تاریخ دریافت: ۹۷/۰۳/۱۱
تاریخ پذیرش: ۹۷/۰۸/۲۶

رویکرد تامین منبع بهبود یافته برای برنامه‌های کاربردی چندلایه با استفاده از سیستم استنتاج عصبی– فازی تطبیق پذیر در محیط رایانش ابری

سمانه کربلائی‌مهدی*
کارشناس ارشد،گروه مهندسی کامپیوتر، واحد محلات، دانشگاه آزاد اسلامی، مرکزی، ایران.
پست الکترونیکی: samaneh.karbalaie@gmail.com

مصطفی قبائی آرانی
استادیار، گروه مهندسی کامپیوتر، واحد قم، دانشگاه آزاد اسلامی، قم، ایران.
پست الکترونیکی: m.ghobaei@qom-iau.ac.ir

چکیده

رایانش ابری، امکان دسترسی به محدوده وسیعی از منابع مجازی شده (سخت‌افزار، نرم‌افزار، سرویس‌ها و...) را فراهم می‌آورد که این منابع می‌توانند به صورت پویا و با حجم متغیر، به‌کاربران خود سرویس‌دهند، یعنی هر کاربر به اندازه‌ای که از آن منابع نیاز دارد، از آن استفاده می‌کند که این کار به استفاده بهینه از منابع منجر می‌شود. تأمین خودکار منابع برای برنامه‌های کاربردی چندلایه در محیط‌های ابری، چالش‌های جدیدی را در زمینه تعادل پویا و یا تخصیص منابع مطرح می‌کند که در تامین برنامه‌های تک لایه شاهد آن نبوده‌ایم. سازوکارهای انعطاف‌پذیر و عموماً خودکار نیاز به تعیین مقدار منابع مجازی مورد نیاز برای به حداقل رساندن مصرف منابع و برآورده کردن توافقنامه سطح کیفی خدمات دارند. در این مقاله، به ارائه رویکردی بهبود یافته برای تامین خودکار منابع برای برنامه‌های کاربردی چندلایه پرداخته‌ایم، رویکرد ارائه شده دارای سه مرحله پایش، تحلیل و اجراست که در مرحله تحلیل از سیستم استنتاج عصبی–فازی تطبیق پذیر استفاده می‌شود. این سیستم از قدرت آموزش شبکه عصبی و مزیت زبانی سیستم‌های فازی به منظور تحلیل فرآیندهای پیچیده به صورت بسیار قدرتمند عمل می‌کند. سپس رویکرد پیشنهادی را تحت بارهای کاری واقعی و مصنوعی با دو روش ADRP_Fuzzy و ADRP_AL مورد ارزیابی و مقایسه قرار دادیم. نتایج به‌دست آمده نشان می‌دهد که رویکرد پیشنهادی موجب کاهش زمان پاسخگویی به میزان ۲,۳ درصد و افزایش بهره‌وری به میزان ۱,۹ درصد می‌گردد.

واژه‌های کلیدی: تامین منابع، برنامه‌های کاربردی چندلایه، مقیاس‌پذیری، سیستم استنتاج عصبی– فازی تطبیق پذیر.

۱- مقدمه

رایانش ابری نسل جدیدی از مراکز داده با گره‌های مجازی‌سازی شده است که دارای مجموعه‌هایی از منابع هستند[۱و۲] که به دریافت کنندگان سرویس خود، امکان دسترسیِ بر حسب تقاضا را در هر منطقه مکانی به منابع

* نویسندهٔ مسئول

محاســباتی گســترده می‌دهد. همچنین قابلیت پیکربندی مجدد، اشــتراک نظیر شــبکه‌ها و فضای ذخیره‌سازی و برنامه‌هــای کاربردی، ارائه انواع ســرویس‌های مبتنی بر محتوا و سایر خدمات حوزه فناوری اطلاعات و ارتباطات را برای کاربران خود فراهم می‌آورد. این منابع می توانند بــا بالاترین ســرعت و کمترین تــلاش مدیریتی و دخالت کاربران از ســوی ارائه‌دهنده سرویس، تخصیص یافته یا آزاد شــوند. به عبارتی، باید منابع مورد نیاز را به‌کاربران تخصیــص دهد و اگر کاربری مقداری بیشــتر از منابع را خواستار باشد، باید آن منابع بدون آن‌که سرویس کاربر با مشکل مواجه شود به مشتری تخصیص یابد و در صورتی که منابع مورد اســتفاده بیش از حد مورد نیاز باشــد باید منابع اضافی به صورت موقت خاموش شده تا در صورت نیاز مجددا مورد استفاده قرارگیرند[۳].

از آنجایــی کــه برنامه‌هــای کاربــردی، مخصوصا برنامه‌هــای کاربردی تحت وب چندلایــه، دارای الگوهای حجم کاری منظمی نیســتند، بنابراین عملیات مقیاس‌بندی (افزایــش یــا کاهش مقیــاس) باید به صــورت بیدرنگ و بــا حداقل دخالت انســان انجام گیرد، تا منابع در اســرع وقت بــرای برنامه‌های کاربردی تامین شــود. به این نوع مقیاس‌بندی کردن منابع، که با کمترین دخالت انسان و به صورت خودکار انجام می‌شود، مقیاس‌بندی خودکار گفته می‌شــود. مقیاس‌بندی خودکار، راهبرد اصلی در راستای تامیــن پویای منابع برای برنامه‌هــای کاربردی تحت وب است.. هدف مقیاس‌بند خودکار، تطبیق پویای منابع انتساب داده شــده بــه برنامه‌های کاربردی بر اســاس بار کاری ورودی اســت و باید توازنی بین بــرآورده کردن اهداف توافقات سطح سرویس و حداقل‌سازی هزینه، برقرار نماید. هــر مقیاس‌بندی خودکار، با چالش‌های زیر روبرو خواهد شد:

«کســر تامین»[1]: در این مــورد، برنامه‌کاربردی، منابع کافی برای پردازش همه درخواست‌های ورودی را با توجه به محدودیت‌های زمانی اعمال شده توسط توافقات سطح سرویس ندارد. در مورد ترافیک‌های ناگهانی، کسر تامین منابع برای برنامه‌کاربردی ممکن اســت منجر به تخطی از توافقات سطح سرویس شود.

«اضافه تامیــن»[2]: در این مورد، برنامه کاربردی منابع بیشتر از نیاز برای برآورده کردن توافقات سطح سرویس دارد. اگر چه این حالت، برای برآورده کردن اهداف توافقات سطح سرویس مناسب است اما به دلیل وجود منابع بیکار، مشتری باید هزینه‌های غیرضروری پرداخت نماید.

«نوسان»[3]: این مورد، ترکیبی از دو پدیده نامطلوب کسر تامین و اضافه تامین است. نوسان، هنگامی‌اتفاق می‌افتد که اعمال مقیاس‌بندی خیلی ســریع انجام شوند قبل از این‌که تاثیر هر عمل مقیاس‌بندی را روی برنامه کاربردی ببینیم. استفاده از ظرفیت میانگیر، از معروف‌ترین رویکردها برای اجتناب از نوسان است[٤].

بنابراین مبحثی که در خور توجه اســت مساله تامین منابع ابری به صورت بهینه می‌باشد تا تمام کاربران بتوانند از هر منبعی بر اساس میزان مورد نیازشان استفاده کنند. در صورتی که این منابع بــه صورت بهینه برای کاربران تامین و مدیریت شوند شاهد کاهش بسیار چشمگیری در معیارهایــی همچون، افزایش «بهره‌وری»[4] و کاهش «زمان پاسخگویی»[5] خواهیم بود.

در این مقالــه، رویکردی بهبود یافته برای تامین منابع در برنامه‌های کاربردی چندلایه با اســتفاده از سیســتم استنتاج عصبی- فازی تطبیق‌پذیر در محیط رایانش ابری ارائه داده‌ایم. رویکرد پیشــنهادی، داری سه مرحله پایش، تحلیل و اجراست، در مرحله تحلیل از سیستم‌های استنتاج عصبی- فازی تطبیق‌پذیر استفاده می‌کنیم با توجه به این‌که در این سیســتم‌ها از قدرت آموزش شبکه عصبی و مزیت زبانی سیستم‌های فازی به منظور تحلیل فرآیندهای پیچیده بســیار قدرتمند استفاده می‌شود بعد از ارزیابی و مقایسه روش پیشــنهادی با دو روش ADRP_Fuzzy و ADRP_AL

1- Under-Provisioning
2- Over-Provisioning
3- Oscillation
4- Utilization
5- Response Time

تحت بارهای کاری واقعی و مصنوعی شاهد کاهش زمان پاســـخگویی و افزایش بهره‌وری به میزان قابل ملاحظه‌ای هستیم.

ادامه این مقاله بدین صورت ســـازماندهی شده است: بخش دوم به‌کارهای مربوط اختصاص دارد، بخش سوم مقدمه‌ای بر سیستم اســـتنتاج عصبی-فازی تطبیق‌پذیر و بخش چهارم رویکرد پیشـــنهادی را تشریح خواهیم نمود، در بخـــش پنجم، بـــه ارزیابی کارایی رویکرد پیشـــنهادی خواهیم پرداخت و نهایتا بخش ششـــم بـــه نتیجه‌گیری و پیشنهادها تعلق دارد.

۲- کارهای مربوطه

تحقیقات متنوع و زیادی در ســـال‌های اخیر در رابطه با تامین منابع برای برنامه‌های کاربردی چندلایه صورت گرفته است، بیشـــتر کارهای موجود براساس تکنیک‌هایی اســـت که در خصوص مدیریت منابع اســـتفاده شده است که هر یک از روش‌های موجود مزایا و معایبی دارد که بر تصمیم‌گیری در مورد انتخاب تکنیک، تاثیر خواهد گذاشت. در این بخـــش مروری بر تحقیقات اخیـــر در زمینه تامین منابع برنامه‌های کاربردی چندلایه خواهیم داشت.

Dong Huang و همکارانش در سال ۲۰۱٤ [٤] به بررسی در خصـــوص مدیریت منبع در کاربردهای وب چندلایه‌ای نمودند تا ببیننـــد که الگوریتم‌های تخصیص منبع می‌تواند چالش‌های موجود در این خصوص را برطرف نماید، طبق بررســـی‌ها چهار چالش مهم در این زمینه وجود دارند که عبارتنـــد از: ۱-رده‌های مختلفی از منابع که دارای تاثیرات مختلفی بر روی کیفیت ســـرویس می‌باشـــد ۲-تخصیص منبع برای کاربردهای چندلایه‌ای در مقایسه با کاربردهای تک‌لایه‌ای دشـــوار است زیرا ســـطح تقاضای منبع در هر لایـــه متفاوت اســـت ۳-اســـتفاده از روش‌های نامناســـب موجب افت کارایی و کاهش نرخ استفاده از منبع می‌شوند ٤- ســـرویس‌دهندگان ابـــری قادر به تضمیـــن کارایی بر اساس کارایی سطح کاربرد نیســـتند. پهنای باند ورودی/ خروجی نیز پویا و غیرقابل پیش‌بینی است.

رحیمی‌زاده و همـــکاران در ســـال ۲۰۱۵[٥]، یک مدل تحلیلـــی بر اســـاس"QN"[6] بـــه منظور تخمیـــن معیارهای کیفیت ســـرویس برای برنامه‌های کاربـــردی چندلایه در یک مرکزداده مجازی ارائه نمودند. همچنین یک روشـــگان جهت اندازه‌گیری ویژگی‌های "VMTA"[7] از درخواســـت‌ها بین لایه‌ها و معیارهای عملکرد برنامه‌های کاربردی ارائه نمودند ســـپس آزمایش‌های مختلفی را به منظور ارزیابی عملکرد تلفیقـــی لایه‌های VMTAها، در خصوص حجم کار تدریجی و انفجاری انجام دادند

Marta Beltran در ســـال ۲۰۱۵[٦]، جهت تامین منابع برنامه‌های کاربردی چندلایه، تکنیکی را ارائه داده است که در آن تامین برنامه به صورت خودکار و مجازی صورت می‌پذیرد. تمام تصمیم‌گیری‌ها نیـــز به صورت کاملا پویا و بر اســـاس شرایط و بدون دخالت انسان انجام می‌شود. تکنیکی که توســـط وی ارائه گردیده است، AutoMap نام دارد.

Marwah Hashim Eawna و همـــکاران در ســـال ۲۰۱۵[۷]، با اســـتفاده از تکنیک‌های فراشـــهودی همچون الگوریتم‌هـــای "SA"[8]، "PSO"[9] و ترکیبـــی بـــه تامین منابع پویـــا در برنامه‌های تک لایه و چندلایه پرداختند. الگوریتم ترکیبی شامل ترکیب دو الگوریتم SA و PSO می‌باشد. آن‌ها دو معماری کلی شامل تک‌لایه و چندلایه را بیان کردند که معماری تک‌لایه ساختار نسبتا ساده دارد و راه‌اندازی آن آسان اســـت به عبارتی همه چیز در پایگاه داده، برنامه و ارائه در محل خودشان قرار گرفته‌اند.

