


Vol. 14/ No. 54/Winter 2025

Research Article

# Spot Price Prediction of Resources in Cloud Computing by Proposing a New Structure in Deep Learning Method Considering the Level of QOS

Seyed Soroush Nezamdoust, Ph.D. Student<sup>1</sup>  | Mohammad Ali Pourmina, Associate Professor<sup>2</sup>  | Farbod Razzazi, Associate Professor<sup>3</sup> 

<sup>1</sup>Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University (IAU), Tehran, Iran, [soroush.nezamdoust@srbiau.ac.ir](mailto:soroush.nezamdoust@srbiau.ac.ir)

<sup>2</sup>Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University (IAU), Tehran, Iran, [pourmina@srbiau.ac.ir](mailto:pourmina@srbiau.ac.ir)

<sup>3</sup>Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University (IAU), Tehran, Iran, [razzazi@srbiau.ac.ir](mailto:razzazi@srbiau.ac.ir)

#### Correspondence

Mohammad Ali Pourmina, Associate Professor of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University (IAU), Tehran, Iran, [pourmina@srbiau.ac.ir](mailto:pourmina@srbiau.ac.ir)

**Received:** 5 November 2023

**Revised:** 13 January 2024

**Accepted:** 17 February 2024

## Abstract

Cloud computing is a computing model that uses three instance, on-demand, reserved, and spot, to provide resources to users. The price of spot instances is on average lower than other patterns and fluctuates based on supply and demand. When a user requests a spot instance, they must provide an offer. Only if the price offered by the user is higher than the spot price, the user can use this type of resources. Therefore, predicting the price of spot instances is very important and challenging. Forecasting such dynamic time series that follow the nonlinear model requires intelligent tools such as neural networks to be able to predict the future values with the least error by observing the values of a time series. Therefore, the reliability and as a result the quality of the service is improved. For this purpose, we considered Amazon EC2 as an experimental platform and used the spot price history to predict the future price by building a new model based on deep learning. The obtained results showed that the model presented in the article based on the proposed structure of MGRU(modified GRU) can well predict nonlinear values and perform better than other methods used in this field.

**Keywords:** Spot price prediction, Cloud computing, Deep neural network, Modified GRU(MGRU).

## Highlights

- Examining deep learning structures for predicting time series.
- Providing an efficient and powerful algorithm to analyze the historical developments of Amazon EC2 spot prices and predict the future price of resources.
- Presenting a proposed architecture based on modified GRU (MGRU).
- Forecasting price trends in the future with the aim of improving the quality of services.
- Accurate prediction of real-world time series with highly volatile data.

**Citation:** SS. Nezamdoust, MA. Pourmina, and F. Razzazi, "Spot Price Prediction of Resources in Cloud Computing by Proposing a New Structure in Deep Learning Method Considering the Level of QOS," *Journal of Southern Communication Engineering*, vol. 14, no. 54, pp. 1–16, 2025, doi:10.30495/jce.2025.1993480.1327, [in Persian].

## مقاله پژوهشی

## پیش‌بینی قیمت منابع در شبکه ابری با پیشنهاد ساختار جدیدی در یادگیری عمیق با در نظر گرفتن سطح کیفیت خدمات

سیدسروش نظام دوست<sup>۱</sup> | محمدعلی پورمینا\*<sup>۲</sup> | فرید رزازی<sup>۳</sup>

## چکیده:

رایانش ابری یک مدل محاسباتی است که برای ارائه منابع به کاربران از سه الگوی برحسب تقاضا، رزرو شده و نقطه‌ای بهره می‌برد. قیمت نمونه‌های نقطه‌ای به طور متوسط کمتر از سایر الگوها بوده و بر اساس عرضه و تقاضا دارای نوسان است. هنگامی که کاربر یک نمونه نقطه‌ای را درخواست کند، باید پیشنهادی ارائه دهد. تنها در صورتی که قیمت پیشنهادی کاربر بالاتر از قیمت نقطه‌ای باشد، کاربر می‌تواند از این نوع منابع استفاده کند. لذا پیش‌بینی قیمت نمونه‌های نقطه‌ای بسیار مهم و چالش‌برانگیز است. پیش‌بینی این‌گونه سری‌های زمانی پویا که از مدل غیرخطی پیروی می‌کنند، نیازمند ابزار هوشمندی مانند شبکه‌های عصبی است تا بتواند با مشاهده مقادیری از یک سری زمانی، مقادیر آتی را با کم‌ترین خطا پیش‌بینی کند. بنابراین قابلیت اطمینان و در نتیجه کیفیت سرویس ارتقاء می‌یابد. بدین منظور، ما آمازون EC2 را به عنوان یک بستر آزمایشی در نظر گرفتیم و از تاریخچه قیمت نقطه‌ای برای پیش‌بینی قیمت آینده با ساخت مدلی نوین مبتنی بر یادگیری عمیق استفاده کردیم. نتایج به دست آمده نشان داد که مدل ارائه شده مقاله بر پایه ساختار پیشنهادی MGRU (GRU اصلاح شده) به خوبی می‌تواند پیش‌بینی مقادیر غیرخطی را انجام دهد و عملکرد بهتری نسبت به سایر روش‌های مورد استفاده در این حوزه داشته باشد.

**کلیدواژه‌ها:** پیش‌بینی قیمت نقطه‌ای، رایانش ابری، شبکه عصبی عمیق، GRU اصلاح شده (MGRU)

<sup>۱</sup> دانشجوی دکتری مهندسی برق، دانشکده برق و کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، soroush.nezamdoust@srbiau.ac.ir

<sup>۲</sup> دانشیار گروه برق، دانشکده برق و کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، pourmina@srbiau.ac.ir

<sup>۳</sup> دانشیار گروه برق، دانشکده برق و کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، razzazi@srbiau.ac.ir

نویسنده مسئول

<sup>۴</sup> محمدعلی پورمینا، دانشیار گروه برق، دانشکده برق و کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، pourmina@srbiau.ac.ir

تاریخ دریافت: ۱۴ آبان ۱۴۰۲

تاریخ بازنگری: ۲۳ دی ۱۴۰۲

تاریخ پذیرش: ۲۸ بهمن ۱۴۰۲

<https://doi.org/10.30495/jce.2025.1993480.1327>

## ۱-مقدمه

رایانش ابری<sup>۱</sup> ساختاری است که دسترسی آسان به منابع را بر اساس تقاضای کاربر از طریق بستر شبکه بدون نیاز به سرمایه اولیه میسر می‌کند. ابرها دارای محیطی مقیاس‌پذیر و انعطاف‌پذیر بوده و الگوهای متنوع و به‌صرفه پرداخت دارند. لذا در سال‌های اخیر رشد چشمگیری با توجه به مزیت‌های خود داشته و به الگوی مناسبی جهت اجرای برنامه‌هایی با قابلیت اطمینان و امنیت بالا تبدیل شده است. امروزه رایانش ابری توانسته است برای دسترسی به زیرساخت‌ها، بسترهای نرم‌افزاری و نرم‌افزار به عنوان سرویس به عنوان یک گزینه قابل توجه و مناسب جهت استفاده شرکت‌های تجاری و مباحث تحقیقاتی شناخته گردد [۱].

<sup>۱</sup> Cloud computing

مدل زیرساخت به عنوان سرویس IaaS<sup>۱</sup>، PaaS<sup>۲</sup>، SaaS<sup>۳</sup> امکانات پردازشی و فضای ذخیره سازی را به صورت سرویس های استاندارد در شبکه به کاربر ارائه می دهد. به این ترتیب کاربران به جای خرید سرورها، فضای مرکز داده و تجهیزات شبکه این منابع را به عنوان سرویس زیرساخت از فراهم کننده سرویس ابر دریافت می کنند. وب سرورهای آمازون<sup>۴</sup> (AWS) مجموعه ای از سرویس های ابری است که توسط سایت آمازون ارائه می شود. در این مدل، ارائه دهنده خدمات IaaS به عنوان مالک تجهیزات شناخته می شود و مسئولیت تهیه مکان فیزیکی، راه اندازی و نگهداری تجهیزات را بر عهده دارد [۲].

در مدل بسترکاری به عنوان سرویس PaaS، مسئولیت اجرا و نگهداری سیستم نرم افزاری و زیرساخت منابع رایانشی را فراهم کننده سرویس ابر بر عهده دارد. این سرویس یک لایه نرم افزاری را به صورت بسته ارائه می دهد که می توان از آن برای تولید سرویس های سطح بالاتر استفاده کرد. در این مدل به کاربر فرصت داده می شود تا مستقیماً بر روی ابر به طراحی، توسعه و آزمایش برنامه مورد نظر خود بپردازد. یک نمونه کاربردی در این زمینه، موتور برنامه گوگل است که امکان پیاده سازی و اجرای برنامه های کاربردی توسط زیرساخت ایجاد شده توسط گوگل را فراهم می کند [۲-۳].

نرم افزار به عنوان سرویس SaaS<sup>۵</sup>، به فراهم کردن برنامه های کاربردی بر حسب تقاضا در بستر اینترنت اطلاق می گردد. این برنامه ها نیازی به نصب و راه اندازی بر روی رایانه های مشتری ندارند. SaaS، نرم افزارها را به صورت یک سرویس بر روی اینترنت تحویل داده و بدین وسیله کاربر می تواند به آن متصل شده و از آن بهره ببرد. بدین شکل، سرویس دهی نرم افزار بر روی ابر بسیار آسان بوده و بروز رسانی، نگهداری و پشتیبانی به صورت متمرکز تنها بر روی ابر صورت می گیرد ولیکن امکان تغییر و سفارشی سازی در این برنامه وجود ندارد. از CRM، Salesforce، Youtube، Gmail می توان به عنوان نمونه های کاربردی این شیوه که امروزه استفاده می شود، نام برد [۲-۳].

