\mathrm{n})$ را به وزن $W_\pi(n)$ اعمال می‌کند. این تصحیح وزن با مشتق جزئی E (n) نسبت به $W_\pi(\mathrm{n})$ متناسب می‌باشد، یعنی:

$$\frac{\delta E(n)}{\delta W_\pi}$$

مشتق جزئی وزن مناسب را در فضای وزن‌ها تعیین می‌کند.
تعیین تعداد لایه های پنهان و تعداد گره در هر لایه و نوع تابع فعال سازی سعی و خطا می باشد. در نهایت، الگوریتم BP و الگوریتم بهینه‌سازی برای به روز رسانی وزن W و اریبی B برای به دست آوردن نتایج مورد انتظار استفاده می‌شود [۷].

## ۳- جنگل تصادفی

جنگل تصادفی[۱۳] جز الگوریتم‌های گروهی و در حوزه بانظارت است که توسط بریمن[۱۴] [۴] معرفی شده است. مطالعات نشان می‌دهد که در بعضی مواقع عملکرد الگوریتم‌های

یادگیری گروهی[۱۵] بهتر از استفاده از یک الگوریتم است[1]. جنگل تصادفی شامل مجموعه‌ای از درخت‌های هرس نشده است که مجموعه بزرگی از درخت‌های مستقل و بدون وابستگی به یکدیگر را می‌سازد و سپس از خروجی آن‌ها میانگین می‌گیرد. فرض کنید S یک مجموعه داده شامل n مشاهده و p متغیر پیشگو می‌باشد. G مجموعه داده‌ی آموزشی با m مشاهده که(m<n) است. به صورت زیر:

۱- از مجموعه داده‌ی آموزشی G به روش باز نمونه‌گیری خودگردان یک مجموعه داده‌ی آموزشی تولید می‌شود.

۲- از مجموعه داده‌ی آموزشی تولید شده که به روش باز نمونه‌گیری خودگردان انجام شد، q ویژگی که متغیر پیشگو می‌باشد به روش نمونه‌گیری تصادفی بدون جایگذاری ازp ویژگی(q<p) انتخاب می‌گردد.

۳- درخت تصمیم روی این مجموعه داده‌ی آموزشی تولید شده با q ویژگی که متغیر پیشگو می‌باشد، پیاده‌سازی می‌شود.

این مراحل یک تا سه B بار تکرار می‌گردد.

این الگوریتم روش جعبه[۱۶] سیاه نامیده می‌شود چون نمی‌توان رابطه‌ی بین پیش‌بین‌ها و خروجی‌ها را به طور جداگانه برای هر درخت از درختان جنگل تصادفی آزمون کرد.

در این الگوریتم از روش بازنمونه‌گیری خودگردان برای تولید چندین مجموعه داده آموزش متفاوت استفاده می‌گردد.