Sukhpal Singh و همـــکاران در ســـال ۲۰۱۵[۸]، بـــه بررسی تامین منابع مناسب در حجم کارهای ابری وابسته بـــه نیازمندی‌های کیفیت ســـرویس پرداختند. پارامترهای کیفیت ســـرویس بر اســـاس تکنیک تامین منبع به منظور تامیـــن موثر منابـــع مورد نیاز اســـت. آن‌ها بـــه تحلیل و

6-Queuing Network
7- Virtualized Multi-Tier Application
8- Particale Swarm Optimization
9- Simulated Annealing

طبقه‌بندی بر اســـاس الگوهای رایج و تامین حجم کار ابر قبل از برنامه‌ریزی واقعی پرداختند. از آنجایی که پیچیدگی مدیریت منابع در ابرها روز به روز در حال افزایش است، یک تکنیک موثر به منظور مدیریت منابع نیاز است.

اقبال و همکاران در ســـال ۲۰۱۱[۹]، روشـــگانی برای تامین پویای منابع برای برنامه‌های کاربردی چندلایهٔ فقط خواندنی، به منظور کشـــف و تحلیل خـــودکار گلوگاه‌ها و مقابلـــه با منابع با ظرفیت اضافی شـــده، به منظور کاهش زمان پاسخ پیشـــنهاد دادند. علت استفاده از پارامتر زمان پاســـخ به‌عنوان پارامتـــر هدف، الگوهای ترافیکی بســـیار پویای برنامه‌های کاربردی تحت وب و پیچیدگی ذاتی آن‌ها می‌باشد. روشـــگان آن‌ها برای تامین پویای منابع، از مدل واکنشـــی[۱۰] برای بالابردن مقیـــاس و از مدل پیش‌گویانه[۱۱] برای پایین‌آوردن مقیاس استفاده کرده‌اند. آن‌ها، روشگان پیشـــنهادی خود را بر روی بستر EUCALYPTUS به همراه بارکاری ترکیبی، ارزیابی کردند. نتایج آزمایش‌ها نشـــان می‌دهد که روشگان پیشنهادی، زمان پاسخ مشخص شده در SLA را تضمیـــن کـــرده و در عین حال بهره‌وری را نیز بهبود می‌بخشد.

Sireesha Muppala و همکارانش در سال ۲۰۱٤[۱۰]، یک رویکـــرد تامین منبع تطبیقی به همـــراه کنترل پذیرش هماهنگ شـــده برای ســـرویس‌های چندلایه‌ای ارائه دادند. رویکرد پیشنهادی آن‌ها بر اساس تکنیک یادگیری تقویتی مستقل از مدل برای پیکربندی خودکار ماشین‌های مجازی طراحی شده بود. آن‌ها برای چابک‌سازی روش پیشنهادی خود از شـــبکه عصبی آبشـــاری اســـتفاده کردنـــد. نتایج آزمایش‌های آن‌ها نشان می‌دهد که رویکرد پیشنهادیشان در بارهای کاری انفجاری موثرتر، چابک‌تر و مقیاس‌پذیرتر نسبت به سایر رویکردها عمل می‌کند.

Jing Bi و همکارانـــش در ســـال ۲۰۱۳، بـــه مســـئلهٔ تخصیص منابع مجازی‌شـــده برای مراکـــز داده ابری در مقیاس بزرگ "CDC"[۱۲]ها [۱۱] پرداختند. آن‌ها یک معماری مدیریت خودمختار را بر اســـاس سازوکار مجازی‌سازی برای برنامه‌های وب چندلایه در CDCها به منظور افزایش تخصیص منابع ارائه دادند که این معماری می‌تواند اجرای موثر محیط‌های ســـرویس برنامه‌مجازی شده"VASE"[۱۳] و نیازمندی‌های کارخواه‌های SLA را تضمین کند سپس یک مدل صف ترکیبـــی انعطاف‌پذیر ارائه دادند و به‌طور کامل به زمان پاسخ، توان عملیاتی، هزینه و درآمد مشخص شده SLA پرداختند. پس از آن یک مسئلهٔ بهینه‌سازی غیرخطی محدود شده را بیان کردند.

از الگوریتـــم شـــهودی با هـــدف حداکثر کردن ســـود کل"CIP"[۱٤]هـــا جهت حل مســـئله اســـتفاده کردنـــد که این الگوریتم بر اساس تغییرات پویای حجم کار منابع مجازی شـــده را تخصیص می‌دهد. آن‌هـــا تاثیر راهبرد تخصیص پویای منبع را بر اساس ماشین‌های مجازی "DVM-RA"[۱۵] با دو راهبرد "Stat-RA"[۱٦] و "DPM-RA"[۱۷] مقایسه کردند.

تامین پویای منبع یک تکنیک چالشی برای نیازمندی‌های SLA از برنامه‌هـــای کاربـــردی چندلایه در ابـــر مبتنی بر مجازی‌سازی است. در همین راستا Heng Wu و همکارانش در سال ۲۰۱۳[۱۲] به ارائه یک روش آگاه از سود با نظریهٔ کنترل بازخورد پرداختند که طبق روش ارائه شـــدهٔ آن‌ها، هزینه تامین منبع در مقایسه با روش هزینه آشکار تا ۳۰٪ کاهش می‌یابد و به‌طور موثر می‌تواند موارد نقض SLA را در مقایسه با روش آگاه از هزینه کاهش دهد.

تغییـــرات درخواســـت هـــا از ســـوی مســـتاجران و استفاده‌کنندگان متفاوت است، بنابراین انعطاف‌پذیری برای آن‌ها افزایش می‌یابد. آن‌ها یک مدل تخریب کارایی را مبتنی بر یادگیری ماشین جهت محاسبه هزینه انتقال ایجاد کردند سپس یک تابع ســـود و هزینه را برای همه پیکربندی‌های مجدد ارائه کردند. در نتیجه حداقل هزینه با بیشترین سود را به دست آوردند.

10- reactive
11- predictive
12- Cloud Data Center
13- Virtualised Application Service Environment
14-Cloud Infrastructure Provider
15-Dynamically Resourse Allocation Strategy Based On Virtual Machine
16- Static Resource Allocation Strategy
17- Dynamically Resource Allocation Strategy Based On Physical Machine

Bhuvan Urgaonkar و همکاران[۱۳]، تکنیکی پویا برای برنامه‌های چندلایه اینترنت را ارائه نمودند که شـامل یک مدل صـف انعطاف‌پذیر برای تعییـن چگونگی تخصیص منابـع به هر لایه برنامه‌کاربـردی و ترکیبی از روش‌های پیش‌بینی و واکنش‌پذیر جهت تعیین زمان به منظور تامین منابع در مقیاس‌های زمانی کوچک و بزرگ است. با توجه به روشـی که آن ها ارائه دادند اهداف زمان پاسـخ حفظ می‌شـود و سربار کاهش می‌یابد همچنین سربار تعویض کارسـازها از چندین دقیقه به کمتر از یک ثانیه می‌رسـد. باتوجـه به این که تغییرات حجم کار پویاسـت و تغییرات بارکاری در زمان‌های مختلف متفاوت است بنابراین تکنیک پیشنهادی آن‌ها مناسب است. همچنین سازوکارهای تامین پیش‌بینی و واکنشی به‌عنوان معماری مرکز داده بر اساس پایش ماشین‌مجازی ارائه شده است.

مطالعه تحقیقات فعلی نشـان می‌دهد که کسـر تامین، اضافه تامین و نوسان در تامین منابع برنامه‌های کاربردی چندلایه به‌عنوان مسئله‌ای چالش برانگیز مطرح بوده است زیرا منابع باید به صورت بهینه تامین شوند تا تمام کاربران بتوانند از هر منبعی بر اساس میزان مورد نیازشان استفاده کنند. رویکردهای قبلی، هـر کدام مزایا و معایبی دارند که آن‌ها را به‌طور مختصر در جدول (۱) آورده‌ایم. در روش پیشـنهادی این مقاله، از سیستم اسـتنتاج عصبی- فازی تطبیق‌پذیر اسـتفاده کرده‌ایم که بعد از ارزیابی و مقایسه شاهد کاهش زمان پاسخگویی و در نتیجه بهبود در تامین منابع بوده‌ایم.

۳- مقدمـه‌ای بر «سیسـتم اسـتنتاج عصبـی- فازی تطبیق‌پذیر»[۱۸] (ANFIS) [۱۴]

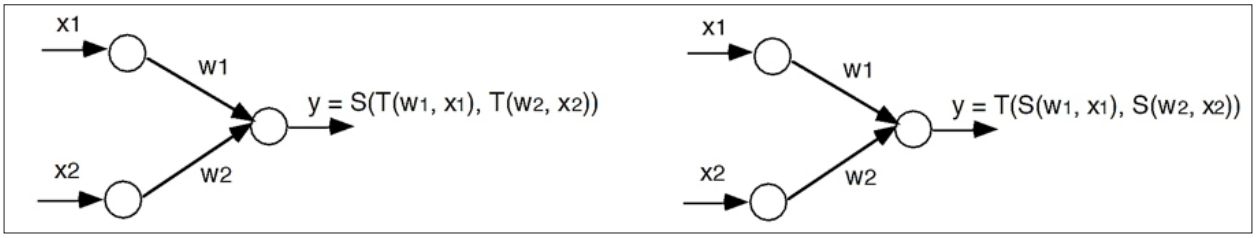
شبکه‌های عصبی مصنوعی و سیستم‌های فازی از نظر سـاختاری تا حد زیادی با یکدیگر متفاوت هسـتند، اما با توجـه به نقاط ضعف و قوت آن‌هـا، می‌توان گفت این دو سیسـتم، دارای ماهیت مکمل نسـبت به یکدیگر هستند. با ایجاد شبکه عصبی فازی، استفاده از عبارات به‌کار گرفته شـده در زبان طبیعی برای تشـریح مفاهیمی که معمولاً دارای ابهام و عدم قطعیت هستند در اجزای شبکه عصبی مصنوعی (ورودی، خروجی، نورون و ...) محقق می‌شود. انجـام این امر با تغییراتی ویژه در اجزای شـبکه عصبی مصنوعـی رخ می‌دهد؛ برای مثال در حالی‌که شـبکه‌های عصبی معمولی از نورون‌های یکسان و مشابه هم تشکیل شده‌اند، نورون‌های تشکیل‌دهنده شبکه‌های عصبی- فازی، معمولا نامتجانس هسـتند و شـبکه‌های عصبی- فازی از نورون‌های متنوع که ویژگی‌های محاسـباتی آنها مختلف است (مانند AND و OR) تشکیل می‌شوند شکل (۱).

منطق فازی با اسـتفاده از قوانین «اگر چنین شـود»، می‌توانند دانش انسان را به صورت کیفی مدل‌سازی نموده و ایـن فرآیند بدون به‌کارگیری تحلیل‌های کمّی امکان‌پذیر نمی‌باشد. روش تحلیل عددی برای سامانه‌های فازی ابتدا به وسیله محققان بررسی شد و یک مجموعه از تحقیقات در این رابطه به عمل آمده است، سامانه‌هایی که از این نظریه اسـتفاده می‌کنند می‌توانند قوانین یا دانش را به‌عنوان فرم «اگر چنین شود» دنبال نمایند و مزایایی را می‌توان از تحلیل ریاضی آن در مدل‌سازی‌ها کسب نمود. اما این گونه مدل‌ها می‌توانند از ساختار مدل و انتخاب پارامتر مناسب نیز بهره گیرند. حذف قوانین غیرضروری و انتخاب عناصر ورودی کافی می‌تواند در بهبود عملکردها نقش اساسی داشته باشد و می‌تواند از هزینه محاسبات و تحلیل قوانین موجود بکاهد که این یکی از مهم‌ترین معیارهای سامانه‌های فازی است. ترکیب شبکه‌های عصبی و سامانه‌های فازی، به‌عنوان یک روش قدرتمند برای توسعه سامانه‌های فازی شناخته شده اسـت. در سامانه‌های فازی رابطه میان ورودی- خروجی به صورت واضح به فرم «اگر چنین شود» ظاهر می‌شود، اما در شبکه عصبی به وسیله پارامترها کدبندی نمی‌شود. ترکیـب منطق فازی مبتنی بر قانون با شـبکه‌های عصبی می‌توانـد توانایی شـبکه‌های عصبـی را افزایش دهد. در ضمن، مدل‌سازی عصبی- فازی به‌عنوان یک ابزار قدرتمند شناسایی شده است که می‌تواند در توسعه مؤثر مدل‌ها به

18- Adaptive Neuro Fuzzy Inference System

جدول۱: مقایسه تکنیک‌های موجود در مدیریت منبع به صورت خودکار در برنامه‌های کاربردی چندلایه وب