هم اکنون درآمد خدمات ابری و پیش بینی های مربوط به آن نشان می دهد که IaaS سریع ترین بخش در حال رشد است، مدلی که منابع زیرساختی را بر اساس ماشین های مجازی در اختیار کاربران قرار داده و امکان اجاره ماشین های مجازی با ظرفیت های متفاوت را به صورت کشسان فراهم می کند [۳]. سرویس های محاسباتی با مدل های متنوع قیمت گذاری به کاربران ارائه می شود. مسئله تعیین قیمت سرویس با رویکردی که سرویس دهنده به بیشترین منفعت ممکن دست پیدا کند و از طرف دیگر موجب حفظ کیفیت سرویس و رضایت کاربران شود، یکی از مهم ترین چالش های اساسی برای ارائه دهندگان سرویس های ابری است. در مدل قیمت گذاری ایستا، منابع مورد تقاضا با دریافت مبلغی مشخص در مدت زمان توافق شده ارائه می شود که می توان به مکانیسم قیمت گذاری پرداخت به میزان استفاده<sup>۶</sup> (OD) اشاره کرد. در الگوی قیمت گذاری مورد تقاضا، کاربران نرخ ثابتی با توجه به نوع منطقه<sup>۷</sup> و ناحیه در دسترس<sup>۸</sup> پرداخت می کنند که هزینه آن ها نسبت به تعداد هسته های پردازنده مورد نیاز، سرعت اجرای پردازنده، اندازه حافظه و سایر پارامترهای مؤثر متفاوت است. ماشین های مجازی قیمت ثابت از ویژگی قابلیت اطمینان<sup>۹</sup> و دسترسی پذیری بالا برخوردار هستند اما قیمت بالاتری نسبت به سایر الگوها دارند [۴].

هم چنین ارائه یک مدل محاسباتی بر اساس میزان عرضه و تقاضا و شرایط حاکم در محیط، مفهوم جدیدی را با نام قیمت گذاری پویا مطرح نموده است. قیمت گذاری نقطه ای<sup>۹</sup> بر پایه مدل قیمت گذاری پویا است و با هدف ایجاد انگیزه برای کاربران به منظور استفاده از منابع اضافی پیشنهاد شده است. سرویس دهنده (مانند آمازون) منابع مازاد خود را که اجاره داده نشده، مجدداً قیمت گذاری می کند و از طریق اجاره منابع با ارائه بیشترین تخفیف ممکن (قیمت پایه) کسب سود کرده و از هزینه های اضافی نگهداری منابع جلوگیری می نماید.

فرایند قیمت گذاری این منابع هر یک ساعت یک بار صورت می گیرد و کاربران متقاضی بایستی برای منبع مورد نظر خود پیشنهاد قیمت ارائه کنند، در صورتی که پیشنهاد ارائه شده از قیمت پایه بیشتر باشد منبع برای مدت یک ساعت در اختیار وی قرار داده

<sup>1</sup> Infrastructure as a service

<sup>2</sup> Platform as a service

<sup>3</sup> Software as a service

<sup>4</sup> Amazon Web Services

<sup>5</sup> On-demand instance

<sup>6</sup> Region

<sup>7</sup> Availability zone

<sup>8</sup> Reliability

<sup>9</sup> Spot pricing

می‌شود. در پایان یک ساعت مجدداً قیمت‌گذاری جدید انجام شده و در صورتی که پیشنهاد قیمت قبلی کاربر از قیمت جدید بیشتر باشد اجاره منبع برای یک ساعت دیگر ادامه می‌یابد تا زمانی که قیمت پیشنهادی کاربر به زیر قیمت پایه برود. در صورتی که قیمت کاربر کم‌تر از قیمت پایه جدید شود، منابع از کاربر پس گرفته می‌شوند. این وضعیت با نام شکست خارج از قیمت پیشنهادی شناخته می‌شود و آمازون، هشدار را دو دقیقه قبل از این اتفاق فعال می‌کند. این قطع می‌تواند منجر به از دست رفتن داده‌ها و غیر قابل اعتماد کردن این روش شود و چالش‌هایی را در پی دارد [۴-۵]. از همین رو در پیشنهادات مختلف در سال‌های گذشته به موارد بیشتری در رابطه با نحوه‌ی در اختیار گرفتن این نمونه‌ها و راهکارهای ارتقاء عملکرد آن‌ها پرداخته شده است.

یکی از راهکارهای مورد استفاده در سال‌های اخیر پیش‌بینی دقیق قیمت نقطه‌ای بوده است. به طور کلی پیش‌بینی دقیق این الگوی قیمت‌گذاری، منجر به کاهش ریسک، افزایش قابلیت اطمینان، افزایش دسترسی‌پذیری و در نتیجه کیفیت سرویس الگوی نقطه‌ای می‌شود. تنظیم زمان خرید بهینه با آگاهی از روند آتی قیمت‌گذاری الگوی نقطه‌ای، مزیت دیگر پیش‌بینی دقیق قیمت است. در نتیجه تمایل کاربران به استفاده از این الگو افزایش یافته و می‌تواند از هزینه‌های پایین‌تر آن در کنار حفظ کیفیت سرویس خود بهره ببرند.

از طرف دیگر، موضوع پیش‌بینی در پژوهش‌های زیادی بررسی شده است. در این میان تکنیک‌های پیش‌بینی سری زمانی در دسته‌های مختلفی قرار می‌گیرند. مدل‌های آماری سنتی مانند مدل میانگین متحرک یکپارچه خود بازگشتی<sup>۱</sup> (ARIMA) به دلیل آنکه شامل اجزای خطی هستند در دنیای واقعی استفاده کمی دارند. از طرف دیگر، برای پیش‌بینی سری‌های زمانی با الگوهای غیرخطی چندین روش آماری غیرخطی مانند مدل خود رگرسیون واریانس ناهمسان شرطی<sup>۲</sup> (ARCH) و مدل خود رگرسیونی واریانس ناهمسان شرطی تعمیم‌یافته<sup>۳</sup> (GARCH) پیشنهاد شده است که معمولاً در مدل‌های غیرخطی خاص مناسب هستند. روند یافتن یک مدل مناسب برای سری‌های زمانی در دنیای واقعی پیچیدگی‌هایی دارد. در سیستم محاسبات ابری نیز واریانس زیادی در دیتاست‌ها وجود دارد لذا برای انجام پیش‌بینی مؤثر بایستی از روش‌هایی استفاده کرد که با داده‌های بسیار متغیر محیط رایانش ابری سازگار باشند [۶-۷].

در سال‌های اخیر برای مسائل پیش‌بینی زمانی، از تکنیک‌های هوش محاسباتی مانند شبکه‌های عصبی مصنوعی<sup>۴</sup> استفاده شده است. شبکه‌های عصبی مصنوعی چشم‌انداز فناوری اطلاعات را ارتقا داده و با توجه به ویژگی‌های تقریب عمومی<sup>۵</sup>، داده محور بودن<sup>۶</sup> و توانایی الگو گرفتن مدل‌های غیرخطی منجر به پیشرفت‌های زیادی در پیش‌بینی سری‌های زمانی شده است. در سال ۱۹۸۰ شبکه عصبی بازگشتی<sup>۷</sup> (RNN) مطرح شدند که توسعه هدفمندی از شبکه‌های عصبی است و با اتصال بین لایه‌ها و اتصال بین نرون‌های داخل لایه، امکان مدل‌سازی رابطه‌های پیچیده موجود در داده‌های سری زمانی را فراهم می‌کند. شبکه‌های عصبی بازگشتی با استفاده از یک حلقه بازگشتی این امکان را به وجود می‌آورد تا اطلاعات به دست آمده توسط شبکه در پایان هر محاسبه برای محاسبات آتی مورد استفاده قرار گیرد. بنابراین می‌توان اطلاعات را در طول مراحل، گره به گره منتقل کرد. تمامی شبکه‌های عصبی بازگشتی به شکل زنجیره‌ای از تکرار ماژول‌های شبکه‌های عصبی هستند. در شبکه‌های عصبی بازگشتی استاندارد این ماژول‌های تکرارشونده ساختار بسیار ساده‌ای همانند یک تابع  $\tanh$  دارند. ساختار RNN ساده در شکل ۱ نشان داده شده است. شبکه‌های عصبی بازگشتی ساده در مواقعی که ما به اطلاعات بیشتر و دورتری نیاز داریم، ناکارآمد است زیرا مکانیزمی جهت مقابله با محوشدگی گرادیان وجود ندارد [۲۱]. در فاز آموزش مجموعه‌های بزرگ داده، مقادیر گرادیان‌ها به تدریج با حرکت به سمت ابتدای شبکه به حدی کوچک می‌شوند که تغییرات وزن به صورت ناچیزی صورت می‌گیرد و به این علت فرایند آموزش بشدت کند می‌شود و در حالات شدیدتر این مسئله باعث متوقف شدن فرایند آموزش می‌گردد.