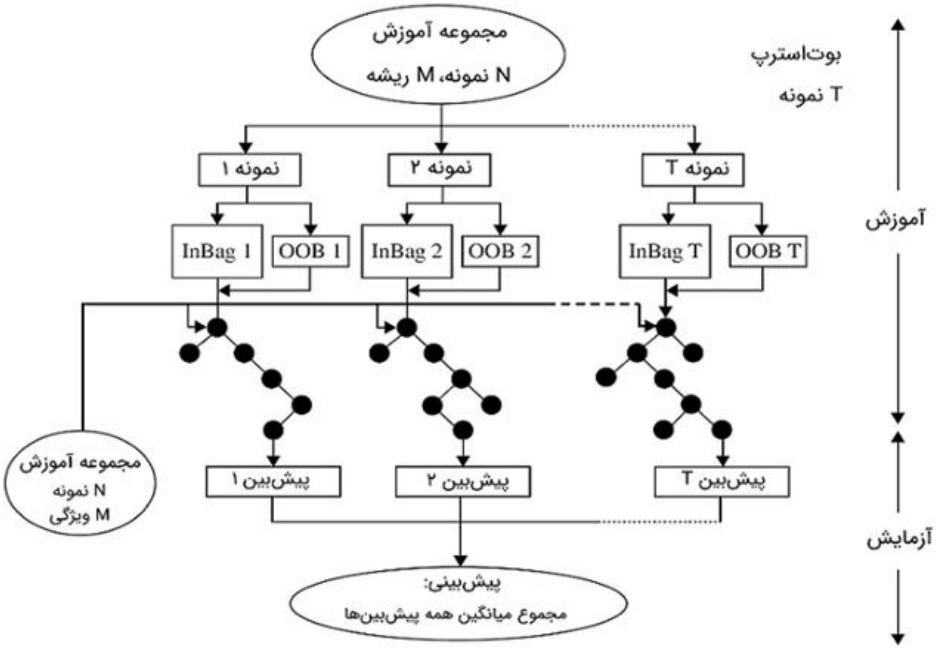
## ۳-۱- روش باز نمونه‌گیری خودگردان

افرون[۱۷] [۶] در سال‌های ۱۹۷۹ و ۱۹۸۲ روش باز نمونه‌گیری خود گردان را معرفی کرد. فرض کنید $c_1 , c_2 , \dots , c_n$ نمونه‌ای تصادفی و دارای توزیع نامعلوم F باشد. از متداول‌ترین برآوردگرهای تابع توزیع تجمعی نامعلوم جامعه که براساس نمونه تصادفی در نظر گرفته می‌شود، تابع توزیع تجربی است که به صورت زیر تعریف می‌شود:

$$\widehat{Fn}(c) = 1/n \sum_{i=0}^{n} I(Ci < c) \quad (۴)$$

اگر پیشامد A اتفاق بیافتد، I(A)=1 و اگر پیشامد A اتفاق نیفتد I(A)=0. که با توجه به مجهول بودن توزیع F ، توزیع T نامعلوم می‌باشد. حال آماره T=T(c) در نظر بگیرید. با استفاده از روش باجایگذاری از نمونه‌ی اولیه $C_1, C_2, \dots, C_n$ با استخراج نمونه‌هایی به حجم n، می‌توان نمونه‌ای از T را داشته باشیم که بر اساس آن ویژگی‌های توزیع را مورد بررسی قرار می‌دهیم. زمانی که B بار نمونه $C^*_{(i)}$ ، $i = 1,2, \dots, B$ از نمونه اولیه‌ی استخراج شود، در آن صورت نمونه $T(C^*_{(1)})$ ، . . . ، $T(C^*_{(B)})$ از توزیع T داریم.

در الگوریتم جنگل برخی از داده های آموزشی یعنی مشاهداتی که داریم در مجموعه داده تولید شده به روش باز نمونه گیری خودگردان انتخاب نمی گردندکه داده های OOB[۱۸]، خارج از کیسه می نامند.




**شکل ۴-** ساختار کلی از نمونه‌گیری خودگردان در الگوریتم جنگل‌تصادفی Guo et al.,2011

# ۴- تحلیل داده‌ها و نتایج

## ۴-۱- معیار درصد میانگین قدر مطلق خطا

شاخص‌های عملکرد مختلفی برای ارزیابی مدل‌های تخمین و پیش‌بینی وجود دارد. در این پژوهش از درصد میانگین قدر مطلق خطا در هر دو الگوریتم استفاده شده است. معیار درصد میانگین قدرمطلق خطا[۱۹] معیاری برای بدست آوردن بهترین مدل پیش‌بینی است که مورد تایید آماردانان نیز می‌باشد. این معیار از پرکاربردترین معیارهای بدون واحد است که به وسیله تعداد زیادی از محققین برای ارزیابی پیش‌بینی انتخاب شده است. بنابراین مدلی که کم‌ترین MAPE را داشته باشد، انتخاب می‌شود. به صورت زیر محاسبه می‌گردد:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_{i-}\hat{Y}_i}{Y_i}\right| \times 100 \quad (۵)$$

در این رابطه n تعداد کل داده‌های مشاهداتی و $Y_i$ مقادیر مشاهداتی و $\hat{Y}_i$ مقادیر پیش‌بینی شده است.

## ۴-۲- مدل شبکه عصبی

مجموعه داده برای مدل شبکه عصبی شامل ۱۹۳۶ مشاهده است. متغیر وابسته برابر با مقدار قیمت پایانی(قیمت پایانی سهم بانک ملت) که در محاسبات با نماد endprize نشان داده شده است و متغیرهای مستقل عبارتند از:

- date: تاریخ(زمان)
- vollum: حجم(تعداد سهام مبادله شده مربوط به سهام بانک ملت در یک روز)
- prize: ارزش(ارزش معاملات بانک ملت در یک روز)
- yprize: قیمت روز قبل(قیمت روز قبل سهام وبملت)
- efficiency: بازدهی سهام بانک ملت
- transactions: ارزش معاملات(ارزش معاملات کل بازار سهام)

- number: تعداد معاملات(تعداد خرید و فروش بازار سهام)
- total: بازدهی کل سهام بازار
- mm: مجموع خریدار و فروشنده حقیقی و حقوقی

مدل مورد نظر را روی مجموعه داده آموزشی(training set) ایجاد و سپس روی مجموعه داده آزمون(test data) عملکرد آن را ارزیابی می‌کنیم. در این پژوهش ۰٫۸ داده‌ها برای آموزش و ۰٫۲ داده‌ها برای آزمون انتخاب شدند.