مرجع	پارامترهدف	روش	بستر آزمایش	نوع تکنیک	مزایا	معایب	توضیحات
Dong Huang و همکاران در سال ۲۰۱۴[۴]	کاهش هزینه منبع	مبتنی بر مدل جهت تامین منبع در رایانش ابری به منظور تضمین SLA با کمترین هزینه منبع	بیان نشده	مبتنی بر مدل	الگوریتم ها می‌توانند تقاضای منبع را با یادگیری داده‌های تاریخچه‌ای تخمین بزنند	نیازمند دانش زیاد جهت مدل‌سازی	بازبینی فعالیت‌ها در رابطه با تخصیص منبع برای کاربردهای وب چندلایه‌ای-تجزیه و تحلیل مزایا و محدودیت‌های روش‌های مبتنی بر مدل و قاعده برای مسئله-تشخیص مسائل باز در مدیریت منبع کاربردهای وب چندلایه‌ای
کیوان رحیمی‌زاده و همکاران در سال ۲۰۱۵[۵]	تخمین معیارهای کیفیت سرویس	استفاده از مدل تحلیلی مبتنی بر شبکه صف	چارچوب معماری مجازی	شبکه صف	افزایش امنیت وکاهش هزینه	تاثیر مستقیم ویژگی لایه‌ها بر عملکرد کلی VMTA و افزایش VM مسدود شده	در این روش از معیارهای کیفیت سرویس استفاده می‌شود.
Marta Beltran در سال ۲۰۱۵[۶]	کاهش تعداد VM	استفاده از مقیاس‌گذاری افقی و عمودی	چارچوب Cloudsim	AutoMap	سادگی فرایند و انعطاف‌پذیری بالا	نیاز به مدل‌سازی هر VM در هر لایه	مدل برنامه کاربردی برای n لایه بر اساس شبکه‌های جدا از صف است.
Marwah Hashim و همکاران در سال ۲۰۱۵[۷]	کاهش زمان اجرا- کاهش هزینه	استفاده از تکنیک فراشهودی	شبیه‌سازی با تکنیک فراشهودی	مبتنی بر الگوریتم PSO، SA و ترکیبی	میانگین زمان اجرای کمتر- قابلیت ترکیب دو الگوریتم PSO و SA	ممکن است امنیت را کاهش دهد.	تامین منبع بر اساس الگوریتم ترکیبی در برنامه چندلایه سریع‌تر از الگوریتم PSO و SA انجام می‌شود.
Singh و همکاران در سال ۲۰۱۵[۸]	کاهش زمان اجرا- کاهش هزینه	استفاده از k-means مبتنی بر الگوریتم خوشه بندی	چارچوب Cloudsim	مبتنی بر RP	کاهش زمان صف‌بندی	امکان نقض SLA و افزایش پیچیدگی سیستم شود.	در این روش حجم کار را بر اساس الگوهای رایج و تامین حجم کار ابر قبل از برنامه‌ریزی واقعی تحلیل و طبقه‌بندی می‌کند.
اقبال [۹]	زمان پاسخ	استفاده از مدل واکنشی برای بالابردن مقیاس و از مدل پیشگویانه برای پایین‌آوردن مقیاس	چارچوب EUCALYPTUS	بیان نشده	زمان پاسخ مشخص شده در SLA را تضمین کرده و در عین حال بهره وری را نیز بهبود می‌بخشد.	بیان نشده	ارائه رویکردی برای تشخیص خودکار گلوگاه ها برای برنامه‌های کاربردی چندلایه خواندن-محور
Muppala[۱۰]	تعداد درخواست کاربران و تعداد ماشین مجازی	رویکرد تامین منبع تطبیقی مبتنی بر یادگیری تقویتی به همراه کنترل پذیرش هماهنگ شده سرویس‌های چندلایه ای	بیان نشده	یادگیری تقویتی(RL) و شبکه‌های عصبی	در بارهای کاری انفجاری موثرتر، چابک تر و مقیاس پذیرتر نسبت به سایر رویکردها عمل می‌کند.	بیان نشده	ابزار شبیه‌سازی MATLAB استفاده از شبکه عصبی برای چابک‌سازی
Jing Bi و همکاران در سال ۲۰۱۳[۱۱]	حداکثر سود-بهبود پاسخ	ارائه یک معماری خودمدیریتی و ساخت یک مدل صف ترکیبی	چارچوب توزیع‌شده و ناهمگون	مبتنی بر VM با راهبرد Stat-RA و DPM-RA	دارای بیشترین تضمین SLA	نیازمند بهینه‌سازی مسئله	دستیابی به ۱۵٪ الی ۲۰٪ سود بیشتر
Heng Wu و همکاران در سال ۲۰۱۳[۱۲]	کاهشهزینه زیرساخت- افزایش سود	ارائه یک روش آگاه از سود با نظریهٔ کنترل بازخورد	الگوریتم کنترل بازخورد مبتنی بر حداکثر سود	یادگیری ماشین	کاهش نقض SLA-افزایش انعطاف پذیری و سود	افزایش پیچیدگی سیستم	کاهش هزینه زیرساخت تا ۳۰٪
Bhuvan Urgaonkar و همکاران در سال۲۰۱۳[۱۳]	کاهش سربار و زمان پاسخ	ارائه یک معماری مرکز داده مبتنی بر پایش VMها	چارچوب میزبانی ۴۰ ماشین مبتنی بر لینوکس	پیش‌بینی و واکنش‌پذیر	مدیریت نوسانات در حجم کار بالا	لزوم ترکیب روش‌پیش بینی و واکنش‌پذیر	کاهش سربار تعویض کارسازها به کمتر از یک ثانیه.



شکل (۱) مدل شماتیک نورون‌های AND (سمت راست) و نورون‌های OR (سمت چپ) [۱۴]

وسیله ترکیب اطلاعات از منابع مختلف مفید باشد و در عین حال با مدل‌های تجربی، داده‌ها و اکتشاف نیز همراه می‌باشد. بنابراین در بسیاری از موارد، مدل‌های عصبی- فازی می‌توانند برای توضیح دادن راه‌حل‌ها برای کاربران به نحو بهتری مورد استفاده قرار گیرند تا استفاده از مدل‌های جعبه سیاه نظیر شبکه‌های عصبی.

سیستم استنتاج عصبی- فازی تطبیق‌پذیر یک شبکه پس‌خور چندلایه می‌باشد و از الگوریتم‌های یادگیری شبکه عصبی و منطق فازی به منظور طراحی نگاشت غیرخطی بین فضای ورودی و خروجی استفاده می‌کند. ANFIS با توجه به توانایی در ترکیب قدرت زبانی یک سیستم فازی با قدرت عددی یک شبکه عصبی، نشان داده است که در مدل‌سازی فرایندهایی همچون داده‌کاوی بسیار قدرتمند می‌باشند. ANFIS شبکه تطبیق پذیر و قابل آموزشی است که به لحاظ عملکرد کاملا مشابه سیستم استنتاج فازی است. برای سادگی کار فرض می‌کنیم که سیستم فازی ما دو ورودی x و y دارد و خروجی آن z است. حال اگر قوانین به صورت رابطه (۱) باشند.

(۱)

$$Rule1: if\ x\ is\ A_1\ and\ y\ is\ B_1\ then\ f_1 = p_1x + q_1y + r_1$$
$$Rule2: if\ x\ is\ A_2\ and\ y\ is\ B_2\ then\ f_2 = p_2x + q_2y + r_2$$

و اگر برای غیرفازی ساز از غیرفازی ساز میانگین مراکز استفاده کنیم خروجی به صورت رابطه (۲) خواهد بود.

$$f = \frac{w_1f_1 + w_2f_2}{w_1 + w_2} = \overline{w}_1f_1 + \overline{w}_2f_2 \quad st \qquad (۲)$$
$$\overline{w}_1 = \frac{w_1}{w_1 + w_2}\ ,\ \overline{w}_2 = \frac{w_2}{w_1 + w_2}$$

ساختار معادل ANFIS به صورت زیر خواهد بود.

لایه ۱: در این لایه ورودی‌ها از «توابع عضویت»[19] عبور می‌کنند.

$$O_{1,i} = \mu A_i(x) \qquad for\ i = 1,2$$
$$O_{1,i} = \mu B_i(x) \qquad for\ i = 3,4 \qquad (۳)$$

توابع عضویت هر تابع پارامتر مناسبی می‌تواند باشد که در اکثر موارد توابع گاوسین انتخاب می‌شوند. مثل تابع زنگی شکل عمومی.

$$\mu A(x) = \frac{1}{1 + \left|\frac{x - c_i}{a_i}\right|^{2b_i}} \qquad (٤)$$

که $\{a_i, b_i, c_i\}$ مجموعه پارامترها هستند. پارامترهای این لایه به «پارامترهای اولیه»[20] معروف هستند.

لایه ۲: خروجی این لایه ضرب سیگنال‌های ورودی است که در واقع معادل قسمت اگر قوانین هستند.

$$O_{2,i} = w_i = \mu A_i(x)\mu B_i(y) \quad i = 1,2 \qquad (۵)$$

لایه ۳: خروجی این لایه نرمالیزه شده لایه قبلی است.

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \qquad (۶)$$

لایه ٤: این لایه خروجی‌ها را از توابع عضویت خروجی عبور می‌دهد.

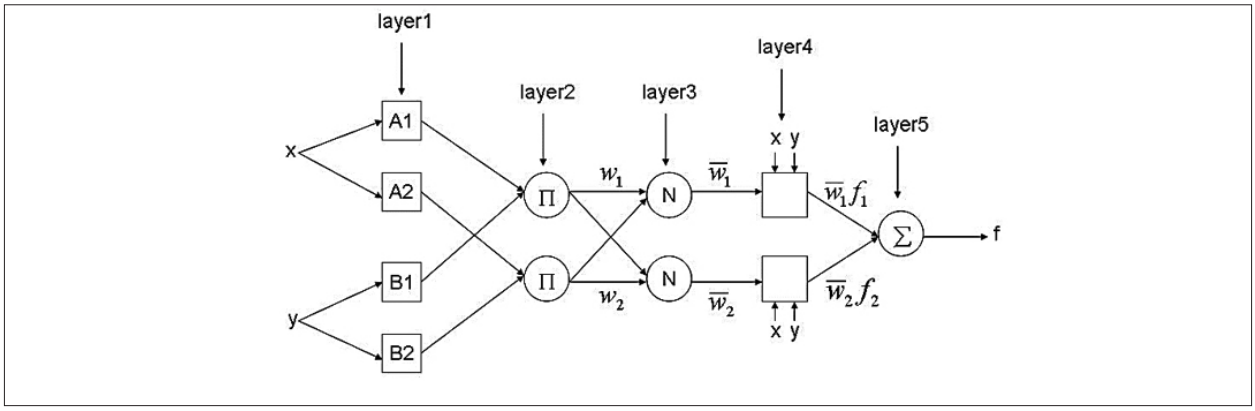
$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i(p_ix + q_iy + r_i) \qquad (۷)$$

لایه ٥: خروجی این لایه خروجی کلی سیستم است.

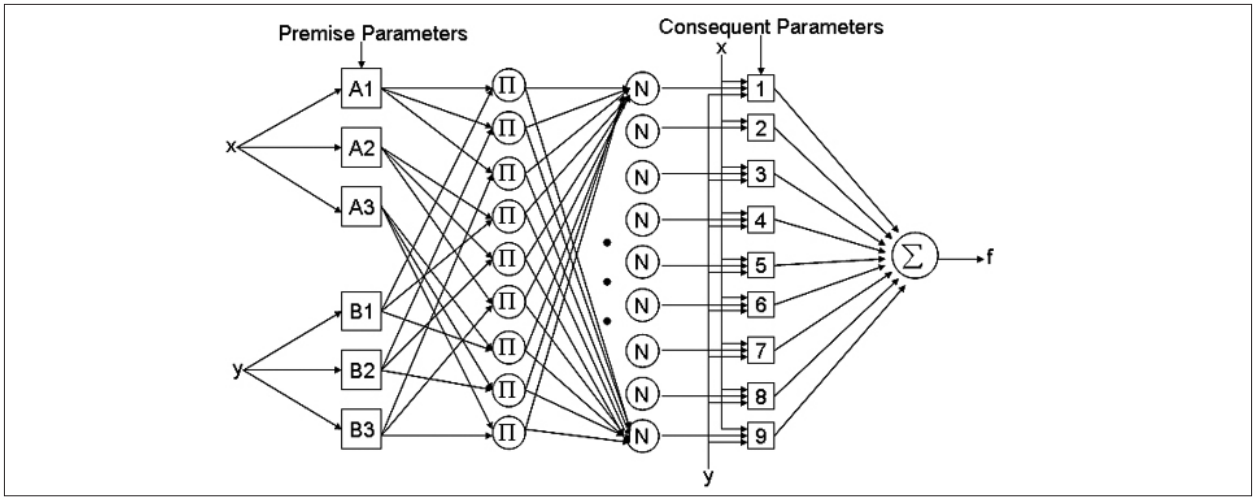
$$O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad (۸)$$

این شبکه براساس یادگیری با نظارت، آموزش داده می‌شود. بنابراین هدف ما آموزش شبکه تطبیقی است که قادر به تخمین توابع نامشخص حاصل از اطلاعات آموزش

19- Membership Functions
20- Premise Parameters



شکل ۲: ساختار معادل ANFIS [۱۴]



شکل ۳: نمایش تعمیم شبکه ANFIS با مثالی که دارای ۲ ورودی و ۳ تابع عضویت[۱۴]