<sup>1</sup> Autoregressive Integrated Moving Average

<sup>2</sup> Auto Regressive Conditional Heteroscedastic

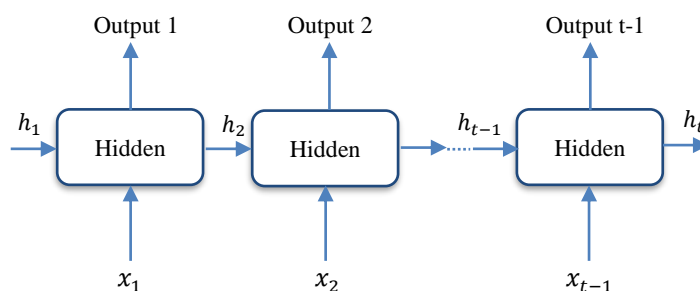
<sup>3</sup> General Autoregressive Conditional Heteroscedastic

<sup>4</sup> Artificial Neural Network

<sup>5</sup> Approximation universal

<sup>6</sup> Being data driven

<sup>7</sup> Recurrent Neural Network

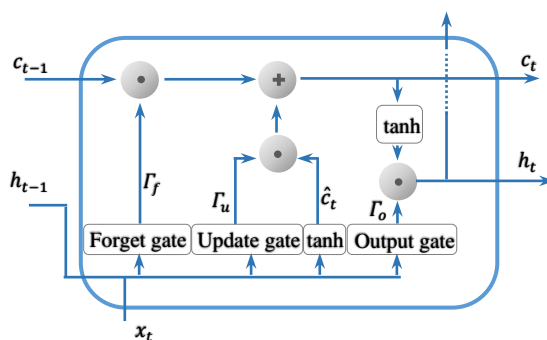


شکل ۱: ساختار باز شده دنباله‌ای شبکه عصبی بازگشتی

Figure 1. RNN unfolded structure

در سال‌های گذشته، پیشرفت‌هایی در مدل‌های برگرفته از شبکه‌های بازگشتی به وجود آمده که عملکرد مناسبی در رفع کاستی‌های کاربرد توالی بلند مدت مدل RNN داشته‌اند. به عنوان مثال هوخرایتر<sup>۱</sup> و اشمیدهور<sup>۲</sup> [۲۲] در سال ۱۹۹۷ میلادی، معماری برای شبکه‌های عصبی بازگشتی پیشنهاد کردند که به شبکه حافظه طولانی کوتاه مدت<sup>۳</sup> (LSTM) معروف شد. این معماری به این علت طراحی شد تا توالی‌های طولانی را به خاطر بسپارد و پدیده محو شدگی گرادیان در شبکه‌های بازگشتی، از طریق تغییر لایه میانی شبکه حل شود.

شبکه‌های LSTM نیز همانند شبکه بازگشتی ساده، حالت تکرار شدنی خود را دارد اما ماژول تکرار شونده آن اندکی تفاوت دارد. همان‌گونه که در شکل (۲) نشان داده شده است.



شکل ۲: ساختار مدل LSTM

Figure 2. Structure of LSTM

در این شبکه اصطلاحاً سه دروازه<sup>۴</sup> وجود دارد که از طریق آن، شبکه نسبت به کنترل جریان داده درون خود اقدام می‌کند. این سه دروازه عبارت‌اند از: دروازه فراموشی<sup>۵</sup>، دروازه به‌روزرسانی<sup>۶</sup> یا دروازه ورودی<sup>۷</sup> و دروازه خروجی<sup>۸</sup>.

علاوه بر سه دروازه مذکور، یک سلول حافظه<sup>۹</sup> نیز در آن وجود دارد که به اختصار C گفته می‌شود. شبکه علاوه بر چهار مفهوم ذکر شده جدید، از یک ورودی از حافظه پنهان h و همچنین ورودی x نیز بهره برده و دو خروجی تولید می‌کند که یک خروجی Ct و خروجی دیگر ht است که خود به دو بخش تقسیم می‌شود بخشی به گام زمانی بعد منتقل شده و بخشی نیز در صورت نیاز به تولید خروجی در گام زمانی فعلی مورد استفاده قرار می‌گیرد. دروازه‌های فراموشی (Gamma\_f) وظیفه کنترل جریان اطلاعات از گام زمانی قبلی داشته و دروازه ورودی یا به‌روزرسانی (Gamma\_u) وظیفه کنترل جریان اطلاعات جدید را بر عهده دارد.

<sup>1</sup> Hochreiter

<sup>2</sup> Schmidhuber

<sup>3</sup> Long Short Term Memory

<sup>4</sup> Gate

<sup>5</sup> Forget gate

<sup>6</sup> Update gate

<sup>7</sup> Input gate

<sup>8</sup> Output gate

<sup>9</sup> Memory Cell

همچنین دروازه خروجی ( $\Gamma_o$ ) مشخص می‌کند چه میزان از اطلاعات گام زمانی قبل با اطلاعات گام زمانی فعلی به گام زمانی بعد منتقل شود. این خروجی با در نظر گرفتن مقدار سلول حالت خواهد بود، ولی از فیلتر مشخصی عبور خواهد کرد [۲۳]. ادامه روند تکامل مدل‌های RNN برای افزایش سرعت محاسبات و بهره‌وری، به ارائه‌ی شبکه عصبی واحد بازگشتی گیتی<sup>۱</sup> (GRU) منجر شد. مدل GRU که در سال ۲۰۱۴ توسط Cho و همکارانش [۲۴] معرفی شد برگرفته از مدل LSTM است. این معماری به منظور برطرف سازی کاستی‌های شبکه عصبی بازگشتی RNN نظیر مشکل محوشدگی گرادیان و همچنین کاهش سربار موجود در معماری LSTM ارائه شده است.

عملکرد بنیادی واحد سلول و گیت‌های GRU مشابه LSTM است. همان گونه که در شکل ۳ نشان دادیم، این نوع معماری که قابلیت یادگیری وابستگی بلندمدت را دارد، از مفاهیمی بنام گیت به‌روزرسانی و گیت بازنشانی استفاده می‌کند. این دو گیت مورد استفاده در اصل دو بردار هستند که با استفاده از آن‌ها تصمیم گرفته می‌شود چه اطلاعاتی به خروجی منتقل شده و چه اطلاعاتی منتقل نشود. نکته خاص درباره این گیت‌ها این است که آن‌ها را می‌توان آموزش داد تا اطلاعات مربوط به گام‌های زمانی بسیار قبل را بدون آن که در گذر زمان دستخوش تغییر شوند، حفظ کرد [۲۵]. فرم ریاضی برای گام زمانی  $t$  با فرض آن که ورودی مجموعه یک مینی بچ<sup>۲</sup>  $X_t$  بوده و  $C_{t-1}$  نشان دهنده مقدار گام زمانی قبلی است، به صورت زیر نمایش داده می‌شود:

$$\Gamma_u = \sigma(W_2 \cdot [C_{t-1}, X_t] + b_u) \quad (1)$$

$$\Gamma_r = \sigma(W_3 \cdot [C_{t-1}, X_t] + b_r) \quad (2)$$

دروازه به‌روزرسانی ( $\Gamma_u$ ) اصطلاحاً سویچی است که مشخص می‌کند در یک گام زمانی، حالت قبلی یا ورودی یا ترکیبی از آن‌ها مورد استفاده قرار گیرد. با استفاده از این قابلیت شبکه قادر خواهد بود تا المان‌هایی را از گذشته دور در حافظه خود نگه داشته و از آن‌ها بهره‌برداری کند. این وظایف در شبکه‌های LSTM توسط دو دروازه مجزای فراموشی و ورودی انجام می‌شود. دروازه ریسیت ( $\Gamma_r$ ) همانند سویچی عمل می‌کند که شبکه با کمک آن مشخص کند چه میزان از اطلاعات گذشته در گام فعلی مورد نیاز نیست و در گام فعلی از چه میزان از اطلاعات گام قبل استفاده شود. به‌طور دقیق‌تر با صفر بودن این سویچ این دروازه در عمل شبکه را وادار می‌کند به گونه‌ای عمل کند که گویا در حال خواندن اولین بخش از دنباله ورودی است و این‌طور شبکه را قادر به فراموشی حالت محاسبه شده قبلی می‌کند.

کاندیدای برای مقدار جاری از گیت ریسیت برای ذخیره اطلاعات قبل استفاده می‌کند. ورودی در وزن ضرب شده و همچنین گیت ریسیت و مقدار گام زمانی قبلی در هم ضرب می‌شود. سپس از طریق تابع فعال‌سازی تانژانت هایپربولیک، مقدار کاندیدای<sup>۳</sup>  $\hat{C}_t$  به دست می‌آید:

$$\hat{C}_t = \tanh(W_1 \cdot [\Gamma_r \odot C_{t-1}, X_t] + b_C) \quad (3)$$

در پایان تأثیر دروازه به‌روزرسانی در نظر گرفته می‌شود که در تعیین مقدار جدید ( $C_t$ ) میزان اطلاعات مورد استفاده هریک از مقادیر کاندیدای چه میزان است.

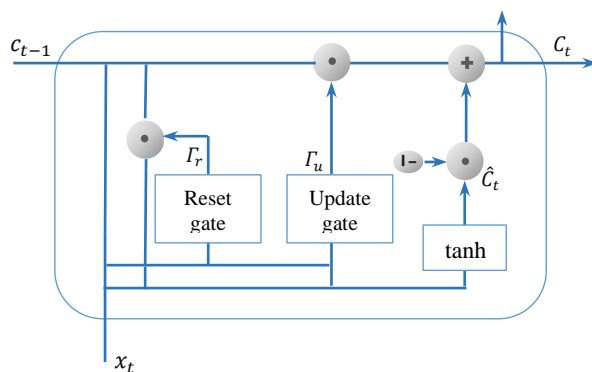
$$C_t = \Gamma_u \odot C_{t-1} + (1 - \Gamma_u) \odot \hat{C}_t \quad (4)$$

معرفی مدل‌های حافظه طولانی کوتاه مدت (LSTM) و شبکه عصبی واحد بازگشتی گیت (GRU) به دلیل ظرفیت وسیع مدیریت اطلاعات آن‌ها، بسیاری از مشکلات شبکه‌های عصبی بازگشتی برطرف کرد [۸-۹]. ولیکن این روش‌ها نیز در پیش‌بینی سری‌های زمانی محدودیت‌هایی دارند که پژوهش‌های مختلفی در این راستا با توجه به اهمیت موضوع انجام شده است.