گاهی اوقات الگوریتم‌های سری زمانی کلاسیک برای ایجاد و پیش‌بینی‌های قوی، کفایت چندانی ندارند. در این صورت منطقی به نظر می‌آید که داده‌های سری زمانی را با ساخت ویژگی‌هایی از متغیر زمان یا همان تاریخ، به داده‌های یادگیری ماشین تبدیل کنیم. می‌توان ویژگی‌های زمان مانند:

Year- yday(day of the year)- quarter- month- day- weekdays - weekend and…

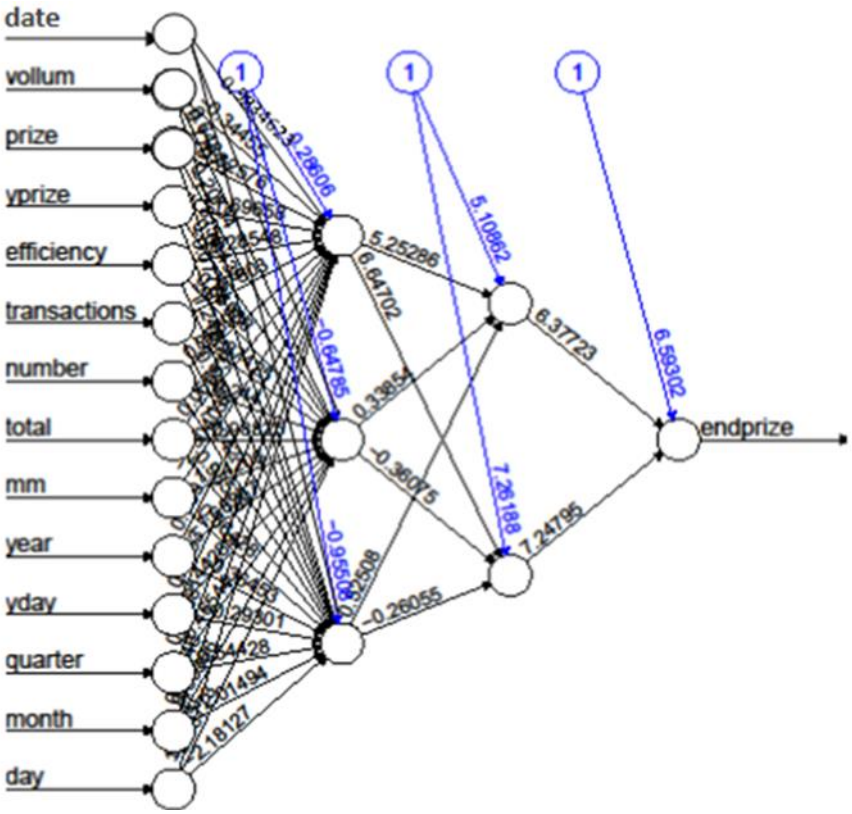
را ساخت. پس از ایجاد ویژگی‌های زمان، آن‌ها را به نوعی از داده مورد نظر خود تبدیل می‌کنیم.

با توجه به متغیرها مدل به دست آمده به صورت زیر است:

```
nn = neuralnet(endprize ~  date + vollum + prize + yprize + efficiency
+ transactions + number + total + mm + year + yday + quarter + month
+ day   ,train,linear.output = FALSE, hidden = c(3, 2))
```

در مدل شبکه عصبی فوق تعداد لایه‌های پنهان را برابر با دو لایه در نظر گرفته‌ایم طوری که در لایه پنهان اول ۳ نرون و در لایه پنهان دوم، ۲ نرون وجود داشته باشد (hidden = c(3, 2)). توجه شود که مدل فوق را روی داده‌های قسمت آموزش در فرآیند یادگیری ماشین ساخته‌ایم. تابع فعال‌سازی نیز برابر با لجستیک است. ماتریس نتایج نشان می‌دهد مقدار آستانه برای فعال‌سازی هر نرون برابر با 7.821488e-03 است به