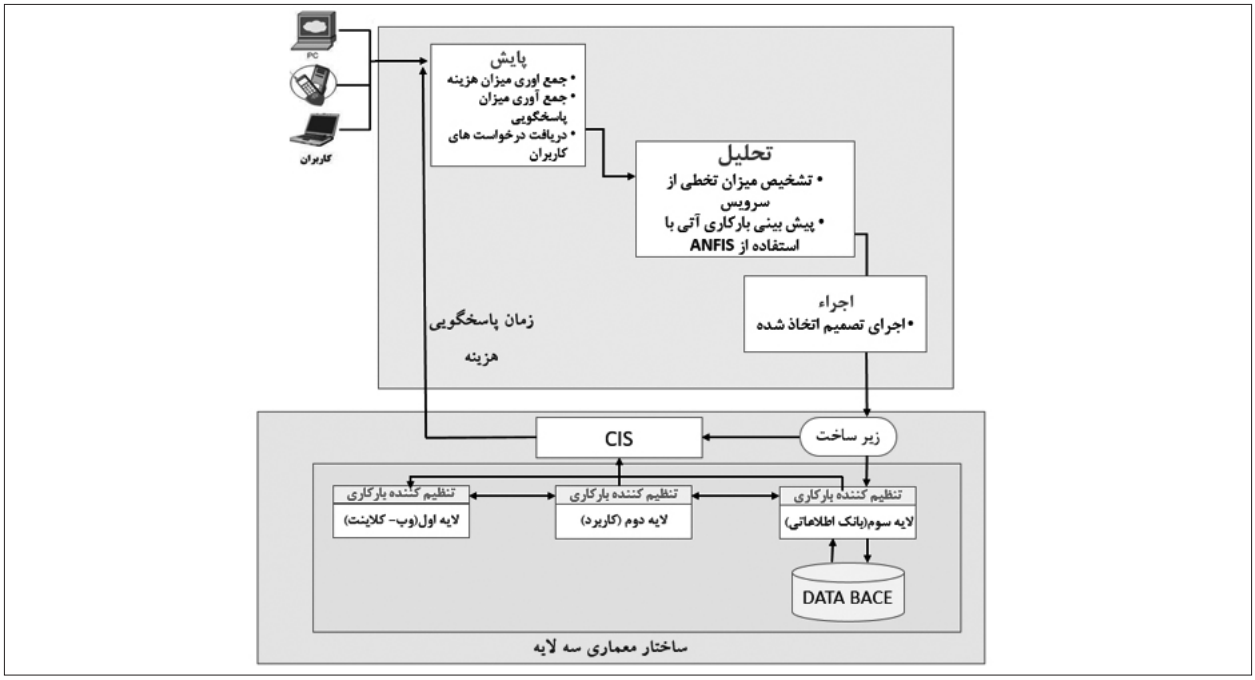
بوده و مقدار دقیقی برای پارامترهای بالا پیدا کنند.

ویژگی متمایزکننده ANFIS، فراهم کردن الگوریتم یادگیری پیوندی، روش شیب گرادیان و روش حداقل مربعات، به منظور اصلاح پارامترها می‌باشد. روش «گرادیان نزولی»[۲۱] به‌کار گرفته می‌شود تا پارامترهای غیرخطی مقدماتی (ai ,bi) را تنظیم کند، درحالی‌که روش حداقل مربعات به‌کار گرفته می‌شود تا پارامترهای خطی بخش تالی را تعیین کند. روند آموزش دو مرحله دارد: در مرحله اول، در حالی‌که پارامترهای بخش مقدم (توابع عضویت) ثابت فرض می‌شوند، با استفاده از روش حداقل مربعات پارامترهای بخش تالی تعیین می‌شوند. سپس سیگنال‌های خطا انتشار می‌یابند. روش گرادیان نزولی استفاده می‌شود تا پارامترهای مقدماتی از طریق حداقل کردن تابع هزینه درجه دوم کلی، اصلاح شود. برای آموزش این شبکه ابتدا در لایه ۱ تمام قوانین موجود را تشکیل می‌دهیم. به‌طور مثال اگر ۲ ورودی داشته باشیم که هر کدام ۳ تابع عضویت داشته باشد ۹ قانون باید تشکیل دهیم که به صورت شکل(۳) خواهد بود.

21- Gradient Descent

۴- رویکرد پیشنهادی

در این بخش رویکردی بهبود یافته مبتنی بر سه مرحله پایش، تحلیل و اجرا به منظور تامین منابع برنامه‌های کاربردی چندلایه ارائه می‌کنیم. بدین صورت که داده‌های مورد نیاز در مرحله پایش برداشت شده و به مرحله تحلیل فرستاده می‌شوند. ما در مرحله تحلیل از سیستم استنتاج عصبی- فازی تطبیق‌پذیر برای پیش‌بینی استفاده کرده‌ایم. سپس بر اساس پیش‌بینی انجام شده و میزان تخطی از SLA اعلام شده، سیستم پیشنهادی تصمیم می‌گیرد که



شکل ۴: چارچوب پیشنهادی

چه تعداد ماشـین‌مجازی جدید اضافه یا حذف شـود، و نهایتا تصمیم گرفته‌شـده در مرحله تحلیل، به مرحله اجرا فرستاده و اجرایی می‌شود.

٤-١ چارچوب پیشنهادی

سـاختار چارچوب پیشـنهادی با نام " ADRP_AN"[۲۲] دارای سـه لایه کارخواه یا وب، لایـه کاربرد و لایه بانک اطلاعاتی است. لایه وب وظیفه دریافت درخواست، ساخت نشست برای کاربر، احراز هویت اولیه و شناخت بدافزار را به عهده دارد. لایه کاربرد وظیفه اجرای برنامه‌های کاربردی و ارائه خدمات به‌کاربر را به عهده دارد. اصلی‌ترین لایه در معماری سـه لایه، لایه کاربرد اسـت که وظیفه ارتباط دو لایه دیگر را به عهده دارد. لایه سـوم، لایه بانک اطلاعاتی اسـت. این لایه وظیفـه مدیریت داده‌ها را بـه عهده دارد. در این سـاختار، هر ماشـین‌مجازی در یکی از این لایه‌ها قرار می‌گیرد. البته ممکن است ماشین‌مجازی با پیکربندی مجدد از لایه‌ای به لایه دیگر مهاجرت کند. در سـاختار ابر "CIS"[۲۳] وظیفه انتقال درخواسـت‌ها را به لایه اول به عهده دارد. تنظیم‌کننده بـارکاری در هر لایه، در صورتی که در خواسـت‌های ورودی به لایه از حد سرویس بالاتر باشد، صفی از درخواسـت ایجاد می‌کند که به واسـطه آن پس از آزاد شـدن هر ماشین مجازی درخواستی را در اختیار آن قـرار می‌دهد. CIS نظارت بر تنظیم‌کننده بارکاری را به عهـده دارد. بدین معنی که اطلاعات را در زمان‌های معینی از تنظیم‌کننده بارکاری هر لایه گرفته و نگهداری می‌کند.

با توجه به شـکل (٤) زمان پاسخگویی درخواست‌ها، هزینه درخواسـت‌ها و میزان بـارکاری به‌عنوان کارآیی سیسـتم در هـر مرحله توسـط سـاختار پایـش از CIS درخواست و در نهایت برداشت می‌شود. داده‌های برداشت شـده توسـط پایش در اختیار تحلیل قرار داده می‌شود. وظیفه تحلیل پیش‌بینی وضعیت آتی سیسـتم بر اسـاس وضعیت فعلی است. عملا تحلیل علاوه بر تشخیص میزان تخطـی از SLA، میزان بار کاری بعدی را نیز با اسـتفاده از سیسـتم اسـتنتاج عصبی- فازی تطبیق‌پذیر پیش‌بینی می‌کند. سـپس بر اسـاس پیش‌بینی انجام شـده و میزان تخطی از SLA اعلام شـده، تصمیم می‌گیـرد که چه تعداد ماشـین‌مجازی جدیـد اضافه یا حذف کنـد (مقیاس‌بندی افقی) یا چه تعداد ماشـین‌مجازی قبـل را مجدد پیکربندی

22- Automatic Dynamic Resource Provisioning Using ANFIS
23- Cloud Information System

جدول ۲: ساختار هر درخواست کاربر

شناسه کاربر	User SLA	
	SLO1	SLO2
	Cost	Deadline Time

(مقیاس‌بندی عمودی) کند. تصمیم‌گیری مرحله تحلیل برای بخش اجراء ارسال می‌شود تا این تصمیم اجرایی شود.

٤-٢ الگوریتم پیشنهادی

در روش پیشنهادی درخواست‌ها در صف درخواست قرار می‌گیرند و به تناسب نیاز به لایه مناسب در اختیار آن لایه قرار می‌گیرند. الگوریتم (۱) نحوه قرارگیری درخواست‌ها را در صف درخواست نشان می‌دهد.

Algorithm 1: Pseudo code for add Request to Queue

```
1: for i ← 1 to n do
2:     for i ← 1 to Req_Count_j do
3:         ADD Req_j^i to Request_queue
4:     end for
5: While(Request_queue is not empty)
6:     foreach req in Request_queue do
7:       Send req to L tier
8:     end for
9: end while
```

در ادامه این بخش، اجزای مهم این ساختار که شامل پایش، تحلیل و اجراء است، بررسی می‌گردد. در روش پیشنهادی، درخواست کاربر مطابق قوانین SLA ساخته می‌شود. هر SLA از تعدادی هدف یا SLO٢٤ تشکیل شده‌است. ساختار درخواست کاربر در این تحقیق شامل دو هدف هزینه و زمان پاسخگویی می‌باشد. کاربر می‌تواند به‌صورت همزمان چندین درخواست را به فراهم‌کننده ابر ارائه دهد. جدول (۲) ساختار هر درخواست کاربر را نشان می‌دهد.

با توجه به ساختار جدول (۲) هر درخواست شامل شناسه کاربر، هزینه در نظرگرفته شده و حد مجاز زمان پاسخگویی برای آن درخواست است. در صورتی که هزینه و زمان پاسخگویی از این حد بالاتر برود، تخطی از «شرایط سرویس»[25] رخ داده است. برای تامین بهینه منابع و کاهش تخطی، ساختار پیشنهادی استفاده می‌شود. الگوریتم (۲) شبه کد این ساختار را به صورت کلی نشان می‌دهد.

Algorithm 2: Pseudo code for Autonomic Resource Provisioning(Workload)

```
1: Initialization: boots an appropriate number of VMs for Layer
2: while (the system is running and in the beginning of interval Δt) do
3:    for i ←1 to 3 do %number of tier=3
4:      [W[i] , R[i] ,C[i]] ←Monitoring (Li.Response, Li.Cost , Workload);
5:      [Predict[i], SLAV[i]] ←Analysis ([W[i] , R[i] ,C[i] ]);
    [scale_type[i] , scale_value[i] ] ← Analysis (Predict[i] ,SLAV[i] );
6: Execution ([scale_type[i] , scale_value[i]]);
7: end for
8: end while
```

٤-٢-١ پایش

وظیفه این بخش از رویکرد پیشنهادی، دریافت بارکاری، زمان پاسخگویی به هر درخواست و هزینه هر درخواست است. بخش پایش پایگاه دانشی در اختیار دارد. این بخش در بازه‌های زمانی یکسان وضعیت سیستم را بررسی می‌کند و متغیرهای مورد نیاز خود را برداشت کرده و آن‌ها را در پایگاه دانش خود ثبت می‌کند. پایگاه دانش این بخش دارای دو بخش است. بخش نگهداری بارکاری و بخش نگهداری زمان پاسخ و هزینه هر درخواست. ساختار رکوردهای پایگاه دانش بخش اول به صورت جدول (۳) و ساختار رکوردهای پایگاه دانش بخش دوم به صورت جدول(٤) است.

پایگاه دانش این بخش در اختیار قسمت تحلیل قرار می‌گیرد. الگوریتم(۳) شبه کد مرحله پایش را نشان می‌دهد. تابع مربوط به پایش در بازه‌های زمانی معین اجرا می‌شود تا به‌طور دقیق و مستمر بر فعالیت لایه‌ها و ورودی‌های سیستم نظارت کند.

Algorithm 3: Pseudo code for Monitoring Phase (Workload)

```
1: foreach Request in (Received Request to cloud in interval Δt) do
2: Req ←Monitor (Request);
3:User.id<-Req.UserID;
4: R ← Monitor (Response_Time of cloud Layers interval Δt);
5: C ← Monitor (Total Cost of cloud for Response to Request in interval Δt);
6:Rec[i,1] ← [ Req.id,Req] ;
7:Rec[i,2] ← [User.id,Req.id,R,C] ;
8: end for
9:return Rec;
```

٤-٢-٢ تحلیل (Analyze)

یکی از مهم‌ترین بخش‌های رویکرد پیشنهادی، مرحله تحلیل است. در قسمت تحلیل، ابتدا دو مورد مهم بررسی

24- Service Level Objective
25- SLA Violation

جدول ۳: ساختار رکوردهای پایگاه دانش بخش اول

درخواست	شناسه

جدول ۴: ساختار رکوردهای پایگاه دانش بخش دوم

هزینه	زمان پاسخگویی	شناسه درخواست	شناسه کاربر

می‌گردد. مورد اول پیش‌بینی بـــارکاری آتی و مورد دوم میزان تخطی از شـــرایط مورد قبول سرویس. در صورتی کـــه زمـــان پاســـخگویی از زمان مجـــاز پاســـخگویی به درخواست بیشتر باشد، محاسبه میزان تخطی از سرویس در مورد زمان پاســـخگویی با توجه به رابطه (۹) به‌دست می‌آید. به همین ترتیب در صورتی که هزینه پاسخگویی به درخواست از هزینه در نظر گرفته شده برای پاسخگویی به درخواســـت بیشتر باشد، رابطه (۱۰) نحوه محاسبه میزان تخطی از سرویس را در مورد هزینه نشان می‌دهد.