<sup>1</sup> Gated Recurrent Unit

<sup>2</sup> Mini-batch

<sup>3</sup> Candidate Value



شکل ۳: ساختار کلی شبکه GRU  
Figure 3. Structure of GRU

در این مطالعه، نویسندگان مدل جامعی را برای پیش‌بینی قیمت منابع در شبکه ابری با استفاده از روش یادگیری عمیق با در نظر گرفتن سطح کیفیت خدمات پیشنهاد داده‌اند. پس از انجام اصلاحات لازم جهت افزایش دقت و انعطاف‌پذیر کردن ساختارها، آزمایش‌ها انجام شده و نتایج بر اساس معیارهای مورد استفاده در مقاله با سایر روش‌های مطرح در این حوزه مقایسه می‌گردد. بقیه مقاله به شرح زیر تنظیم شده است. بخش ۲ بررسی مختصری از پیشینه مربوط به این مطالعه را ارائه می‌دهد. معماری و تشریح روش پیشنهادی ما در بخش ۳ نشان داده شده است. ما نتایج ارزیابی عملکرد را در بخش ۴ ارائه و تجزیه و تحلیل می‌کنیم و در نهایت، نتیجه‌گیری در بخش ۵ ارائه شده است.

## ۲- پیشینه تحقیق

همان گونه که اشاره شد، ویژگی متمایزکننده‌ی نمونه‌های نقطه‌ای، قیمت‌گذاری پویای آن‌ها است. قیمت این نمونه‌ها با توجه به عرضه و تقاضای منابع تغییر می‌کند. چالش اصلی مشتریان در استفاده از این منابع، پیش‌بینی قیمت قبل از ارائه‌ی پیشنهاد است. با توجه به این که پیش‌بینی دقیق قیمت نقطه‌ای موجب کاهش هزینه استفاده کاربران از منابع و صرفه‌جویی در زمان محاسباتی وظیفه می‌شود، بسیار مهم است. لذا در سال‌های گذشته تکنیک‌هایی در کارهای پژوهشی مختلف پیشنهاد شده است.

در مقاله [۱۰]، Dutta و Singh مدلی بر پایه روش اتورگرسیو<sup>۱</sup> جهت پیش‌بینی قیمت نقطه‌ای منابع پیشنهاد داده‌اند. در حالی که مدل‌های خطی عملکرد مناسبی در فضای پیچیده ابر ندارند. نویسندگان در [۱۱] از تکنیک‌های میانگین متحرک (مانند ساده، وزنی و نمایی) برای پیش‌بینی قیمت‌های نقطه‌ای در ساعت بعدی خود با استفاده از برآوردها استفاده می‌کنند. هم‌چنین مقاله‌ی [۱۲] با استفاده از مدل رگرسیون جنگل تصادفی به پیش‌بینی قیمت نمونه‌های نقطه‌ای منابع ابری برای یک هفته و یک روز پیش‌رو می‌پردازد. ابتدا یک مجموعه داده با استفاده از دنباله‌ی قیمت نمونه‌ها تشکیل می‌شود که بردار پیش‌بینی حاصل از آن نشان‌دهنده‌ی زمان‌هایی است که قیمت تغییر می‌کند. هم‌چنین Alkharif و همکارانش در [۱۳] یک مدل میانگین متحرک خود همبسته یکپارچه فصلی<sup>۲</sup> (SARIMA) جهت بهبود دقت پیش‌بینی تغییر قیمت نمونه‌های نقطه‌ای پیشنهاد کرده‌اند. این مدل‌ها معمولاً قابلیت سازگاری بهینه‌ای با توجه به وجود نوسان زیاد قیمت‌های نقطه‌ای در پلت فرم ابر ندارند.

در مقاله [۱۴] که در سال ۲۰۲۰ برای به دست آوردن عملکرد بهینه پیش‌بینی قیمت نقطه‌ای منتشر شده، مدل نزدیک‌ترین همسایه K (KNN) برای پیش‌بینی قیمت منابع ابری پیشنهاد شده است. Knn با وجود مزایایی که دارد، معمولاً منجر به هزینه‌های بالای محاسباتی نیز می‌شود.

پژوهش انجام شده Wallace و همکاران در [۱۵] پیش‌بینی قیمت نقطه‌ای را بر اساس پرسپترون<sup>۳</sup> چند لایه استاندارد پیشنهاد داده است در حالی که در این گونه ساختارهای شبکه عصبی نمی‌توان به صورت مؤثر مسئله توالی بلند را پردازش کرد.

<sup>1</sup> Autoregressive

<sup>2</sup> Seasonal Auto-Regressive Integrated Moving Average

<sup>3</sup> Perceptron

به طور کلی در اغلب روش‌های کلاسیک پیشنهادی توسط محققین، نیاز به مجموعه سری‌هایی با ترتیب مشخص برای پیش‌بینی دقیق است. ولیکن در مراکز ابری مشاهده می‌شود که سری‌های بزرگی با نوسان بالا وجود دارد و به همین دلیل اغلب شاهد پیش‌بینی‌های بهینه‌ای از طریق این گونه روش‌ها نیستیم. لذا جهت ارتقاء عملکرد پیش‌بینی قیمت‌های نقطه‌ای که نوسان زیادی دارند نیاز به استراتژی‌های هوشمندانه‌تری است.

شبکه‌های عصبی بازگشتی (RNN) نسخه قدرتمندی از شبکه‌های عصبی مصنوعی با توانایی مدیریت الگوهای زمانی است. به دلیل ساختار داخلی‌شان، ورودی قبلی خود را به خاطر سپرده و از این حافظه برای پردازش دنباله‌ای از ورودی‌ها استفاده می‌کند. RNN در سری‌های با وابستگی کوتاه مدت کارکرد مناسبی دارد ولیکن پژوهشگران در [۱۶] عدم توانایی مناسب آن‌ها را در مسئله وابستگی‌های بلندمدت به اثبات رساندند و نشان دادند در کارهایی که فاصله بین اطلاعات مرتبط با نقطه‌ای که این اطلاعات مورد نیاز است زیاد باشد RNN دچار مشکل محو تدریجی در آموزش شده و هم‌زمان با افزایش داده‌ها عدم توانایی به خاطر سپردن وابستگی طولانی مدت جدی‌تر شده و مانع آموزش بیشتر RNN می‌شود.

بنابراین شبکه‌های LSTM با توانایی یادگیری وابستگی‌های بلندمدت برای توالی‌های سری زمانی برای رفع مشکل RNN پیشنهاد شد. در مقاله [۱۷] از ساختار شبکه عصبی عمیق LSTM در Amazon EC2 جهت پیش‌بینی قیمت نقطه‌ای استفاده کرده‌اند. همچنین مقاله‌ی [۱۸] با در نظر گرفتن قیمت نمونه‌های نقطه‌ای به صورت سری زمانی و با استفاده از یک معماری شبکه‌ی عصبی، قیمت‌های آینده را پیش‌بینی می‌کند. این معماری شبکه‌ی عصبی، متشکل از ۳ لایه است، دو لایه LSTM و یک لایه برای متراکم و یکی کردن خروجی‌های دولایه‌ی قبلی. نتایج مقاله نشان از عملکرد مناسب روش دارد. ساختار GRU مدل دیگری برگرفته از RNN است که علی‌رغم مزایای ساختاری و کارایی محاسباتی آن، کارهای تحقیقاتی کمی جهت پیش‌بینی در فضای ابری با استفاده از آن انجام شده است [۱۹]. Kong و همکارانش در سال ۲۰۲۱ از داده‌های تاریخی قیمت نقطه‌ای آمازون استفاده کرده و با استفاده از مدل GRU به پیش‌بینی قیمت نقطه‌ای منابع ابری و بررسی عملکرد آن‌ها در قیاس با روش‌های دیگر پرداخته‌اند [۲۰].

به طور کلی در کارهای تحقیقاتی محدودی که به بررسی و پیش‌بینی قیمت‌های نقطه‌ای پرداخته شده، به صورت جامع و دقیق چالش‌های این حوزه مورد توجه قرار نگرفته است که ممکن است موجب افزایش بار محاسباتی، صرف زمان بیش از حد و پایین آمدن دقت پیش‌بینی شود. لذا ما با در نظر گرفتن دغدغه‌های مختلف، ساختاری ساختیم تا منجر به بهینه‌سازی دقت پیش‌بینی و بهره‌وری محاسباتی در محیط رایانش ابری شود. دقت مدل ما در مقایسه با چندین مدل دیگر تأیید می‌شود.

### ۳- روش پیشنهادی

در مدل قیمت‌گذاری نقطه‌ای پلتفرم ابر، بهره‌برداری از روشی بهینه جهت پیش‌بینی دقیق و سریع قیمت با هدف افزایش قابلیت اطمینان و حفظ سطح کیفیت خدمات، یک گام اساسی در راستای کمک به کاربر در روند تصمیم‌گیری استفاده از منابع است تا بتوان با کم‌ترین ریسک از مزیت هزینه پایین این الگو بهره برد.

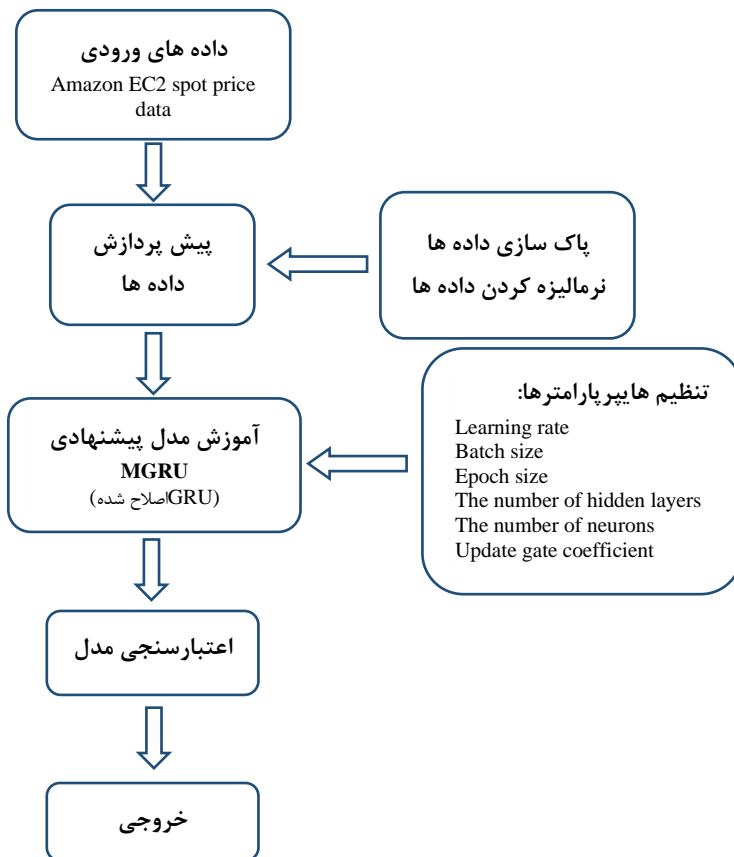
ساختار کلی مدل ارائه شده پیشنهادی که مبتنی بر یادگیری عمیق است در شکل ۴ نشان داده شده است. در فصل قبل مقاله، مدل‌های شبکه‌های عصبی بازگشتی RNN، LSTM و GRU به اختصار بیان شد و در ادامه فصل جاری با استفاده از مفاهیم مذکور به ارائه روش نوینی بر اساس اصلاح مدل GRU می‌پردازیم.