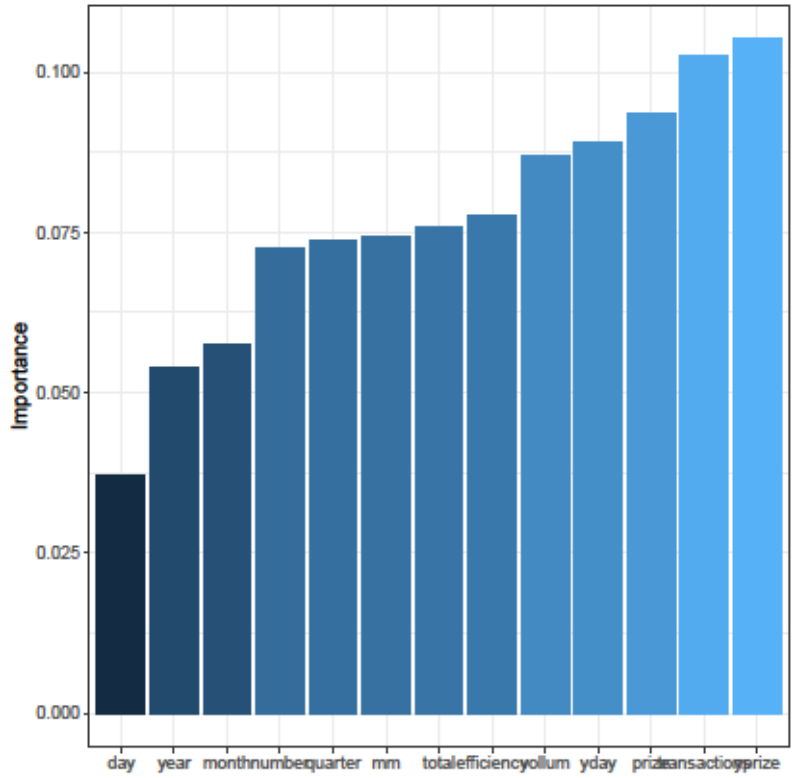
این معنی که هر نرون زمانی فعال خواهد بود که مقدار تابع فعال‌سازی آن از این مقدار بیشتر باشد و تعداد مراحل نیز برابر است با 7.400000e+01.




**شکل ۵-** نمودار شبکه عصبی با ۱۴ متغیر

نتایج عملکرد مدل نشان می‌دهد که معیار MAPE روی مجموعه آموزشی ۹۹/۹۴۶۰۵ درصد و روی داده‌های آزمون ۹۹/۹۴۶۳۵ درصد است. میزان نزدیکی و شباهت مقادیر MAPE به‌دست آمده روی مجموعه‌های train و test، شاخصی برای تشخیص قوی بودن مدل و هم‌چنین قدرت بالای تعمیم‌پذیری آن است.

یک تکنیک مهم برای بررسی میزان قدرت مدل به دست آمده این است که تنها متغیرهای مهم را در مدل وارد نماییم. برای اجرای این عمل می‌توان از تابع garson() موجود در پکیج "nnet" است بهره گرفت.