(۹)

$$SLA_Violation_{Response}=Response_{Time}-Deadline_{Time}$$

(۱۰)

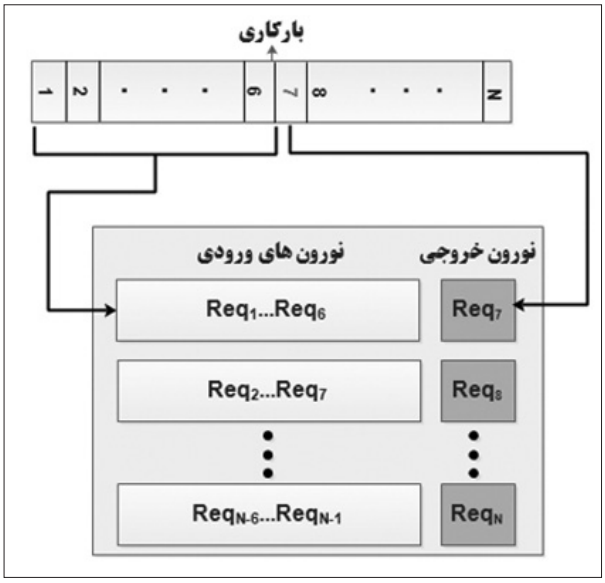
$$SLA_Violation_{Cost}=Cost_{Res}-Cost_{Req}$$

در رابطـــه (۹)، $Response_{Time}$ زمان پاســـخگویی به درخواســـت و $Deadline_{Time}$ زمان مجاز پاســـخگویی به درخواســـت اســـت. در رابطه (۱۰)، $Cost_{Req}$ هزینه در نظر گرفته شده ابتدایی برای درخواست و $Cost_{Res}$ هزینه نهایی پاسخگویی به درخواست است.

بـــه هرمیزان پیش‌بینـــی در این بخش دقیق‌تر باشـــد، تصمیم‌گیری نیز ، کیفیت بالاتری خواهد داشـــت. در روش پیشـــنهادی برای پیش‌بینی بارکاری از سیســـتم استنتاج عصبی- فازی تطبیق‌پذیر استفاده می‌شود. ورودی سیستم اســـتنتاج عصبی- فازی تطبیق‌پذیر یک سری زمانی است. پس در مرحله اول باید از بارکاری سری زمانی تهیه شود.

۴-۲-۲-۱ ساخت سری زمانی

ســـری زمانی در ســـاختار پیشنهادی متشـــکل از دو آرایه ورودی و هدف است. به‌طور مشخص آرایه ورودی وضعیت حـــال و آرایه هدف وضعیت آینـــده بارکاری را مشـــخص می‌کند. یکی از مهم‌ترین نکات در ساخت سری زمانی ســـاخت دقیـــق این دو آرایه اســـت. بـــه هر میزان



شکل ۵: ساخت سری زمانی

برش‌هـــای زمانی دقیق‌تری در مورد ایـــن دو آرایه انجام شـــود، دقت تشـــخیص و پیش‌بینی در مورد بارکاری بالا می‌رود. در صورتی که تعـــداد عناصر آرایه ورودی عدد k باشـــد، می‌توان تعداد عناصر آرایـــه خروجی را در حد $k/2$ در نظـــر گرفت. امـــا به هرمیزان تعـــداد لایه خروجی کمتر باشـــد دقت پیش‌بینی افزایش می‌یابد. نکته مهم دیگر در این زمینه این اســـت که عناصری که در یک بازه زمانی به‌عنوان خروجی اســـتفاده شـــده‌اند، در بازه بعدی زمانی به‌عنوان ورودی قرار می‌گیرد. این امر موجب افزایش دقت پیش‌بینی می‌شود. اما مشـــخص است که این روش تعداد داده مورد استفاده در سری زمانی را افزایش می‌دهد و به تناسب، اندکی سرعت پردازش را کاهش می‌دهد.

در روش پیشـــنهادی بـــه ازای هر ۶ واحـــد زمانی یک خروجی در نظر گرفته می‌شـــود و با توجه به توضیحات ارائه شـــده همین عمل در مـــورد بازه‌های بعدی نیز انجام می‌گیرد. برای ساخت ســـری زمانی به صورت شکل (۵) عمل می‌شود.

روش ساخت آرایه ورودی و هدف در سری زمانی به این صورت است که هر ۶ واحد بارکاری در آرایه ورودی ســـری زمانی قرار می‌گیـــرد و واحد کار بعـــدی به‌عنوان خروجی یا پیش‌بینی وضعیت بعد است. به همین ترتیب یک

شکل ۶: داده‌های آموزش، اعتبارسنجی و آزمایش

واحد در بارکاری پیش آمده و عمل قبلی تکرار می‌شـود. الگوریتم (٤) شبه‌کد ساخت سری زمانی را نشان می دهد.

Algorithm 4: Pseudo code for Time Series Creation

```
1: for  i=1 to N do
2:      for  j=1 to i+6 do
3:       Series..Input[i][j]=Request{j];
4:      end for
5:     Series..Output[i] = Request {i+7];
6:end for
7: return Series;
```

٤-٢-٢-٢ سـاخت داده های آموزش، اعتبارسنجی و آزمایش

در این بخش ۷۰ درصد از سری زمانی برای آموزش، ۱۵ درصد برای اعتبار سـنجی و ۱۵ درصـد دیگر برای آزمایـش نهایی اسـتفاده می‌شـود. شـکل(٦) سـاختار دسته‌بندی سری زمانی را نمایش می‌دهد.

شـبه‌کد سـاخت داده‌های آموزش، اعتبار سـنجی و آزمایـش در الگوریتم(۵) آمده اسـت. ورودی و خروجی سـری زمانی تمامی بخش‌ها باید سـاخته شود. برای ۷۰ درصد آموزش ابتدا ٦ واحد درخواسـت اول در ماتریس ورودی سـری زمانی قرارمی‌گیرند. سـپس درخواسـت بعدی(۷ام) در ماتریس خروجی سری زمانی آموزش قرار می‌گیـرد. همین عمل برای داده‌های اعتبارسـنجی و برای داده‌های آزمایشی انجام می‌شود.

٤-٢-٢-٣ ساختار نهایی شـبکه عصبی- فازی جهت پیش بینی

شکل(۷) ساختار نهایی شبکه عصبی- فازی پیشنهادی را نمایـش می‌دهد، که در آن خروجـی هر لحظه ورودی لحظه بعد خواهد بود.

Algorithm 5: Pseudo code for create_Teach_Test_Validation

```
1:  Teach_N=round(N * 0.7);
2: Valid_N=round(N * 0.15);
3: Test_N=round(N * 0.15);
4:  for  i=1 to Teach_N  do
5:   for  j=1 to i+6 do
6:   Teach_Series.Input[i][j]= Series.Input [i]{j];
7: end for
8:   Teach_Series.Output [i] = Series.Output [ i ];
9:end for
10:  for  i=Teach_N to Teach_N+Valid_N  do
11:   for  j=1 to i+6 do
12:   Validation_Series.Input [i][j]= Series.Input [i]{j];
13: end for
14:   Validation _Series.Output[i] = Series.Output [ i ];
15:end for
16:  for  i=Teach_N+Valid_N  to  Teach_N+Valid_N+Test_N do
17:   for  j=1 to i+6  do
18:   Test_ Series.Input [i][j]= Series.Input [i]{j];
19: end for
20:   Test_ Series.Output [i] = Series.Output [ i ];
21:end for
```

٤-٢-٢-٤ لایه ورودی شبکه عصبی

با توجه به این که بارکاری در روش پیشنهادی به ازای ٦ مقدار ورودی، یک مقدار خروجی وجود خواهد داشـت، تعداد نورون‌های لایه ورودی ٦ نورون خواهد بود.

٤-٢-٢-۵ لایه میانی شبکه عصبی

لایه میانی شبکه عصبی به تعداد برابر با شاخص‌های ورودی در نظر گرفته می‌شود.

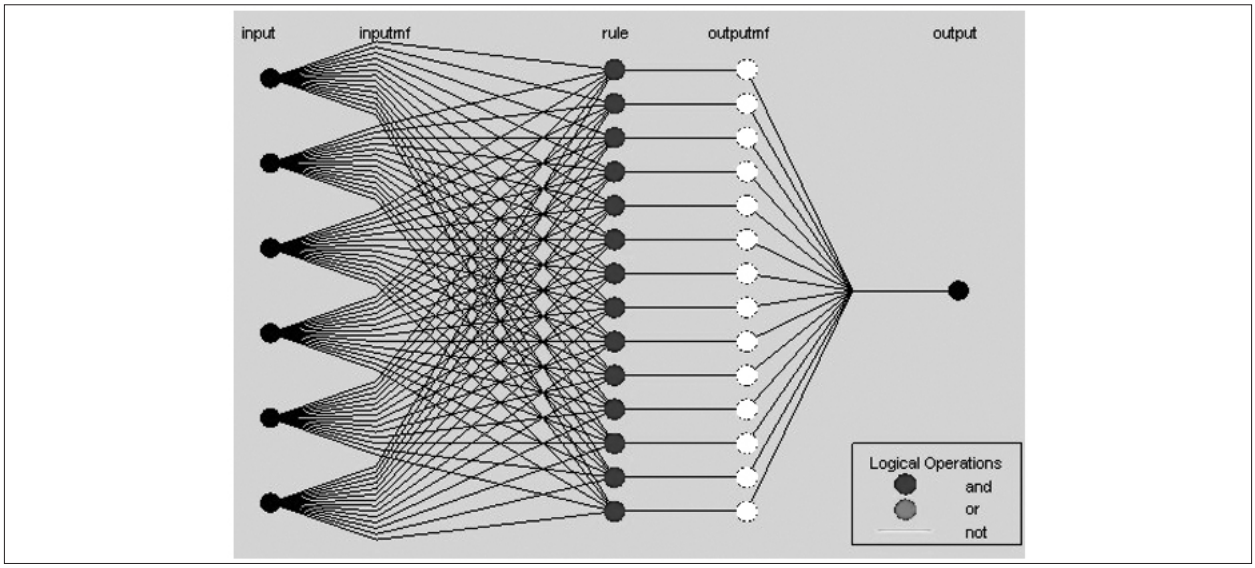
٤-٢-٢-٦ لایه خروجی شبکه عصبی

تعداد نورون‌های خروجی برابر با یک خواهد بود. دلیل ایـن امر پیش‌بینی وضعیت آتی بـارکاری در زمان بعدی اسـت که یک مقدار خواهد بود. به تناسب، تابع محرک لایه خروجی هم خطی خواهد بود.

٤-٢-٢-۷ تابع محرک

در این شـبکه از تابع سیگموید تک قطبی به‌عنوان تابع محرک اسـتفاده شده است. تابع سـیگموید تک قطبی (که نمودار آن S شـکل اسـت) متداول‌ترین فـرم تابع محرک در شـبکه‌های عصبی مصنوعی است. این تابع که به تابع یکنواخت افزایشـی نیز موسـوم اسـت، نمایش جالبی از توازن بین رفتار خطی و غیرخطی از خود نشـان می‌دهد. یک نمونه از توابع سیگموید «تابع لجستیک»[۲۶] است که در آن g «ضریب شیب»[۲۷] تابع سیگموید است. با تغییر پارامتر

26-Logistic Function
27-Slope Gain



شکل ۷: ساختار نهایی شبکه عصبی- فازی پیشنهادی جهت پیش‌بینی بار کاری

g تابع سیگموید با شیب‌های متفاوت به‌دست می‌آید.

هرچه g به سمت بی‌نهایت میل می‌کند تابع سیگموید

$$F\ (net) = \frac{1}{1 + e^{-g.net}} \qquad (۱۱)$$

به تابع حد آستانه تبدیل می‌شود. نکته مهم در مورد این تابع این است که این تابع مشتق‌پذیر است و این یک موضوع بسیار مهم و مؤثر در عملکرد شبکه‌های عصبی است. الگوریتم (٦) نحوهٔ عملکرد سیستم استنتاج عصبی- فازی تطبیق‌پذیر را مشخص می‌کند. ابتدا لایه‌های ورودی، میانی و خروجی تشکیل می‌گردد. بدین ترتیب که تعداد و مقادیر آرایه ورودی سری زمانی آموزشی به لایه ورودی سیستم استنتاج عصبی-فازی تطبیق‌پذیر منتسب می‌شود.