### ۳-۱- تشریح روش پیشنهادی

در شکل چهار به صورت فلوچارت روش پیشنهادی مقاله ارائه شد. اولین مرحله بعد از ورود داده‌های اولیه (مجموعه قیمت‌های نقطه‌ای)، پیش‌پردازش آن‌ها است. داده‌های خام ورودی در اغلب موارد مشکلاتی مانند نویز، بایاس، تغییرات شدید در بازه دینامیکی دارند که استفاده از آن‌ها به همین شکل باعث تضعیف مدل پیشنهادی در مراحل بعد خود می‌شود. در مدل ارائه شده مقاله از اطلاعات تاریخچه قیمت نقطه‌ای مراکز داده ابری آمازون استفاده شده است. ابتدا می‌بایست داده‌های سری زمانی داده شده  $(x_1, x_2, x_3, \dots, x_n)$  به طول  $n$  را پیش‌پردازش کنیم. بدیهی است عملیات پیش‌پردازش بر روی داده‌های قیمت منجر به بهبود کیفی و کمی آن‌ها شده و نتایج بهتری در پی خواهد داشت. در روش پیشنهادی با هدف آماده‌سازی داده‌های ورودی از دو روش



پاک‌سازی<sup>۱</sup> و نرمال‌سازی داده در مرحله پیش‌پردازش بهره گرفته می‌شود. پاک‌سازی داده‌ها می‌تواند برای کنترل نویز در داده‌ها به کار گرفته شود. اگر سری زمانی مورد استفاده حاوی نویز و مقادیر از دست رفته باشد، نویز باید هموار شود و هم چنین مقادیر از دست رفته قبل از نرمال‌سازی داده‌ها، جایگزین شوند. در این مقاله، داده‌های ورودی قیمت‌های منابع ابری که دچار ناسازگاری و مشکل شده و یا ناموجود شده‌اند را شناسایی کرده و آن‌ها را حذف، جایگزین و یا اصلاح می‌کنیم.



شکل ۴: ساختار کلی مدل پیشنهادی

Figure 4. Architecture Overview

هم چنین، از آنجایی که شرط اصلی و لازمه استفاده از داده‌های ورودی در تجزیه و تحلیل‌های آماری و مدل‌سازی، نرمال بودن داده‌ها است، لذا قبل از استفاده داده‌ها باید از نرمال بودن آن‌ها مطمئن شویم. دقت و کارایی الگوریتم‌های استخراج داده، شامل سنجش فاصله توسط نرمال‌سازی بهبود می‌یابد. روش‌های مختلفی در چند سال اخیر پیشنهاد شده است که می‌توان به تبدیل مینیمم-ماکزیمم، نرمال‌سازی Score-Z، نرمال‌سازی رتبه‌ای و ... اشاره کرد. از آنجایی که پیش‌بینی بهینه قیمت با استفاده از شبکه‌های عصبی مصنوعی هدف مقاله است، ورود داده‌ها به صورت استاندارد و انجام پیش‌پردازش اصولی کمک زیادی به کاهش خطا می‌کند.

همان‌طور که در شکل ۴ مشاهده می‌گردد، قبل از مرحله آموزش در مدل پیشنهادی بایستی زیرمجموعه‌ای از مقادیر برای هر هایپر پارامتر تعریف و ترکیبی از بهترین هایپر پارامترها در هر تکرار تخمین زده شود. تنظیم و یافتن مقادیر هایپر پارامترها، یک فرایند قدرتمند برای شناسایی و یافتن بهترین مقادیر ممکن داده‌ها با هدف دستیابی به نتیجه بهینه مدل‌سازی است. با

<sup>1</sup> Data cleaning

استفاده از روش گریدسرچ<sup>۱</sup>، ما شش پارامتر شامل تعداد نمونه‌ها در هر دسته<sup>۲</sup>، تعداد لایه‌های پنهان<sup>۳</sup>، تعداد نرون‌ها در هر لایه<sup>۴</sup>، نرخ یادگیری<sup>۵</sup>، تعداد دوره<sup>۶</sup> و ضریب گیت به‌روزرسانی<sup>۷</sup> مدل پیشنهادی را قبل آموزش تعیین می‌کنیم. در مرحله بعد به آموزش ساختار پیشنهادی مقاله می‌پردازیم. با توجه به ویژگی‌ها و قابلیت‌های مدل GRU که در رابطه با آن صحبت شد، در این مقاله با ارائه تغییرات ساختاری و اصلاحات لازم در آن، ساختار پیشنهادی MGRU (GRU اصلاح شده) پیشنهاد گردید. در کنار مزایایی که مدل GRU دارد، نقص‌هایی نیز در آن مشاهده می‌شود که رفع آن‌ها می‌تواند قابلیت این روش را در بسیاری از کاربردهای موجود، افزایش دهد.

همان‌طور که قبلاً گفته شد، در شبکه‌های LSTM گیت فراموشی و گیت به‌روزرسانی به صورت مستقل وظیفه فراموشی و به‌روزرسانی جریان اطلاعات را برعهده دارند و مکانیزم کنترل میزان نرخ کارکرد هر گیت به صورت دلخواه تنظیم می‌شود. این در حالی است که GRU هر دو عملیات را از طریق گیت بروز رسانی انجام می‌دهد و اطلاعات مقدار جدید از طریق بده و بستن بین مقدار گام زمانی قبلی و  $\hat{C}_t$  کنترل می‌شود.

با توجه به روابط ۱، ۲، ۳ و ۴ اگر غال با دروازه گیت‌های ریسیت فعال باشند، واحد مستعد یادگیری وابستگی‌های کوتاه مدت است. لذا اگر  $\Gamma_u$  به سمت صفر برود شاهد افزایش تأثیر  $\hat{C}_t$  هستیم و عبور زود هنگام اطلاعات وجود دارد. از طرف دیگر اگر غالباً گیت‌های آپدیت فعال باشند شاهد وجود وابستگی بلند مدت هستیم. پس اگر  $\Gamma_u$  به سمت یک میل کند شبکه به حفظ  $C_{t-1}$  تمایل بیشتری داشته و از عبور زود هنگام اطلاعات جلوگیری می‌کند.

$C_t$  تعامل خطی بین مقدار گام زمانی قبلی و مقدار کاندیدای است که می‌توان در رابطه ۴ جهت سادگی تحلیل،  $\Gamma_u$  و  $\Gamma_u^{-1}$  را به ترتیب با  $\alpha_1$  و  $\alpha_2$  جایگزین کرد. بدیهی است که مجموع این دو مؤلفه برابر با یک بوده و نسبت به یکدیگر تأثیر عکس دارند بدین صورت که اگر  $\alpha_1$  با یک نرخ ثابت به سمت صفر برود،  $\alpha_2$  با همان نرخ در جهت خلاف یعنی به سمت مقدار یک می‌رود که این موضوع در رابطه با یادگیری و گردش اطلاعات مشکلاتی را به وجود می‌آورد [۲۶]. هنگام آموزش مدل، ارتباط متقابل پارامترهای  $\alpha_1$  و  $\alpha_2$  ممکن است مانع گردش مناسب اطلاعات شود و در نتیجه شاهد کاهش سرعت یادگیری باشیم.

$$C_t = \alpha_1 \odot C_{t-1} + \alpha_2 \odot \hat{C}_t \quad (5)$$

لذا با هدف تغییر محدودیت خطی و افزایش سرعت یادگیری در ساختار ارائه شده در مقاله دو پیشنهاد ارائه گردید. پیشنهاد اول بدین شکل بیان می‌گردد. در ابتدای کار، جایگزین رابطه  $\alpha_1 + \alpha_2 = 1$ ، رابطه زیر پیشنهاد می‌گردد:

$$\sqrt{\alpha_1^2 + \alpha_2^2} = 1 \quad (6)$$

سپس در پیشنهاد دوم، فاکتوری به نام  $\gamma$  به عنوان ضریب گیت بروز رسانی تعریف می‌کنیم که تنظیم‌کننده حافظه بوده و مقدار آن بین ۰ تا ۱ با توجه به نوع ساختار و شرایط مورد استفاده متغیر است.

$$\alpha_1 = \gamma \Gamma_u \quad (7)$$

با توجه به رابطه ۶ و  $\gamma$  مقدار  $\alpha_2$  به شکل زیر تعریف می‌شود:

$$\alpha_2 = \sqrt{1 - \gamma^2 \Gamma_u^2} \quad (8)$$

سپس با جایگزینی روابط ۷ و ۸ در رابطه ۵ داریم:

$$C_t = \gamma \Gamma_u \odot C_{t-1} + \sqrt{1 - \gamma^2 \Gamma_u^2} \odot \hat{C}_t \quad \gamma \in (0, 1) \quad (9)$$