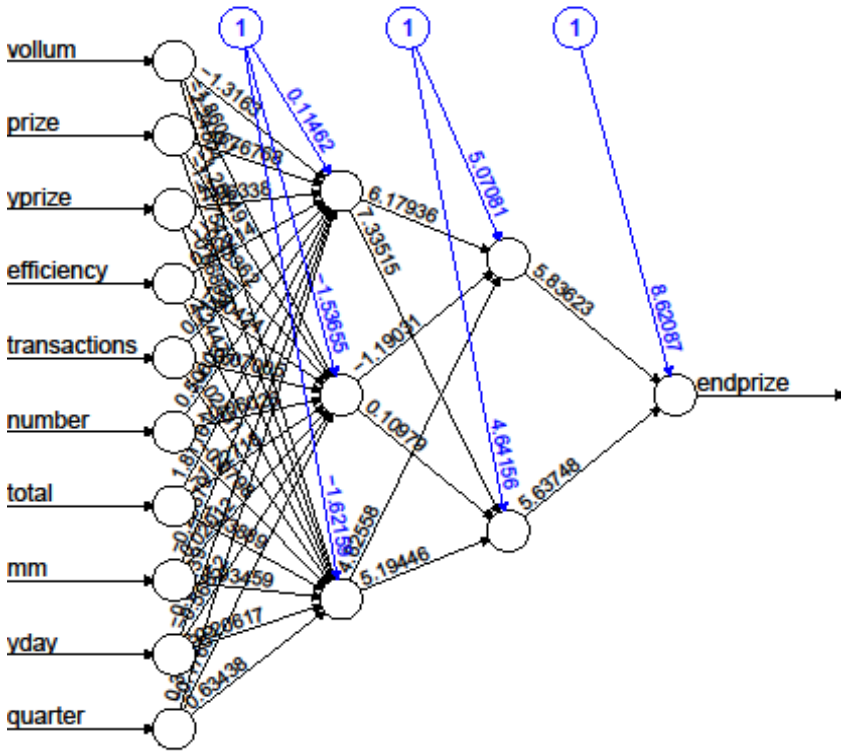

**شکل ۶-** نمودار میزان اهمیت متغیرهای مستقل بر متغیر وابسته

مطابق با نمودار فوق، متغیرهای day، year و month نسبت به سایر متغیرها تاثیر کمتری بر متغیر پاسخ داشته‌اند. لذا می‌توان آن‌ها را از مدل حذف نمود و سپس مقدار MAPE را بررسی کرد.

مدل بدست آمده با توجه به متغیرهای مورد نظر به صورت زیر است:

nn3= neuralnet(endprize ~ date + vollum + prize + yprize + efficiency + transactions + number + total + mm + yday + quarter ,train,linear.output = FALSE, hidden = c(3, 2))

ماتریس نتایج نشان می‌دهد که مقدار آستانه برای فعال‌سازی هر نرون برابر با 8.962055e-03 است به این معنی که هر نرون زمانی فعال خواهد بود که مقدار تابع فعال‌سازی آن از این مقدار بیشتر باشد و تعداد مراحل نیز برابر است با 6.600000e+01. تابع فعال‌سازی نیز برابر با لجستیک است.




**شکل ۷-** شمایی از ساختار شبکه عصبی به دست آمده با دو لایه پنهان و ۱۰ متغیر

با حذف متغیرهای یاد شده از مدل، مقدار معیار MAPE نسبت به مدل قبل بسیار کاهش یافته و برای داده train مقدار ۱/۰۳۵۴۱ و برای داده test مقدار ۱/۰۴۰۳۲ به دست آمده است. این مقدار برای دو مجموعه داده test و train بسیار به هم نزدیک هستند. لذا مدل دوم یک مدل قوی برای پیش‌بینی مقدار متغیر پاسخ و هم برای تعیم‌پذیری مسئله است.

## ۴-۳- مدل جنگل تصادفی

مجموعه داده برای مدل جنگل تصادفی شامل ۱۹۶ مشاهده است. متغیر وابسته برابر است با قیمت نهایی سهام که در محاسبات با نماد response نشان داده شده است و متغیرهای مستقل نیز عبارت هستند از:

- تاریخ یا زمان(date)
- شاخص میانگین متحرک همگرا، واگرا(macd)
- شاخص قدرت نسبی(rsi)
- شاخص حجم متوازن یا تعادلی(obv)
- شاخص تصادفی(so)
- شاخص درصد آر ویلیامز(wpr)
- شاخص بازده(Yeild with Exel%)

مدل مورد نظر را روی مجموعه داده آموزشی(training set) ایجاد و سپس روی مجموعه داده آزمون(test data) عملکرد آن را ارزیابی می‌کنیم. در این پژوهش ۰/۸ داده‌ها به صورت تصادفی برای آموزش و ۰/۲ داده‌ها برای آزمون انتخاب شدند.

گاهی اوقات الگوریتم‌های سری زمانی کلاسیک برای ایجاد و پیش‌بینی های قوی، کفایت چندانی ندارند. در این صورت منطقی به نظر می‌آید که داده‌های سری زمانی را با ساخت ویژگی‌هایی از متغیر زمان یا همان تاریخ، به داده‌های یادگیری ماشین تبدیل کنیم. می‌توان ویژگی‌های زمان مانند:

Year- yday(day of the year)- quarter- month- day- weekdays - weekend and…

را ساخت. پس از ایجاد ویژگی‌های زمان، آن‌ها را به نوعی از داده مورد نظر خود تبدیل می‌کنیم.