Algorithm 6: Pseudo code for ANFIS_predict

```
1:  Anfis_Net.InputLayer = count(Teach_Series.Input);
2: Anfis_Net.HiddenLayer = count (Teach_Series.Input) * 2;
3: Anfis_Net.OutputLayer = count(Teach_ Series.Output);
4: Anfis_Net.TeachData.Input=teach_ Series.Input;
5: Anfis_Net.TeachData.Output=teach_ Series.Output;
6: Anfis_Net.VaildData.Input=Vaild_ Series.Input;
7: Anfis_Net.VaildData.Output=Vaild_ Series.Output;
8: Anfis_Net.Teach();
9: Anfis_Net.TestData.Input=Test_ Series.Input;
10: Anfis_Net.TestData.Output=Test_ Series.Output;
11: Anfis_Net.Test();
12: Predict=ANFIS_Net.sim(WL);
13: return predict ;
```

٤-۲-۳ اجرا

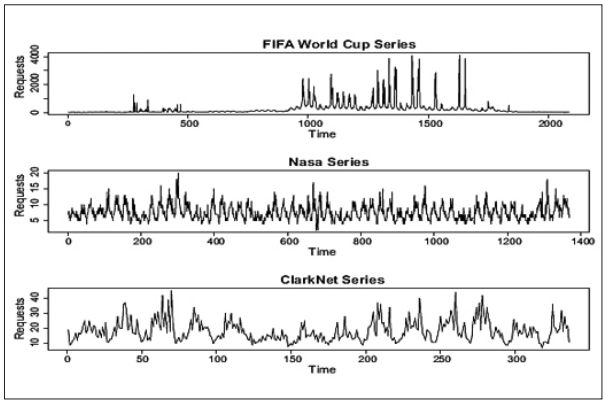
پس از تصمیم‌گیری بخش Execute وظیفه اجرای تصمیم اتخاذ شده را به عهده دارد. بر اساس نوع مقیاس‌بندی و میزان آن، Execute تغییرات را روی بُن‌سازهٔ ابر اعمال می‌کند، تا ساختار را بر اساس تصمیم مرحله تحلیل تنظیم کند. الگوریتم (۷) ساختار الگوریتم Execute را نشان می‌دهد. در این ساختار به ازای هر لایه فرمان نوع مقیاس‌بندی و میزان آن به فراهم‌کننده ابر ارسال می‌گردد، تا فراهم‌کننده ابر فرمان مورد نظر را برای اضافه یا کم‌کردن ماشین مجازی اجرا کند. در نهایت ماشین مجازی به درخواست اختصاص می‌یابد. با توجه به این که تا هنگامی که یک مقیاس‌بندی تمام نشده نباید مقیاس‌بندی جدید آغاز شود، هر نوع مقیاس‌بندی با ساختار FIFO توسط CIS انجام می‌شود، بدین معنی که مقیاس‌بندی‌ها بر حسب زمان ورود به صف CIS اجرا می‌شوند.

Algorithm 7: Pseudo code for Execution Phase (Scale)

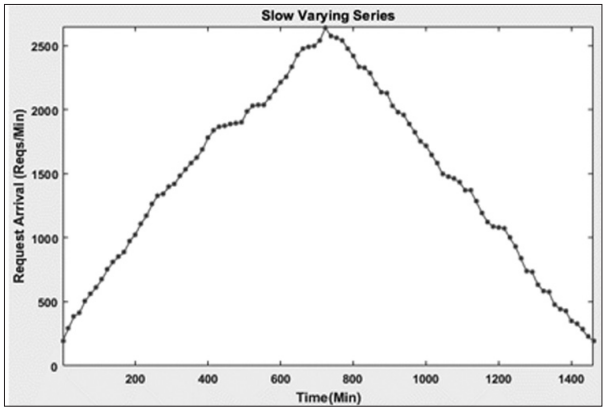
```
1: for i ← 1 to Scale.count
2: ADD Scale(i) to Queue CIS
3: end for
```

۵- ارزیابی کارایی

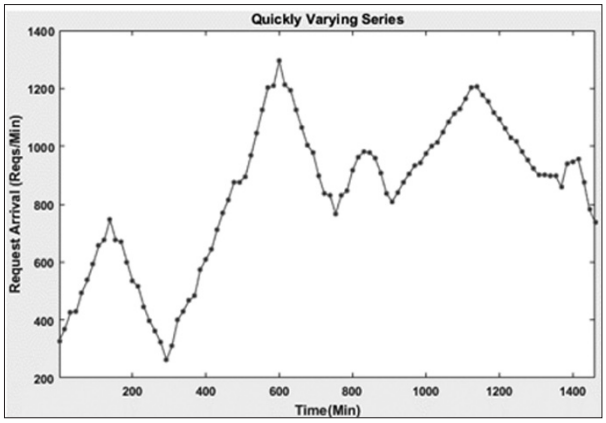
در این بخش سعی داریم رویکرد پیشنهادی را برا اساس دو معیار زمان پاسخگویی و میزان بهره‌وری ارزیابی کنیم. برای شبیه‌سازی رویکرد پیشنهادی، از



شکل ۸: نمایی از ساختار مجموعه داده واقعی



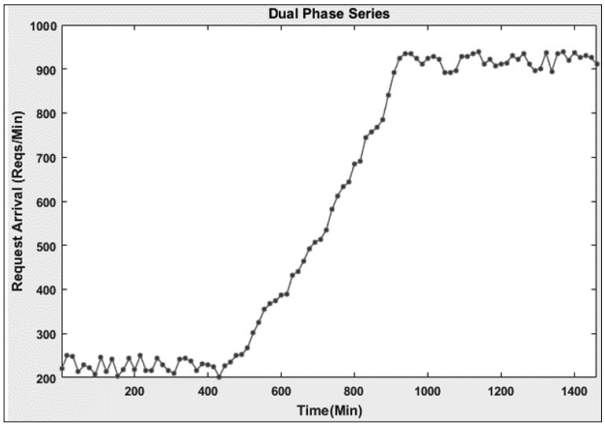
نمودار ۱: بارکاری مصنوعی متغیر کند



نمودار ۳: بارکاری مصنوعی با مرحلهٔ دو گانه

شبیه‌ساز کلودسیم، استفاده نموده‌ایم. درشبیه‌سازی روش پیشنهادی از دو نوع بارکاری واقعی و مجازی استفاده شده است. بارکاری واقعی که شامل سه مجموعه داده[۱۵] FIFA World Cup, NASA و ClarkNet می‌باشد همچنین سه نوع بارکاری مصنوعی شامل متغیر سریع، متغیرآهسته، بارکاری مرحلهٔ دوگانه.

برای بررسی و ارزیابی رویکرد پیشنهادی خود، آن را



نمودار ۳: بارکاری مصنوعی با مرحلهٔ دو گانه

با دو رویکرد ADRP_Fuzzy و ADRP_AL مقایسه نموده‌ایم.

رویکرد ADRP_Fuzzy [۱۶] همانند روش پیشنهادی روی ساختار معماری مشابه عمل می‌کند. این روش در قسمت پیش‌بینی از الگوریتم" ARIMA"[۲۸]و در مورد تامین منابع از روش منطق فازی استفاده می‌کند.

رویکرد الگوریتم ADRP_AL [۱۷] نیز همانند روش پیشنهادی روی ابر چندلایه و با مراحل مشابه عمل می‌کند. این روش با روش پیشنهادیمان در مرحله تحلیل متفاوت می‌باشد. این روش ابتدا در مرحلهٔ تحلیل از ساختار"ARMA "[۲۹] استفاده و در ادامه از روش اتوماتای آموزش‌پذیر استفاده می‌کند.

شاخص‌های ارزیابی رویکرد پیشنهادی عبارتند از: میانگین بهره‌وری، میانگین زمان پاسخگویی.

- میانگین بهره وری

میزان MIPSهای مورد نیاز تقسیم بر MIPSهای در دسترس را بهره‌وری می‌گویند.

$$Utilization = \frac{Allocated_MIPS}{Available_MIPS} \qquad (۱۲)$$

- زمان پاسخگویی

زمان پاسخ واقعی، تفاوت زمانی دقیق بین زمان درخواست کار و زمان تحویل کار انجام شده به‌کاربر می‌باشد.

جهت ارزیابی روش پیشنهای، ۲ سناریو با ساختار

28- Auto regressive integrated moving average
29- Auto-regressive Moving Average

جدول ۵: سناریوهای مورد ارزیابی

سناریو	بار کاری	هدف
سناریو اول	واقعی	مقایسه میانگین بهره وری
	مصنوعی	
سناریو دوم	واقعی	مقایسه زمان پاسخگویی
	مصنوعی	

جدول (۵) تشکیل شد. در شبیه‌سازی هر سناریو یک معیار مهم ارزیابی در سه الگوریتم ADRP_AN و ADRP_Fuzzy و ADRP_AL مورد بررسی قرار می‌گیرد.

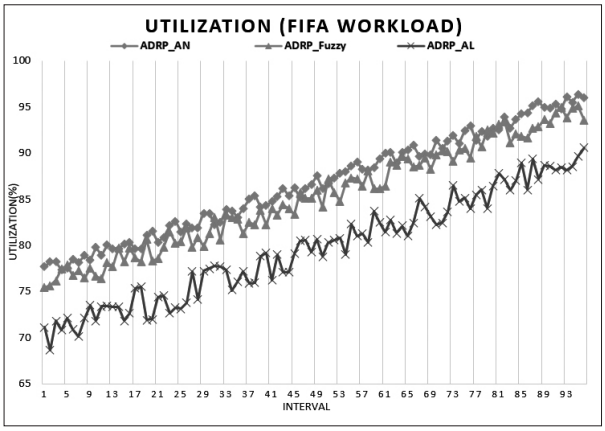
۵-۱ سناریو اول: میانگین بهره وری

یکی از مهم‌ترین شاخصه‌های ارزیابی، میانگین بهره‌وری در روش ارائه شده است. در این بخش میانگین بهره‌وری را تحت دو بارکاری واقعی و مصنوعی در سه الگوریتم ADRP_Fuzzy، ADRP_AN و ADRP_AL بررسی می‌کنیم.

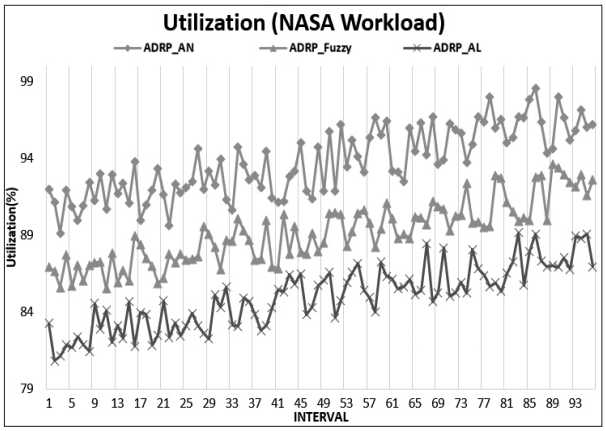
۵-۱-۱ ارزیابی میانگین بهره‌وری با استفاده از بارکاری واقعی

یکی از شاخص‌های مهم در مقایسه عملکرد الگوریتم‌های تامین منابع، میانگین بهره‌وری به ازای درخواست است. در این بخش بهره‌وری بررسی و با دو الگوریتم ADRP_Fuzzy و ADRP_AL مقایسه می‌شود. نمودارهای (٤، ٥ و ٦) میانگین بهره‌وری در طول شبیه‌سازی بارکاری NASA، FIFA و ClarkNet در سه الگوریتم هدف را نمایش می‌دهد. بارکاری FIFA به دلیل تعداد درخواست بالا در دقیقه و نوسانات زیاد آن، یکی از بهترین بارهای کاری برای مقایسه عملکرد روش‌های تامین منبع است. با توجه به نمودار (٤)، به دلیل پیش‌بینی مطلوب بارکاری آتی توسط سیستم استنتاج عصبی-فازی تطبیق‌پذیر، روش پیشنهادی کارآیی بهتری در مورد تامین منبع دارد. استفاده از ساختار ارائه شده برای این روش موجب می‌شود تامین منبع به بهترین نحو انجام شود.

به‌طور مشخص به هر میزان تامین منابع دقیق تر انجام شود منابع یا ماشین‌های مجازی دقیق‌ترین حالت برای



نمودار ۴: میانگین بهره وری در طول شبیه‌سازی ۲۴ ساعت کاری سری FIFA در سه الگوریتم



نمودار ۵: میانگین بهره وری در طول شبیه‌سازی ۲۴ ساعت کاری سری NASA در سه الگوریتم

پاسخگویی به درخواست‌ها را خواهند داشت. به همین دلیل نسبت تعداد MIPSهای درخواست شده به MIPSهای موجود در ابر به یک نزدیک می‌شوند. با توجه به نتایج، عملکرد بهتری نسبت به دو روش دیگر داریم.

با توجه به نتایج در بارکاری ناسا و کلارک روش پیشنهادی عملکرد بهتری نسبت به تصمیم‌گیر فازی و اتوماتای یادگیر دارد.

۵-۱-۲ میانگین بهره وری با استفاده از بارکاری مصنوعی

نمودار (۷، ۸ و۹) میانگین بهره‌وری در بارکاری متغیر سریع، متغیر آهسته و بارکاری مرحلهٔ دوگانه را نمایش می‌دهد. با توجه به نمودار (۷)، نظارت دقیق برنامه اصلی عامل بسیار موثری برای کنترل وضعیت منابع است. تامین صحیح منابع موجب ارائه دقیق خدمات به درخواست‌ها

می‌شـــود. به‌طور مشـــخص به دلیل تامیـــن صحیح منابع پردازشـــگر با بهترین وضعیت در اختیار درخواســـت‌ها قرار می‌گیرد. البته ممکن اســـت در تعداد درخواســـت بالا میزان بهره‌وری به آســـتانه یک برســـد که دلیل آن تعداد درخواســـت‌های بالا اســـت که نیاز بـــه در اختیار گرفتن پردازنده دارند.