همان‌طور که در رابطه‌ی ۹ مشاهده می‌گردد اندازه‌های پارامترهای متغیر ضریب گیت به‌روزرسانی ( $\gamma$ ) پیشنهادی می‌توانند وابستگی‌های کوتاه مدت یا بلند مدت را ایجاد کند. به طور کلی با اعمال تغییرات در ساختار پیشنهادی MGRU می‌توان با حفظ مزایای قابل توجه GRU، محدودیت‌های آن را تا حد زیادی رفع کرد. در ساختار GRU یک محدودیت خطی بین حالت

<sup>1</sup> Grid Search

<sup>2</sup> Batch size

<sup>3</sup> The number of hidden layers

<sup>4</sup> The number of neurons in each layer

<sup>5</sup> Learning rate

<sup>6</sup> Epoch size

<sup>7</sup> Update gate coefficient

قبل و جاری وجود دارد که این معضل موجب عدم بهینگی گردش اطلاعات می‌گردد. با انجام تغییراتی در ساختار پایه‌ای و ارائه مدل پیشنهادی MGRU، گردش اطلاعات افزایش داشته و در نتیجه سرعت یادگیری بالاتر می‌رود. همان‌گونه که در رابطه ۹ ساختار اصلاح شده GRU مشاهده شد، هایپرپارامتر پیشنهادی ضریب گیت بروزرسانی ( $\gamma$ ) موجب اصلاح محدودیت خطی با تنظیم حافظه می‌گردد. این قابلیت موجب می‌شود تا مقدار بهینه ضریب پیشنهادی متناسب با ساختارهای متفاوت، داده‌های متنوع و ... تعیین شود که در نتیجه مزیت قابل توجه انعطاف‌پذیری و بهینگی مدل ارائه شده در هر شرایطی ایجاد می‌گردد.

الگوریتم ۱: GRU اصلاح شده (MGRU)

Algorithm 1. Modified Gated Recurrent Unit (MGRU)

---

**Require:**  $input, Cell_{last}, W_1, W_2, W_3, b_u, b_r, b_c, \gamma$   
**Ensure:**  $Cell_{value}$   
**while** When having input **do**  
      $temp_{update} = (W_2 \cdot [Cell_{last}, input] + b_u)$   
      $temp_{reset} = (W_3 \cdot [Cell_{last}, input] + b_r)$   
      $update_{gate} = \text{sigmoid}(temp_{update})$   
      $reset_{gate} = \text{sigmoid}(temp_{reset})$   
      $\alpha_1 = \gamma(update_{gate})$   
      $\alpha_2 = \sqrt{1 - \gamma^2(update_{gate})^2}$   
      $temp_{cell} = (W_1 \cdot [reset_{gate} \odot Cell_{last}, input] + b_c)$   
      $Cell_{value} = \text{activate}(temp_{cell}) \odot \alpha_1 + C_{t-1} \odot \alpha_2$   
**end while**

---

#### ۴- نتایج آزمایش

آمازون وب سرویس (AWS) از چندین منطقه و ناحیه در دسترس برای بهبود خطا، کاهش تأخیر و مدیریت ترافیک تشکیل شده است. کاربران می‌توانند مکان منابع خود را در الگوهای متنوع EC2 مشخص کنند که هر الگو شامل ویژگی‌های مختلفی در CPU، RAM، Storage و غیره است.

برای اعتبارسنجی مدل پیشنهادی، ما از داده‌های واقعی قیمت‌های نقطه‌ای آمازون EC2 در انواع ماشین‌های مجازی C3.2xlarge، M3.2xlarge و M3.medium در دیتاهای سری‌های زمانی متعلق به دوره بین ۷ مارس الی ۷ ژوئن ۲۰۱۶ US-west region (Oregon) استفاده کردیم [۲۷]. با توجه به این که فاصله زمانی به‌روزرسانی قیمت نقطه‌ای متنوع و نامشخص است، تاریخچه قیمت را به‌صورت بازه یک ساعته بازنمونه‌گیری<sup>۱</sup> می‌کنیم و حداکثر قیمت هر بازه یک ساعته را به عنوان مقدار آن نمونه تعیین می‌کنیم.

همچنین لازم به ذکر است که برای به دست آوردن یک نتیجه دقیق‌تر در مدل پیشنهادی، از روش نمودار جعبه<sup>۲</sup> برای از بین بردن داده‌های پرت و دورافتاده<sup>۳</sup> استفاده می‌شود و سپس بر اساس درون‌یابی خطی<sup>۴</sup> جایگزینی مقادیر از دست رفته انجام می‌گردد.

با توجه به این که در بازه‌های زمانی مختلف تفاوت زیادی در محدوده‌ی مقدار داده‌های نقطه‌ای وجود دارد، لذا داده‌های اصلی باید قبل از رفتن به مرحله بعدی نرمالیزه شوند که باعث افزایش سرعت همگرایی می‌شود. روش نرمال‌سازی در این مقاله از رابطه زیر پیروی می‌کند:

$$x' = \frac{x - \text{mean}(\bar{X})}{\sigma} \quad (10)$$

<sup>1</sup> Resample

<sup>2</sup> Box plot

<sup>3</sup> Outlier

<sup>4</sup> Linear interpolation

$$\sigma = \sqrt{E(\bar{X}^2) - E(\bar{X})^2} \quad (11)$$

در این روابط  $\sigma$  انحراف معیار است.

در مدل پیش‌بینی مقاله، داده‌ها به صورت دسته‌بندی وارد می‌شوند. با جزئیات بیشتر، ما مجموعه داده‌های خود را به صورت تصادفی به این صورت تقسیم می‌کنیم: مجموعه آموزش که برای محاسبه وزن و در نتیجه آموزش مدل بکار می‌رود ۶۰ درصد داده‌ها را به خود اختصاص می‌دهد. مجموعه اعتبارسنجی که برای انتخاب ابرپارامترها و سایر وظایف انتخاب مدل بکار می‌رود ۲۰ درصد و در پایان پس از گزینش مدل بهینه، مجموعه تست که وظیفه ارزیابی عملکرد مدل انتخابی را بر عهده دارد ۲۰ درصد از مجموعه داده‌ها را به خود اختصاص می‌دهد. به طور کلی ۸۰ درصد داده‌ها جهت آموزش و ۲۰ درصد داده‌ها در مجموعه تست بکار می‌روند.

در ساختار پیشنهادی، انتخاب مقادیر هایپرپارامترهای مختلف برای حل مسائل overfitting و underfitting بسیار قابل اهمیت است. در این مقاله از روش جستجوی شبکه‌ای برای ارزیابی تمام ترکیبات هایپرپارامترها استفاده شد. ابتدا زیرمجموعه‌ای از هایپرپارامترهای مورد استفاده تعریف و سپس بهترین ترکیب برای فرایند یادگیری انتخاب و اعمال گردید. لذا در روش ارائه شده مقاله پس از تعیین زیر مجموعه برای هر یک از هایپرپارامترهای مدل پیشنهادی شامل بیج سایز، تعداد لایه‌های پنهان، تعداد نرون‌ها در هر لایه، نرخ یادگیری و تعداد دوره‌ها، به ترتیب مقادیر ۵، ۳، ۶۴، ۰/۰۰۱ و ۱۰۰ از طریق روش گرید سرچ ارزیابی و انتخاب گردید. هم چنین در هایپرپارامتر ضریب گیت بروزرسانی مدل پیشنهادی MGRU، ابتدا پس از انتخاب زیر مجموعه‌ی صفر تا یک با فاصله ۰/۱ می‌باشد (۰/۹، ۰/۸، ...، ۰/۲، ۰/۱) که به مقدار بهینه ۰/۸ در مجموعه هایپرپارامترهای ساختار مقاله رسیدیم.

ما در این مقاله از روش‌های هوش محاسباتی RNN، LSTM، GRU، Transformer و همچنین روش آماری ARIMA برای ارزیابی عملکرد روش پیشنهادی استفاده کردیم. صحت پیش‌بینی برای ارزیابی همه مدل‌ها با استفاده از معیارهای عملکرد مختلف به نام‌های مجذور میانگین مربعات خطا<sup>۱</sup> (RMSE)، میانگین خطای مطلق<sup>۲</sup> (MAE) و ضریب  $R^2$ <sup>۳</sup> به دست می‌آید. رابطه مجذور میانگین مربعات خطا (RMSE) برابر است با:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n}} \quad (12)$$

همچنین میانگین خطای مطلق (MAE) به شکل زیر بیان می‌شود:

$$MAE = \frac{\sum |y_i - \hat{y}_i|}{n} \quad (13)$$

و در آخر ضریب  $R^2$  با رابطه زیر محاسبه می‌شود:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (\hat{y}_i - \bar{y}_i)^2} \quad (14)$$

در روابط فوق  $y_i$ ،  $\hat{y}_i$  و  $\bar{y}_i$  به ترتیب مقادیر واقعی، پیش‌بینی شده و میانگین را نشان می‌دهد. هم چنین تعداد نمونه داده‌های ارزیابی شده با  $n$  نشان داده شده است.

#### ۴-۱- تجزیه و تحلیل عملکرد

در این بخش، یافته‌های خود را نشان داده و عملکرد چارچوب پیشنهادی خود را ارزیابی می‌کنیم.