در این‌جا جنگل تصادفی ما با توجه به متغیر پاسخ از نوع رگرسیون است. خلاصه مدل اجرا شده به صورت زیر است:

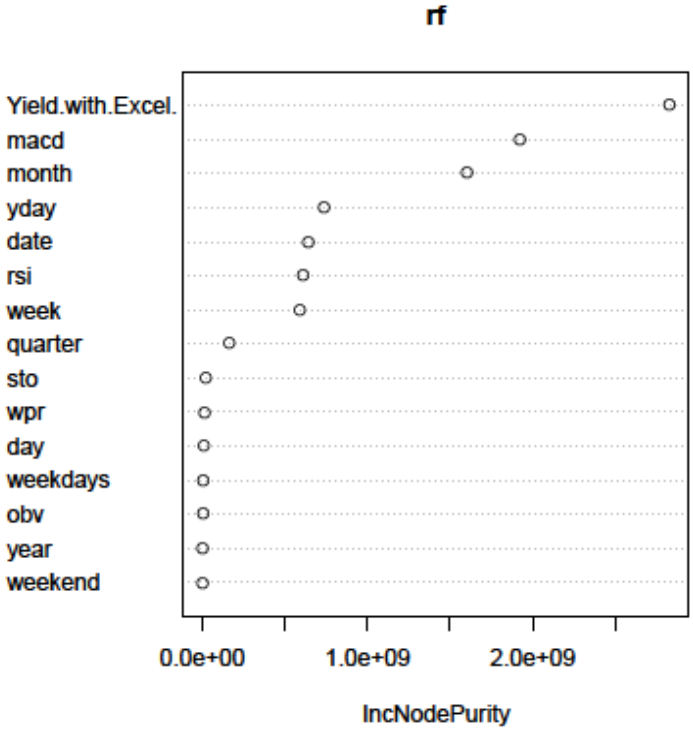
randomForest(formula = response ~ data + Yield.with.Excel.+ macd + rsi + sto + wpr + obv + year + yday + quarter + month + day + weekdays + weekend + week, data = train)

نوع جنگل تصادفی در این پژوهش از نوع رگرسیون، تعداد درختان موجود در آن ۵۰۰ عدد و تعداد متغیرها در هر جداسازی برابر با ۵ است.

حال عملکرد مدل بر روی مجموعه آموزش و آزمون را ارزیابی می‌کنیم. نتایج MAPE روی داده آموزش ۸۵% است در حالی‌که این مقدار روی داده آزمون به ۱/۹% افزایش یافته است.

یک تکنیک مهم برای بررسی میزان قدرت مدل به دست آمده این است که متغیرهای مهم را در مدل وارد نماییم. زیرا هر میزان MAPE به دست آمده برای مجموعه داده train و test کوچک و به هم نزدیک باشند، مدل مناسب‌تری جهت پیش‌بینی خواهیم داشت. نکته‌ای که باید به آن توجه داشت این است که معیار خطا بر روی داده‌های آزمون، نسبت به معیار مشابه در داده آموزش، ملاک معتبرتری است زیرا با پیچیده‌تر شدن مدل، همواره خطای آموزش کاهش پیدا می‌کند و تصمیم‌گیری بر اساس آن اشتباه است، در حالی که خطای آزمون با ورود متغیرهای نامناسب به مدل، کاهش پیدا نمی‌کند و مدل با کم‌ترین مقدار خطای آزمون بهترین مدل را نتیجه می‌دهد. در مدل جنگل تصادفی برای تعیین وجود یا عدم وجود یک متغیر در مدل، میزان اهمیت تک تک متغیرها در مدل را ارزیابی

می‌کنند. برای اجرای این عمل می‌توان از تابع varImpPlot بهره گرفت طوری که IncNodePurity در محور افقی نمودار زیر بیانگر این معیار در درختان رگرسیون است:




**شکل ۸-** نمودار مربوط به متغیرهای اولیه موجود در مدل جنگل‌تصادفی

خروجی این تابع متغیرهایی که مقدار IncNodePurity آن‌ها حدودا بیش از $1.0e^{+09}$ باشد در رده متغیرهایی که حضور آن‌ها در مدل مهم است قرار می‌گیرند و برعکس متغیرهایی که مقدار IncNodePurity آن‌ها کم‌تر از مقدار گفته شده باشد باید از مدل حذف گردند. بنابراین مطابق با نمودار فوق متغیرهای زیر به ترتیب اهمیت آورده شده‌اند:

Yeild with excel- macd – month

قدم بعدی ساخت مدل جنگل تصادفی اصلاح شده با استفاده از متغیرهای فوق است.

randomForest(formula = response ~ Yield.with.Excel. + macd , data = train)

نوع جنگل تصادفی در این پژوهش از نوع رگرسیون، تعداد درختان موجود در آن ۵۰۰ عدد و تعداد متغیرها در هر جداسازی برابر با یک است.