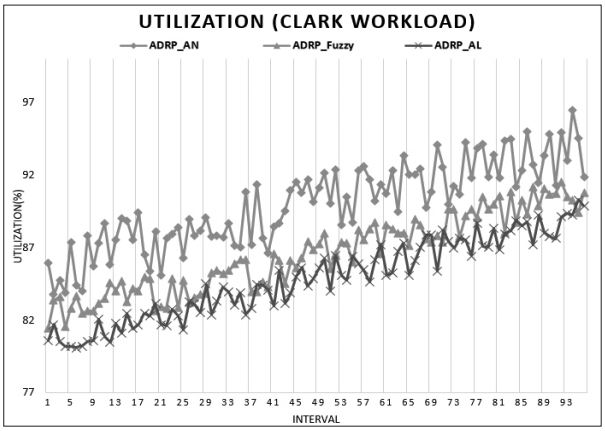
باتوجه به نتایج عملکـــرد الگوریتم ADRP_AN، کیفیت بهره‌وری در این روش در هر ســـه بارکاری نسبت به دو روش بالاتر است.

۵-۲ سناریو دوم: میانگین زمان پاسخگویی

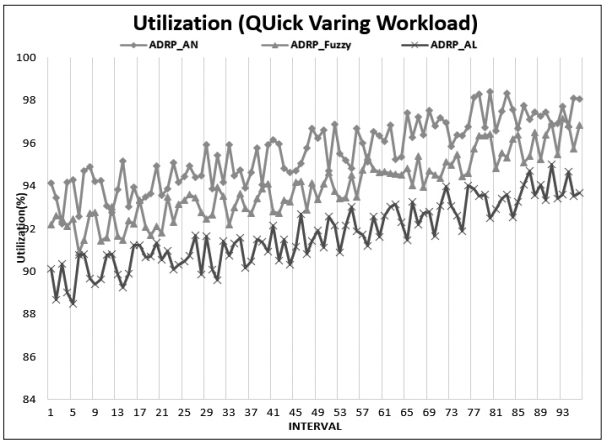
زمان پاســـخگویی به‌عنوان یکـــی از موثرترین اهداف شرایط ســـرویس یا SLA، نقش موثری در انتخاب ماشین مجازی دارد. در صورتی که زمان پاســـخگویی مورد نظر درخواست که با نام حد مجاز زمان پاسخگویی شناسایی می‌شود حاصل نشود، باید مقیاس‌بندی انجام شود. در این بخش زمان پاسخگویی را در بارکاری واقعی و مصنوعی در ســـه الگوریتم ADRP_Fuzzy، ADRP_AN و ADRP_AL بررسی می‌کنیم. برای نمایش میانگین زمان پاسخگویی از ده بازه زمانی استفاده شده است.

۵-۲-۱ ارزیابی زمان پاسخگویی با استفاده از بارکاری واقعی

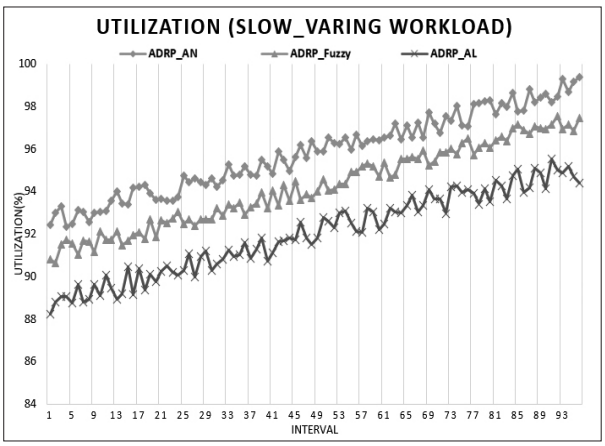
در این بخش زمان پاسخگویی در بارکاری FIFA،NASA و ClarkNet بررســـی می‌شـــود و با دو الگوریتم ADRP_ Fuzzy و ADRP_AL مقایســـه می‌شـــود. نمودار (۱۰، ۱۱ و ۱۲) میانگیـــن زمان پاســـخگویی در طول شبیه‌ســـازی بارکاری NASA، FIFA و ClarkNet در سه الگوریتم هدف را نمایش می‌دهد. اســـتفاده از پیش‌بینی مناسب بارکاری و محاســـبه تخطی از شرایط ســـرویس موجب می‌شود، کمترین حالـــت کمبود منبع پیش آید. این امر موجب تامین بهینه منابع برای درخواست‌ها شده و به تناسب اختصاص منبع دقیق‌تری صـــورت گرفته و نهایتا باعث کاهش زمان پاسخگویی و افزایش کارایی می‌شود.



نمودار ۶: میانگین بهره وری در طول شبیه‌سازی ۲۴ ساعت کاری سری ClarkNet در سه الگوریتم

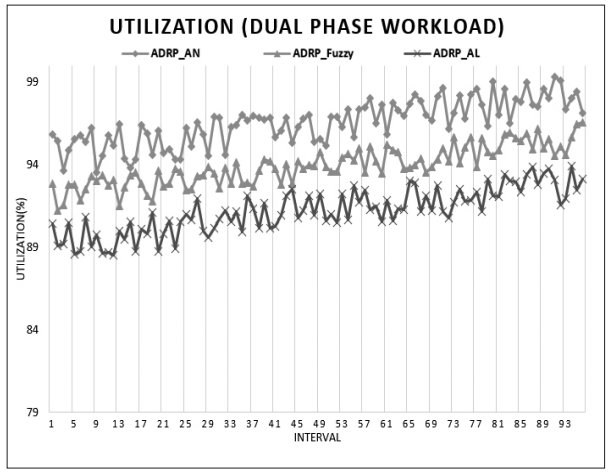


نمودار ۷: میانگین بهره وری در طول شبیه‌سازی ۲۴ ساعت کاری سری متغیر سریع در سه الگوریتم

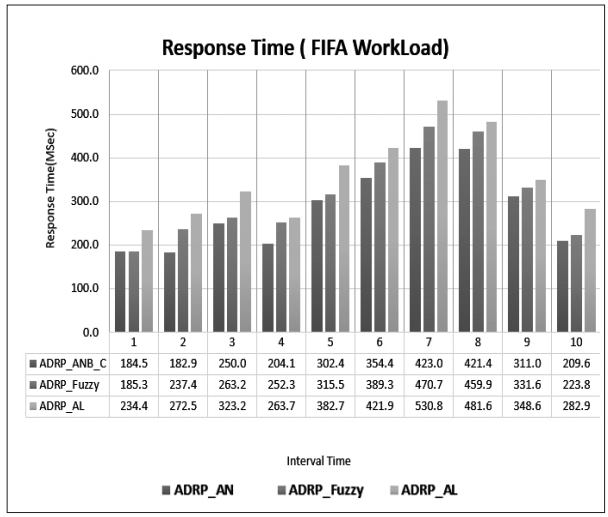


نمودار ۸: میانگین بهره وری در طول شبیه‌سازی ۲۴ ساعت کاری سری متغیر آهسته در سه الگوریتم

به دلیل عملکرد مطلوب سیستم استنتاج عصبی-فازی تطبیق‌پذیر برای پیش‌بینی بارکاری، عمل تعیین مقیاس‌بندی به درستی انجام می‌شـــود، که این امر موجب تامین بهینه

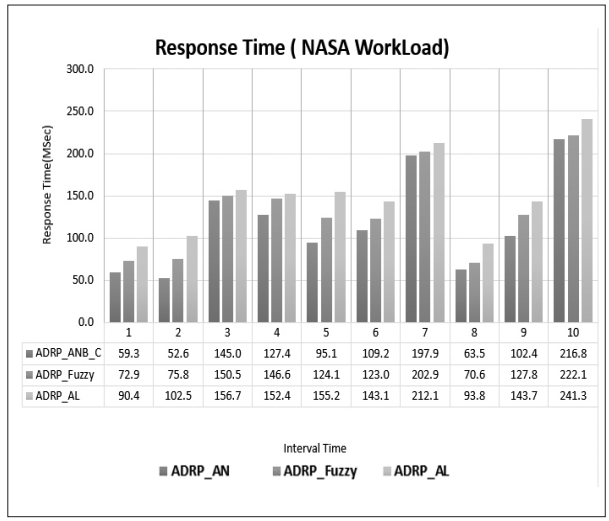


نمودار ۹: میانگین بهره وری در طول شبیه‌سازی ۲۴ ساعت کاری سری مرحلهٔ دوگانه در سه الگوریتم



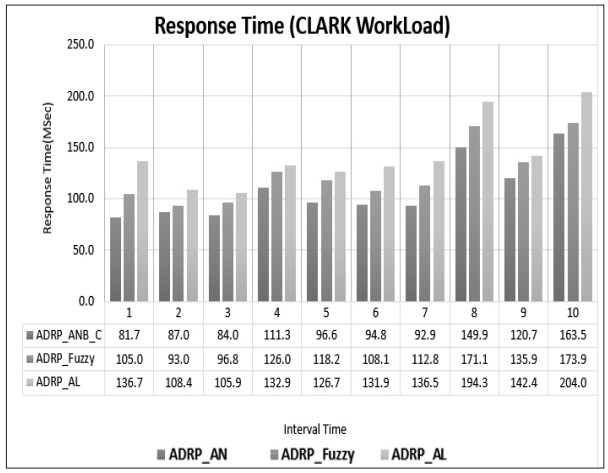
	1	2	3	4	5	6	7	8	9	10
ADRP_ANB_C	184.5	182.9	250.0	204.1	302.4	354.4	423.0	421.4	311.0	209.6
ADRP_Fuzzy	185.3	237.4	263.2	252.3	315.5	389.3	470.7	459.9	331.6	223.8
ADRP_AL	234.4	272.5	323.2	263.7	382.7	421.9	530.8	481.6	348.6	282.9

نمودار ۱۰: میانگین زمان پاسخگویی در طول شبیه‌سازی ۲۴ ساعت کاری سری FIFA در سه الگوریتم



	1	2	3	4	5	6	7	8	9	10
ADRP_ANB_C	59.3	52.6	145.0	127.4	95.1	109.2	197.9	63.5	102.4	216.8
ADRP_Fuzzy	72.9	75.8	150.5	146.6	124.1	123.0	202.9	70.6	127.8	222.1
ADRP_AL	90.4	102.5	156.7	152.4	155.2	143.1	212.1	93.8	143.7	241.3

نمودار ۱۱: میانگین زمان پاسخگویی در طول شبیه‌سازی ۲۴ ساعت کاری سری NASA در سه الگوریتم



	1	2	3	4	5	6	7	8	9	10
ADRP_ANB_C	81.7	87.0	84.0	111.3	96.6	94.8	92.9	149.9	120.7	163.5
ADRP_Fuzzy	105.0	93.0	96.8	126.0	118.2	108.1	112.8	171.1	135.9	173.9
ADRP_AL	136.7	108.4	105.9	132.9	126.7	131.9	136.5	194.3	142.4	204.0

نمودار ۱۲: میانگین زمان پاسخگویی در طول شبیه‌سازی ۲۴ ساعت کاری سری ClarkNet در سه الگوریتم

منابع برای درخواست‌ها شده و به تناسب اختصاص منبع دقیق‌تری صورت گرفته و نهایتا زمان پاسخگویی کاهش یافته که خود موجب افزایش سرعت پاسخگویی می‌شود. با توجه به نتایج، روش پیشنهادی کارآیی بهتری در مورد زمان پاسخگویی دارد.

۵-۲-۲ ارزیابی زمان پاسخگویی با استفاده از بارکاری مصنوعی

در این بخش زمان پاسخگویی الگوریتم پیشنهادی در بارکاری متغیر سریع، متغیر آهسته و بارکاری مرحلهٔ دوگانه بررسی می‌شود و با دو الگوریتم ADRP_Fuzzy و ADRP_AL مقایسه می‌شود. نمودارهای (۱۳، ۱۴و ۱۵) میانگین زمان پاسخگویی در طول شبیه‌سازی بارکاری متغیر سریع، متغیر آهسته و بارکاری مرحلهٔ دوگانه در سه الگوریتم هدف را نمایش می‌دهد. استفاده از پیش‌بینی مناسب بارکاری و محاسبه تخطی از شرایط سرویس موجب می‌شود، کمترین حالت کمبود منبع پیش آید. این امر موجب تامین بهینه منابع برای درخواست‌ها شده و به تناسب اختصاص منبع دقیق‌تری صورت گرفته و نهایتا باعث کاهش زمان پاسخگویی و افزایش کارایی می‌شود.