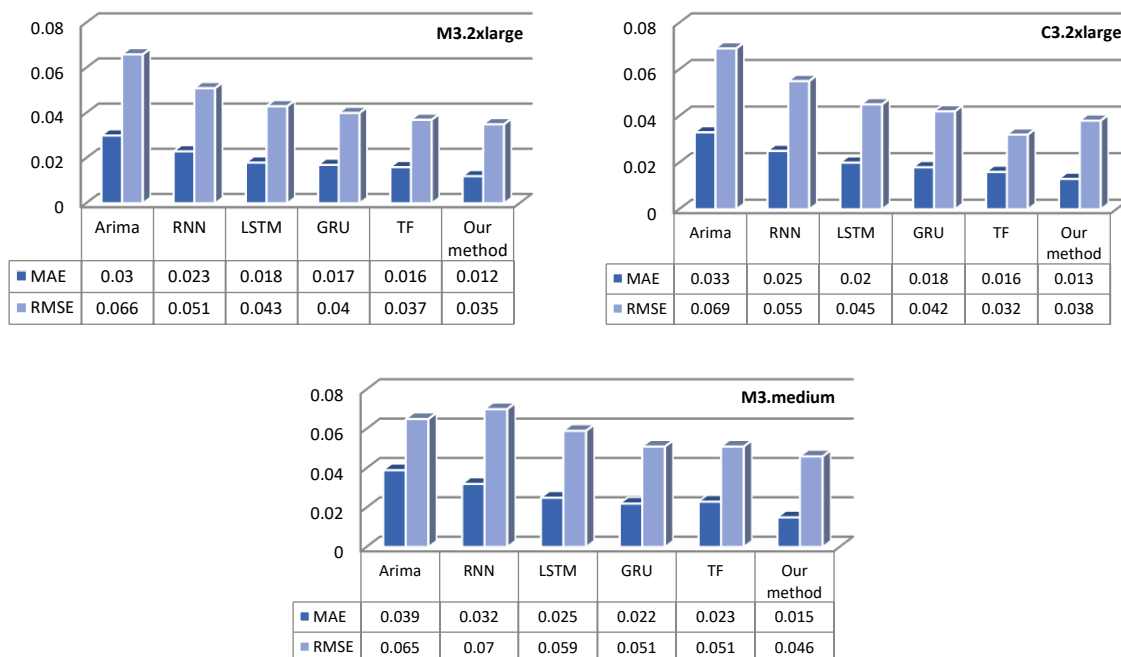
ابتدا برای بررسی قدرت پیش‌بینی مدل پیشنهادی، آن را با پنج روش پیشرفته LSTM، RNN، Transformer، GRU، ARIMA مقایسه می‌کنیم. لازم به ذکر است که جهت مقایسه دقیق و عادلانه نتایج بین روش‌های مورد بررسی، پارامترهای بهینه‌ای که

<sup>1</sup> Root Mean Square Error

<sup>2</sup> Mean Absolute Error

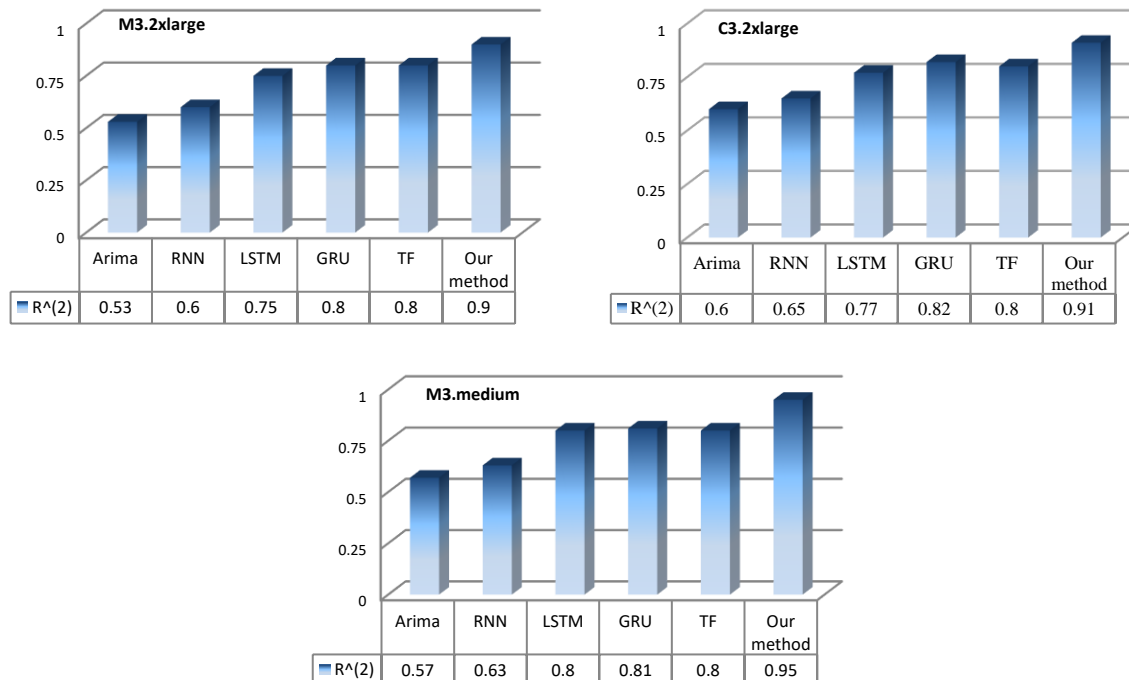
<sup>3</sup> The Coefficient of Determination

بهترین عملکرد را در آن روش بر روی مقادیر دارد جستجو و تعیین کردیم. عملکرد سه ماشین مجازی مشخص شده بررسی و در شکل های ۵ و ۶ عملکرد روش های مربوطه مشخص شده است.



شکل ۵: مقایسه عملکرد پیش بینی (RMSE, MAE) قیمت نقطه ای شش روش مختلف روی سه ماشین مجازی

Figure 5. Comparison of predicting performance (RMSE, MAE) of the Spot price of six different methods on three virtual machines



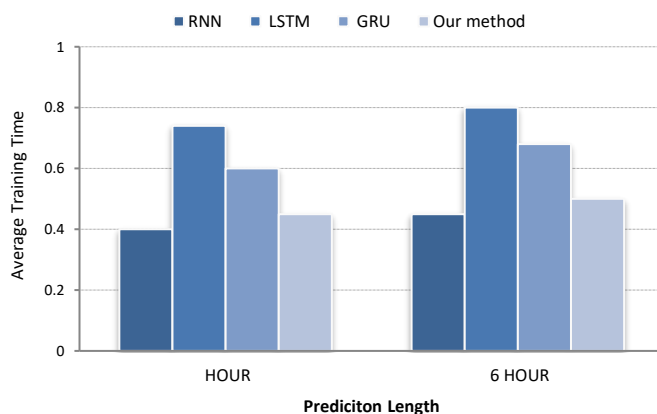
شکل ۶: مقایسه عملکرد پیش بینی R<sup>2</sup> قیمت نقطه ای شش روش مختلف روی سه ماشین مجازی

Figure 6. Comparison of predicting performance (R-Square) of the Spot price of six different methods on three virtual machine

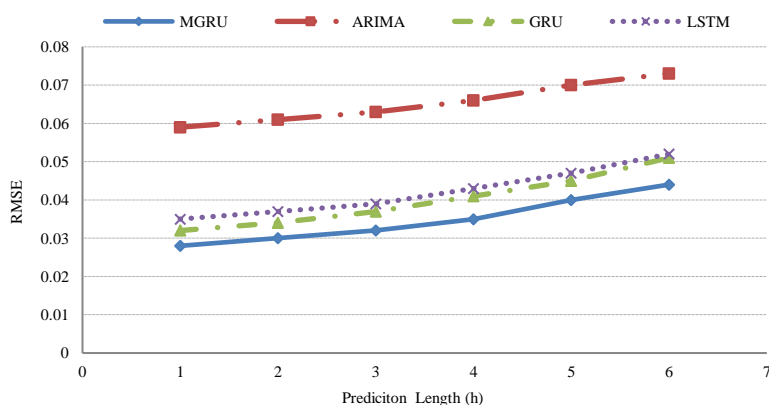
نتایج نشان می دهد که روش پیشنهادی مقاله کمترین خطا (RMSE and MAE) و بالاترین دقت (R<sup>2</sup>) را دارد. خروجی های آزمایش مشهود است که اصلاحات انجام شده و مکانیزم پیشنهادی در بهبود عملکرد ساختار اولیه کاملاً مفید و تأثیرگذار است.

به طور کلی می‌توان دو نتیجه مهم از شکل‌های ۵ و ۶ دریافت کرد. اولاً مدل‌های مبتنی بر یادگیری عمیق عملکرد بهتری در مقایسه با مدل‌های مبتنی بر آمار (ARIMA) دارند. ثانیاً در میان تمام روش‌های بررسی شده، مدل پیشنهادی مقاله بر پایه GRU اصلاح شده (MGRU) به نتایج بهتری دست یافت. خروجی‌ها نشان داد که مدلمان به خوبی می‌تواند وابستگی‌های بلندمدت از داده‌های متوالی دریافت کند. هم‌چنین، مدل TF در آزمایش‌های انجام شده دارای خروجی مناسبی است. البته با توجه به ساختار ترانسفورمرها، این روش در توالی‌های بلندمدت دارای سرعت پایین بوده و هنگام مدل‌سازی دنباله‌های طولانی بسیار کند می‌شود. هم‌چنین حافظه آن‌ها در این نوع مسائل ناکافی و ناکارآمد است که در نتیجه روش بهینه‌ای در این‌گونه سری‌ها به نظر نمی‌رسد.

در ادامه بررسی نتایج، عملکرد مدل پیشنهادی و سایر روش‌های مبتنی بر شبکه‌های عصبی بازگشتی را برای پیش‌بینی قیمت نقطه‌ای با سطوح مختلف طول پیش‌بینی در شکل‌های ۷ و ۸ نشان داده ایم. ابتدا در شکل ۷ مشاهده می‌کنیم که روش RNN در مقایسه با روش‌های دیگری که بررسی شده کم‌ترین زمان آموزش را به دلیل سادگی ساختار خود دارد و پس از آن مدل پیشنهادی ما دارای زمان آموزشی کم‌تری به دلیل نوع ساختار آن است. این ویژگی هنگامی دارای اهمیت است که توأمای خطای کم‌تری هم داشته باشد.