مرحله بعد ارزیابی بر روی داده‌های train و test است. نتایج نشان می‌دهد که معیار MAPE روی مجموعه آموزشی ۱٫۳% و روی داده‌های آزمون ۱٫۷۵% است. میزان نزدیکی مقادیر MAPE به دست آمده روی مجموعه‌های train و test، شاخصی برای تشخیص قوی بودن مدل و هم‌چنین قدرت بالای تعمیم پذیری آن است. هم‌چنین مقدار کاهش رویداده در مقدار این معیار بر روی مجموعه داده آزمون از ۱٫۹% به ۱٫۷% ، نشان می‌دهد که مدل اصلاح شده عملکرد بهتری نسبت به مدل قبل دارد.

## ۴-۴- نتیجه‌گیری و مقایسه نتایج پیش‌بینی

با توجه به بازه‌ی زمانی و داده‌های استفاده شده در این پژوهش و براساس معیار سنجش (MAPE) انتخاب شده در جدول زیر ملاحظه می‌کنیم که مقدار معیار MAPE برای مدل شبکه عصبی نسبت به مدل جنگل تصادفی کمتر است و علاوه بر این مقدار این معیار برای دو مجموعه داده test و train برای مدل شبکه عصبی نسبت به مدل جنگل تصادفی به هم نزدیک‌تر هستند و لذا مدل شبکه عصبی عملکرد بهتری در پیش‌بینی قیمت سهام بانک ملت داشته است.

| Model | MAPE(train) | MAPE(test) |
|---|---|---|
| **Random forest** | 1.302628 | 1.753767 |
| **Neural network** | 1.03541 | 1.04032 |

لذا هر دو مدل شبکه عصبی و جنگل تصادفی با توجه به مقادیر MAPE به دست آمده، به تنهایی دارای توانایی در پیش‌بینی قیمت سهام بانک ملت می‌باشند و نتایج نیز به خوبی نشان می‌دهد که شبکه عصبی دارای قدرت بهتری در پیش‌بینی قیمت سهام است.

## توضیحات

1. Mashin Learning
2. Artificial Intelligence
3. Supervised
4. Un Supervised
5. Semi-regulatory
6. Reinforcement
7. Neural Network
8. Multilayer Perceptron Neural Networks
9. Activation Function
10. Feeder Networks
11. Layer
12. Perceptron
13. Random Forest
14. Breiman
15. Ensemble Learning
16. Black Box
17. Efron
18. Out Of Bag
19. Mean Absolute Percentage Error (MAPE)

# مرجع‌ها


[1] Aggarwal, C.C., and Reddy, C.K. (2014). *Data clustering. Algorithms and applications.* Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra.

[2] Anand, M.C.J., and Devadoss, A.V. (2013). Using new triangular fuzzy cognitive maps (TrFCM) to analyze causes of divorce in family. *International Journal of Communications and networking Systems*, **2**, 205-213.

[3] Binkowski, M., Marti, G., and Donnat, P. (2018). Autoregressive convolutional neural networks for asynchronous time series. In *International Conference on Machine Learning* (pp. 580-589). PMLR.

[4] Breiman, L. (2001). Random forests. *Machine learning*, **45**, 5-32.

[5] Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quant. Finance*, **1**, 223–236.

[6] Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, **7** (1): 1–26. URL http://www. jstor. org/stable/2958830.

[7] Gao, P., Zhang, R., and Yang, X. (2020). The application of stock index price prediction with neural network. *Mathematical and Computational Applications*, **25**(3), 53.(

[8] Guo, L., Chehata, N., Mallet, C., and Boukir, S. (2011). Relevance of airborne lidar and multispectral image data for urban scene classification using Random Forests. *ISPRS Journal of Photogrammetry and Remote Sensing*, **66**(1), 56-66.

[9] Huang, K., Hussain, A., Wang, Q., Zhang, R. (2019). *Deep Learning: Fundamentals, Theory and Applications*. Springer: Berlin, Germany.