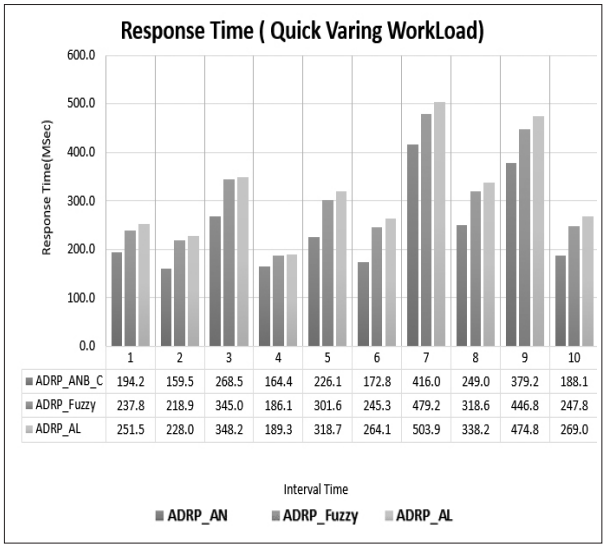
۵-۳ ارزیابی نهایی

در این بخش میانگین نتایج حاصل شده در بخش‌های قبلی را نمایش داده‌ایم. در نمودار(۱۶) میانگین بهره‌وری،

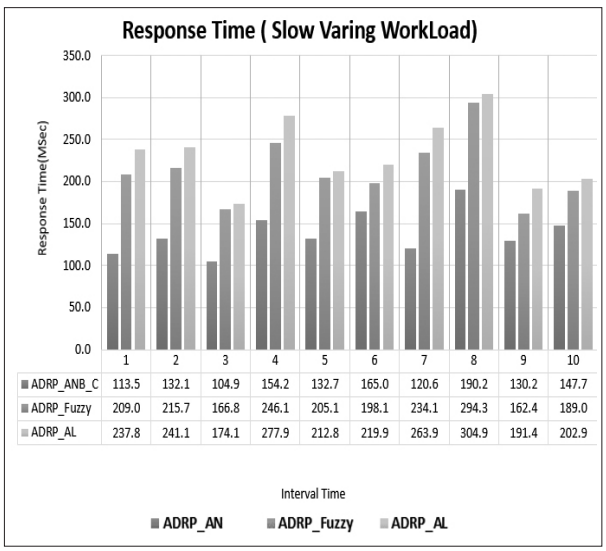
نمودار(۱۷) میانگین زمان پاسخگویی در سه الگوریتم ADRP_Fuzzy، ADRP_AN و ADRP_AL مشخص شده است.

۶- نتیجه‌گیری و پیشنهادها

در این مقاله، رویکردی بهبود یافته برای تامین منابع برنامه‌های کاربردی چندلایه با استفاده از سیستم استنتاج عصبی- فازی تطبیق‌پذیر در محیط‌های ابری ارائه کردیم. در رویکرد ارائه شده از ساختار معماری سه لایه استفاده کرده‌ایم که این معماری دارای سه لایه کارخواه یا وب، لایه کاربرد و لایه بانک اطلاعاتی است. لایه وب وظیفه دریافت درخواست، ساخت نشست برای کاربر، احراز هویت اولیه و شناخت بدافزار را به عهده دارد. لایه کاربرد وظیفه اجرای برنامه‌های کاربردی و ارائه خدمات به‌کاربر را به عهده دارد. اصلی‌ترین لایه در معماری سه‌لایه، لایه کاربرد است که وظیفه ارتباط دولایه دیگر را به عهده دارد. لایه سوم، لایه بانک اطلاعاتی است. این لایه وظیفه مدیریت داده‌ها را به عهده دارد. در این ساختار، هر ماشین مجازی در یکی از این لایه ها قرار می‌گیرد. البته ممکن است ماشین مجازی (VM) با پیکربندی مجدد از لایه‌ای به لایه دیگر مهاجرت کند. در ساختار ابر CIS وظیفه انتقال درخواست‌ها را به لایه اول به عهده دارد. تنظیم‌کننده بارکاری در هر لایه، در صورتی که در خواست های ورودی به لایه از حد سرویس بالاتر باشد، صفی از درخواست ایجاد می‌کند که به واسطه آن پس از آزاد شدن هر ماشین مجازی درخواستی را در اختیار آن قرار می‌دهد. CIS نظارت بر تنظیم‌کننده بارکاری را به عهده دارد. بدین معنی که اطلاعات را در زمان‌های معینی از تنظیم‌کننده بارکاری هر لایه گرفته و نگهداری می‌کند. زمان پاسخگویی درخواست‌ها، هزینه درخواست‌ها و میزان بارکاری به‌عنوان کارآیی سیستم در هر مرحله توسط ساختار پایش از CIS درخواست و در نهایت برداشت می‌شود. داده‌های برداشت شده توسط پایش در اختیار تحلیل قرار داده می‌شود. وظیفه تحلیل

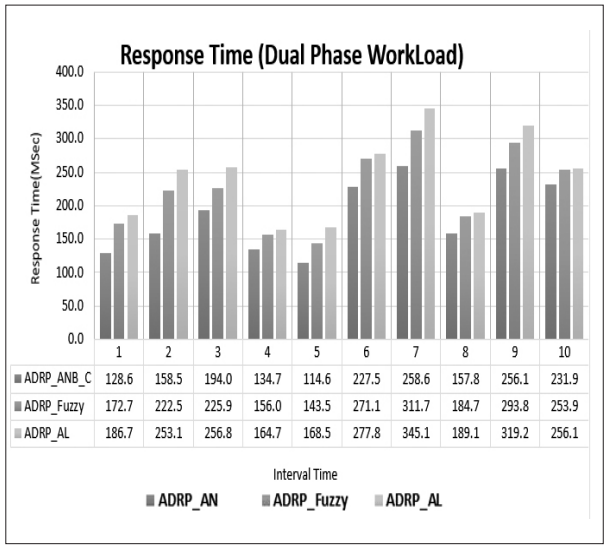


نمودار ۱۳: میانگین زمان پاسخگویی در طول شبیه‌سازی ۲۴ ساعت کاری سری متغیر سریع در سه الگوریتم

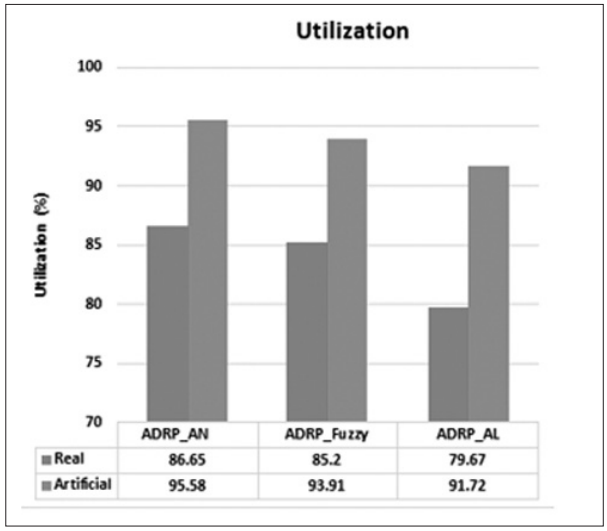


نمودار ۱۴: میانگین زمان پاسخگویی در طول شبیه‌سازی ۲۴ ساعت کاری سری متغیر آهسته در سه الگوریتم

پیش‌بینی وضعیت آتی سیستم بر اساس وضعیت فعلی است. عملا تحلیل علاوه بر تشخیص میزان تخطی از SLA، میزان بار کاری بعدی را نیز با استفاده از سیستم استنتاج عصبی-فازی تطبیق‌پذیر پیش‌بینی می‌کند. سپس بر اساس پیش‌بینی انجام شده و میزان تخطی از SLA اعلام شده، تصمیم می‌گیرد که چه تعداد ماشین‌مجازی جدید اضافه یا حذف کند (مقیاس‌بندی افقی) یا چه تعداد ماشین‌مجازی قبل را مجدد پیکربندی (مقیاس‌بندی عمودی) کند. تصمیم‌گیری مرحله تحلیل برای بخش اجرا ارسال می‌شود تا این تصمیم



نمودار ۱۵: میانگین زمان پاسخگویی در طول شبیه‌سازی ۲۴ ساعت کاری سری مرحلهٔ دوگانه در سه الگوریتم



نمودار ۱۶: میانگین بهره وری کلی

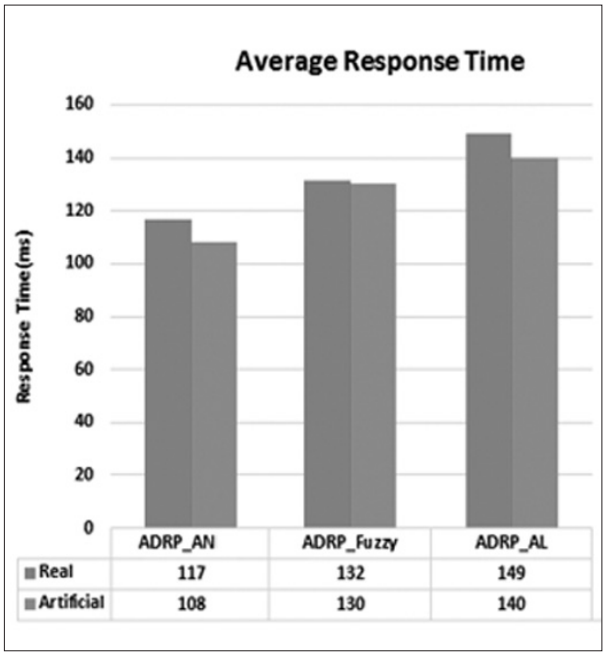
اجرایی شود.

روش پیشنهادی بر روی دو نوع داده واقعی و مصنوعی آزمایش و ارزیابی شـــد. نتایج عملکرد روش پیشنهادی به صورت زیر می‌باشد:

افزایش بهره‌وری به میزان ۱,۹ درصد و کاهش زمان پاسخگویی ۲,۳ درصد.

بـــا توجه به نتایـــج به‌دســـت‌آمده از مقالـــه حاضر و برای ارزیابی بیشـــتر نتایج و تکمیل و توسعه این تحقیق، پیشنهادهای ذیل ارائه می‌گردد:

– اســـتفاده از دیگـــر الگوریتم‌های خوشـــه‌بندی مانند



نمودار ۱۷: میانگین زمان پاسخگویی کلی

K–mean، C–mean یا Fuzzy k–mean برای شـــناخت قوی‌تـــر و بهتر الگوی بارکاری که موجـــب تامین منبع با کیفیت بالاتر می‌شود.

منابع:

[1]Buyya, Rajkumar, James Broberg, and Andrzej M. Goscinski, eds. «Cloud computing: Principles and paradigms”. Vol. 87. John Wiley & Sons, 2111.

[2]Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. «Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility.” Future Generation computer systems 25, no. 6 (2119): 599-616.

[3]Lorido-Botran, T., Miguel-Alonso, J., & Lozano, J. A. (2014).” A review of autoscaling techniques for elastic applications in cloud environments”. Journal of Grid Computing, 12(4), 559-592.

[4]Huang, D., He, B. and Miao, C., 2014. “A Survey of Resource Management in Multi-Tier Web Applications”. IEEE Communications Surveys & Tutorials, 16(3), pp.1574-1590.

[5]RahimiZadeh, K., AnaLoui, M., Kabiri, P. and Javadi, B., 2015. “Performance Modeling and Analysis of Virtualized Multi-Tier Applications Under Dynamic Workloads”. Journal of Network and

ing approach for multi-tier applications in cloud computing". Frontiers of Computer Science, 7(4), pp.459-474.
Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P. and Wood, T., 2008. "Agile dynamic provisioning of multi-tier internet applications". ACM Transactions on Autonomous and Adaptive Systems (TAAS), 3(1), p.1.
[14]Jang, J.S., 1993. "ANFIS: adaptive-network-based fuzzy inference system". IEEE transactions on systems, man, and cybernetics, 23(3), pp.665-685.
[15]Messias, Valter Rogério, Julio Cezar Estrella, Ricardo Ehlers, Marcos José Santana, Regina Carlucci Santana, and Stephan Reiff-Marganiec. «Combining time series prediction models using genetic algorithm to autoscaling web applications hosted in the cloud infrastructure." Neural Computing and Applications vol. 27, no. 8, pp. 2383-2406, 2016.
[16]Khorsand, R., Ghobaei-Arani, M. and Ramezanpour, M., "FAHP approach for autonomic resource provisioning of multitier applications in cloud computing environments". Software: Practice and Experience, 2018.
[17]Ghobaei-Arani, Mostafa, Sam Jabbehdari, and Mohammad Ali Pourmina. «An autonomic approach for resource provisioning of cloud services." Cluster Computing vol 19, no. 3, pp.1017-1036, 2016.
Computer Applications, 56, pp.166-187.
[6]Beltrán, M., 2015. "Automatic Provisioning of Multi-Tier Applications in Cloud Computing Environments". The Journal of Supercomputing, 71(6), pp.2221-2250.
[7]Eawna, M.H., Mohammed, S.H. and El-Horbaty, E.S.M., 2015. "Hybrid Algorithm for Resource Provisioning of Multi-Tier Cloud Computing". Procedia Computer Science, 65, pp.682-690.
[8]Singh, S. and Chana, I., 2015. "Q-Aware: Quality of Service Based Cloud Resource Provisioning". Computers & Electrical Engineering, 47, pp.138-160.
[9]Iqbal, W., Dailey, M.N., Carrera, D. and Janecek, P., 2011. "Adaptive resource provisioning for read intensive multi-tier applications in the cloud". Future Generation Computer Systems, 27(6), pp.871-879.
[10]Muppala, S., Chen, G. and Zhou, X., 2014. "Multi-tier service differentiation by coordinated learning-based resource provisioning and admission control". Journal of Parallel and Distributed Computing, 74(5), pp.2351-2364.
[11]Bi, J., Yuan, H., Tie, M. and Tan, W., 2015. "SLA-based optimisation of virtualised resource for multi-tier web applications in cloud data centres". Enterprise Information Systems, 9(7), pp.743-767.
[12]Wu, H., Zhang, W., Zhang, J., Wei, J. and Huang, T., 2013. "A benefit-aware on-demand provision-