شکل ۷: میانگین زمان آموزش چهار روش مختلف بر پایه شبکه عصبی بازگشتی با دو طول پیش‌بینی مختلف  
Figure 7. Average training time of four different methods based on RNN with two different prediction lengths



شکل ۸: دقت پیش‌بینی (RMSE) روش‌های مختلف مبتنی بر یادگیری عمیق با سطوح مختلف طول پیش‌بینی  
Figure 8. Prediction accuracy (RMSE) of different RNN-based methods with various levels of prediction length

در شکل ۸ می‌بینیم که با توجه به نوسان بالای مقادیر در دیتا سنترهای ابر، استفاده از روش RNN موجب پیش‌بینی داده با خطای بیشتر نسبت به سایر روش‌های در نظر گرفته شده است در حالی که روش پیشنهادی مقاله کم‌ترین خطا را دارد. لذا دو ویژگی توأم دقت و سرعت، کارایی مدلمان را تایید می‌کند. هم‌چنین در شکل ۸ واضح است برای همه مدل‌ها خطا با

مقیاس زمانی افزایش می یابد اما دقت پیش بینی مدلمان با افزایش طول پیش بینی عملکرد بهتری نسبت به سایر روش های بررسی شده دارد که نشان دهنده ویژگی موثر ساختار آن در محوشدگی گرادیان است.

## ۵- نتیجه گیری

برای ارزیابی عملکرد مدل پیشنهادی از چهار روش یادگیری عمیق LSTM، RNN، Transformer، GRU و همچنین روش آماری ARIMA با استفاده از مجموعه داده های مشابه جهت مقایسه عملکرد استفاده گردید. نتایج نشان می دهد پیش بینی قیمت نقطه ای با استفاده از روش پیشنهادی دقت بالاتری نسبت به سایر روش های مورد بررسی دارد. نتایج نشان داد که تکنیک های یادگیری عمیق توانایی خوبی در پیش بینی داده های دنیای واقعی داشته که در این بین ساختار پیشنهادی بهترین عملکرد را نمایش می دهد. دقت روش پیشنهادی نشان از قابلیت اطمینان بالاتر و در نتیجه حفظ کیفیت سرویس دارد. علاوه بر این با رشد طول پیش بینی، در بررسی مدل های مختلف خطای پیش بینی در مدل پیشنهادی رشد کم تری داشت که نشان دهنده توانایی بالای آن در مسائلی با وابستگی های حافظه بلند مدت است. این ویژگی می تواند به حفظ شرایط تغییر پویا در طول دوره های تاریخچه قیمت کمک کند و پیش بینی دقیق با کم ترین خطا انجام شود. استفاده از یک روش کارآمد، به کاربران امکان غلبه بر نوسان بالای نمونه نقطه ای و در نتیجه پیش بینی دقیق روند قیمت در آینده را می دهد. از این طریق می توان با تنظیم زمان خرید بهترین قیمت را ارائه نمود تا از قیمت پیشنهادی بیش از حد که منجر به هزینه بالا و یا پیشنهادی کم تر از حد که منجر به عدم امکان استفاده از نمونه شود، جلوگیری کرد. در نتیجه کاربر می تواند با اطمینان بیشتری جهت حفظ کیفیت سرویس خود از مزیت قیمت پایین این الگو بهره مند گردد. هدف کار آینده مطالعه بیشتر بر پیش بینی دقیق تر و موثر قیمت نقطه ای از جنبه های دیگر و گسترش آن به سری های زمانی دیگر است. علاوه بر این به بررسی تلفیقی سایر مدل های قیمت گذاری خواهیم پرداخت.

## مراجع

- [1] L. Teylo, L. Arantes, P. Sens and L. Drummond, "A dynamic task scheduler tolerant to multiple hibernations in cloud environments," *Cluster Computing*, vol. 24, no. 2, pp. 1051-1073, 2021, doi: 10.1007/s10586-020-03175-2
- [2] J.P.A. Neto, D.M. Pianto and C.G. Ralha, "A prediction approach to define checkpoint intervals in spot instances," *11th International Conference on Cloud Computing (CLOUD SCF)*. Springer, vol. 10967, 2018, pp 84-93, doi: 10.1007/978-3-319-94295-7\_6.
- [3] J. Lancon, J. Kunwar, D. Stroud, M. McGee and R. Slater, "AWS EC2 instance spot price forecasting using LSTM networks," *SMU Data Science Review*, vol. 2, no. 2, 2019.
- [4] V. K. Singh and K. Dutta, "Dynamic Price Prediction for Amazon Spot Instances," *48th Hawaii International Conference on System Sciences, Kauai, HI, USA, 2015*, pp. 1513-1520, doi: 10.1109/HICSS.2015.184.
- [5] P. Varshney and Y. Simmhan, "AutoBoT: Resilient and Cost-Effective Scheduling of a Bag of Tasks on Spot VMs," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 7, pp. 1512-1527, July 2019, doi: 10.1109/TPDS.2018.2889851.
- [6] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," *Applied Soft Computing*, vol. 11, no. 2, pp. 2664-2675, 2011, doi: 10.1016/j.asoc.2010.10.015.
- [7] Y. Liu, Z. Wang and B. Zheng, "Application of Regularized GRU-LSTM Model in Stock Price Prediction," *IEEE 5th International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2019, pp. 1886-1890, doi: 10.1109/ICCC47050.2019.9064035.
- [8] G. Dai, C. Ma and X. Xu, "Short-Term Traffic Flow Prediction Method for Urban Road Sections Based on Space-Time Analysis and GRU," in *IEEE Access*, vol. 7, pp. 143025-143035, 2019, doi:

- 10.1109/ACCESS.2019.2941280.
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones and A. Gomez, "Attention is all you need," *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, pp 5998–6008, 2017.
- [10] V. K. Singh and K. Dutta, "Dynamic Price Prediction for Amazon Spot Instances," *48th Hawaii International Conference on System Sciences, Kauai, HI, USA, 2015*, pp. 1513-1520, doi: 10.1109/HICSS.2015.184.
- [11] J.L. Lucas-Simarro, R. Moreno-Vozmediano, R.S. Montero and I.M. Llorente, "Cost optimization of virtual infrastructures in dynamic multi-cloud scenarios," *Concurr Comput Pract Exp*, vol. 27, no. 9. pp. 2260–2277, doi: 10.1002/cpe.2972.
- [12] V. Khandelwal, A.K. Chaturvedi and C. P. Gupta, "Amazon EC2 Spot Price Prediction Using Regression Random Forests," in *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 59-72, Jan.-March 2020, doi: 10.1109/TCC.2017.2780159.
- [13] S. Alkharif, K. Lee and H. Kim, "Time-series analysis for price prediction of opportunistic Cloud computing resources," *7th International Conference on Emerging Databases. Springer*, vol. 461, pp. 221–229, 2018, doi: 10.1007/978-981-10-6520-0\_23.
- [14] W. Liu, P. Wang, Y. Meng, C. Zhao and Z. Zhang, "Cloud spot instance price prediction using kNN regression," *Hum Cent Comput Inf Sci*, no. 10, no. 1, pp.10–34, 2020, doi: 10.1186/s13673-020-00239-5.
- [15] R. M. Wallace *et al.*, "Applications of neural-based spot market prediction for cloud computing," 2013 *IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, Berlin, Germany, 2013, pp. 710-716, doi: 10.1109/IDAACS.2013.6663017.
- [16] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," in *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841-851, Jan. 2019, doi: 10.1109/TSG.2017.2753802.
- [17] H. Al-Theiabat, M. Al-Ayyoub, M. Alsmirat and M. Aldwair, "A Deep Learning Approach for Amazon EC2 Spot Price Prediction," *IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, Aqaba, Jordan, 2018, pp. 1-5, doi: 10.1109/AICCSA.2018.8612783.
- [18] A. Adebisi, A. Adewumi and C. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, vol. 1, pp. 1-7, 2014, doi: 10.1155/2014/614342.
- [19] Y. Guo and W. Yao, "Applying gated recurrent units approaches for workload prediction," *IEEE/IFIP Network Operations and Management Symposium*, Taipei, Taiwan, 2018, pp. 1-6, doi: 10.1109/NOMS.2018.8406290.
- [20] D. Kong, S. Liu and L. Pan, "Amazon Spot Instance Price Prediction with GRU Network," *IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Dalian, China, 2021, pp. 31-36, doi: 10.1109/CSCWD49262.2021.9437881.
- [21] B. Song, Y. Yu, Y. Zhou, Z. Wang and S. Du, "Host load prediction with long short-term memory in cloud computing," *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6554–6568, 2018, doi: 10.1007/s11227-017-2044-4.
- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp.1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [23] H. Abbasimehr and R. Paki, "Improving time series forecasting using LSTM and attention models," *J. Ambient Intell Human Comput*, vol. 13, no. 1, pp. 673-691, 2022, doi: 10.1007/s12652-020-02761-x.
- [24] K. Cho, B.V. Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase



- representations using RNN encoder decoder for statistical machine translation,” *arXiv* , 2014, doi: 10.48550/arXiv.1406.1078.
- [25] Z. Chen, J. Hu, G. Min, A. Y. Zomaya and T. El-Ghazawi, “Towards Accurate Prediction for High-Dimensional and Highly-Variable Cloud Workloads with Deep Learning,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 4, pp. 923-934, April 2020, doi: 10.1109/TPDS.2019.2953745.
- [26] T. Pham, T. Tran, D. Phung and S. Venkatesh, “Faster training of very deep networks via p-Norm gates,” in *23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 3542-3547, doi: 10.1109/ICPR.2016.7900183.
- [27] D.A. Monge, E. Pacini, C. Mateos and C.G. Garino, “Meta-heuristic based autoscaling of cloud-based parameter sweep experiments with unreliable virtual machines instances,” *Comput. Electr. Eng.*, vol. 69, pp. 364–377, 2019, doi: 10.1016/j.compeleceng.2017.12.007.

---

#### COPYRIGHTS

©2025 by the authors. Published by the Islamic Azad University Bushehr Branch. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0>

---