[10] Laloux, L., Cizeau, P., Potters, M., Bouchaud, J.P. (2000). Random matrix theory and financial correlations. *Int. J.Theor. Appl. Finance*, **3**, 391–397.

[11] Lunga, D., and Marwala, T. (2006). Online forecasting of stock market movement direction using the improved incremental algorithm.

In *International Conference on Neural Information Processing* (pp. 440-449). Springer, Berlin, Heidelberg.

[12] Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta numerica*, **8**, 143-195.

[13] Ramnath, S., Rock, S., and Shane, P. (2008). The financial analyst forecasting literature: A taxonomy with suggestions for further research. *International Journal of Forecasting*, **24**(1), 34-75.

# پیوست

برنامه مدل شبکه عصبی

```
> set.seed(201)
> nn <- neuralnet(endprize ~  date + vollum + prize + yprize + efficiency +
transactions + number + total + mm + year + yday + quarter + month + day
,train,linear.output = FALSE, hidden = c(3, 2))

nn3<- neuralnet(endprize ~  date + vollum + prize + yprize + efficiency +
transactions  + number + total + mm  + yday + quarter ,train,linear.output =
FALSE, hidden = c(3, 2))
```

خروجی مدل شبکه عصبی

```
Output:
> nn$act.fct
function (x)
{
   1/(1 + exp(-x))
}
attr(,"type")
[1] "logistic"
```

```
print(nn3)
Output:
>  nn3$act.fct
function (x)
{
   1/(1 + exp(-x))
}
attr(,"type")
[1] "logistic"
```

نتایج معیار ارزیابی مدل شبکه عصبی

```
> predictions = predict(nn, newdata = train)
> mape(train$endprize, predictions)
[1] 99.94605
> predictions = predict(nn, newdata = test)
> mape(test$endprize, predictions)
[1] 99.94635

> predictions3 = predict(nn3, newdata = train)
> mape(train$endprize, predictions3)
[1] ۱.۰3541
 > predictions3 = predict(nn3, newdata = test)
> mape(test$endprize, predictions3)
[1] ۱.۰4032
```

برنامه جنگل تصادفی

```
> set.seed(101)
> rf = randomForest(response ~ data+Yield.with.Excel.+ macd +rsi+ sto + wpr
+ obv + year + yday+ quarter + month + day + weekdays + weekend + week
, data = train)
> set.seed(100)
> rf_revised = randomForest(response ~ Yield.with.Excel.+ macd + month ,
data = train)
```

خروجی مدل جنگل تصادفی

```
> set.seed(101)
> rf = randomForest(response ~ data+Yield.with.Excel.+ macd +rsi+ sto + wpr
+ obv + year + yday+ quarter + month + day + weekdays + weekend + week
, data = train)
> print(rf)
Output:
Call:
 randomForest(formula = response ~ data+Yield.with.Excel.+  macd + rsi + sto
+ wpr + obv + year + yday + quarter + month + day + weekdays + weekend
+      week, data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 5

Output:
Call:
 randomForest(formula = response ~ data+Yield.with.Excel.+  macd + rsi + sto
+ wpr + obv + year + yday + quarter + month + day + weekdays + weekend
+ week, data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 5
Output:
Call:
 randomForest(formula = response ~ Yield.with.Excel. + macd , data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1
```

نتایج معیار ارزیابی مدل جنگل تصادفی

```
> predictions = predict(rf, newdata = train)
> mape(train$response, predictions)
[1] 0.8548224
> predictions = predict(rf, newdata = test)
> mape(test$response, predictions)
[1] 1.893867
```

```
> predictions = predict(rf_revised, newdata = train)
> mape(train$response, predictions)
[1] 1.302628
> predictions = predict(rf_revised, newdata = test)
> mape(test$response, predictions)
[1] 1.753767
```

**مریم محمدی**
دانشجوی دکترای آمار
کرمانشاه، دانشگاه رازی کرمانشاه، گروه آمار.
رایانشانی: mmaryammmohammaddi1234@gmail.com

**حبیب جعفری**
دکترای آمار
کرمانشاه، دانشگاه رازی کرمانشاه، گروه آمار.
رایانشانی: jafari_habib@yahoo.com

**آزاد خانزادی**
دکترای آمار
کرمانشاه، دانشگاه رازی کرمانشاه، گروه آمار.
رایانشانی: a.khanzadi@razi.ac.ir