دوره هجدهم، شماره تابستان ۱۴۰۴

مجله فناوری اطلاعات در طراحی مهندسی

Information Technology in Engineering Design

http://sanad.iau.ir/journal/ited

تخصیص بهینه منابع در فضای ابر– مه به کمک ساختار DDQN سلسله مراتبی مبتنی بر الگوریتم PSO

سید دانیال علیزاده جواهری[۱] رضا قائمی*[۲] حسین منشی زاده نائین[۳]

(۱) گروه مهندسی کامپیوتر، واحد نیشابور، دانشگاه آزاد اسلامی، نیشابور، ایران

(۲) گروه مهندسی کامپیوتر، واحد قوچان، دانشگاه آزاد اسلامی، قوچان، ایران*

(۳) گروه مهندسی کامپیوتر، واحد نیشابور، دانشگاه آزاد اسلامی، نیشابور، ایران

(تاریخ دریافت: ۱۴۰۳/۱۲/۰۶ تاریخ پذیرش: ۱۴۰۴/۰۲/۲۷)

چکیده

فناوری اینترنت اشیا (IoT) به سرعت در حال گسترش به عرصه‌های مختلف، از جمله مدیریت ترافیک و نظارت بر سلامت افراد است. با پیشرفت این فناوری، وابستگی ما به داده‌ها و اطلاعات جمع‌آوری‌شده از حسگرها و دستگاه‌های متصل به شدت افزایش یافته است. در این شبکه‌ها، کاربران با حجم بالایی از درخواست‌ها مواجه‌اند و نیاز به پردازش سریع و مؤثر این داده‌ها به یک ضرورت اساسی تبدیل شده است. هرگونه تأخیر در این فرآیند می‌تواند پیامدهای منفی عمیقی بر کارایی و کیفیت سیستم داشته باشد.در این راستا، استفاده از فضای ابری برای مدیریت این حجم از درخواست‌ها با چالش‌های جدی همراه است، به‌ویژه در مورد درخواست‌هایی که حساس به تأخیر هستند. در چنین مواردی، کاهش زمان پاسخ‌دهی ضروری است و برای این منظور، وجود زیرساخت‌های قوی و مؤثر الزامی است. یکی از راهکارهای کارآمد در این زمینه، بهره‌گیری از فضای مه و منابع موجود در سمت کاربران است. این رویکرد به کاربران این امکان را می‌دهد که داده‌ها را در منابع نزدیک تر به خود پردازش کنند؛ در نتیجه، تأخیر کاهش یافته و سرعت پاسخ‌دهی افزایش می‌یابد. با این حال، گره‌های مه دارای ظرفیت‌های محدودی هستند و به همین دلیل، مدیریت بهینه درخواست‌ها باید به گونه‌ای انجام گیرد که با حداقل تأخیر و حداکثر کارایی، وظایف محوله به درستی انجام شود. در این مقاله، از الگوریتم یادگیری تقویتی عمیق مبتنی بر شبکه Q دوگانه استفاده شده و در هر مرحله، هایپرپارامترهای آن با کمک الگوریتم ازدحام ذرات به روزرسانی می‌شوند. برای این منظور، از مجموعه داده‌ای که در سال ۲۰۱۹ توسط شرکت گوگل معرفی شده است، بهره‌برداری شده است. نتایج به‌دست‌آمده نشان‌دهنده این است که میانگین کاهش مقدار تابع خطا به میزان ۰,۰۰۰۵ در هر مرحله، افزایش نرخ تکمیل پردازش درخواست‌ها، ایجاد ثبات در مصرف انرژی و کاهش نرخ کاوش، بیانگر کارآیی مطلوب رویکرد پیشنهادی می‌باشد. این یافته‌ها تأکید می‌کنند که استفاده از الگوریتم‌های پیشرفته می‌تواند در بهینه‌سازی عملکرد شبکه‌های اینترنت اشیا نقش بسزایی داشته باشد.

کلمات کلیدی: تخصیص‌دهی منابع، یادگیری تقویتی، شبکه q دوگانه، الگوریتم ازدحام ذرات، فضای ابری– مه.

*عهده‌دار مکاتبات:

رضا قائمی

نشانی: گروه مهندسی کامپیوتر، واحد قوچان، دانشگاه آزاد اسلامی، قوچان، ایران

پست الکترونیکی: rezaghaemi@iau.ac.ir

۱- مقدمه

منابع ابری با این که از ظرفیت‌های محاسباتی و مقدار حافظه قابل توجه‌ای برخوردار هستند، ولی استفاده از آنها دارای هزینه نسبتا بالایی بوده و از سویی دیگر در بعضی از شرایط، در دسترس مشترکین قرار نداشته و برای کاربردهای حساس به زمان گزینه مناسبی نیستند. لذا در این شرایط، از فضای مه برای پردازش درخواست‌های حساس به زمان و قابل اجرا بر روی ماشین‌های مجازی قرارگرفته در لبه شبکه می‌توان بهره گرفت [2-1]. به بیان دیگر، با ظهور تکنولوژی اینترنت اشیا (IOT) [1]، دیگر ساختارهای سنتی همچون استفاده از فضای ابری قادر به تامین نیازهای مشترکین به صورت مناسب و با سرعت بالا نبوده زیرا با چالش‌های متعددی از جمله محدودیت پهنای باند، تاخیرهای طولانی در سرویس‌دهی، کمبود ظرفیت حافظه و سیستم پردازشی روبرو شده‌اند. در این شرایط بخاطر حجم زیاد داده‌های مورد استفاده، مدیریت تبادل اطلاعات و نحوه پردازش آنها از اهمیت زیادی برخوردار است. از سویی دیگر، فضای مه ساختار جدیدی در شبکه‌های کامپیوتری است که امکان استفاده از ظرفیت‌های قرارگرفته در لبه را برای پردازش فعالیت‌ها فراهم می‌آورد. تجهیزاتی که در این فضا مورد استفاده قرار گرفته به نام گره‌های مه (FN)[2] شناخته می شوند. در چنین ساختاری، ظرفیت‌های پردازشی کاربران از تعدادی ماشین فیزیکی (PM)[3] تشکیل شده که هر ماشین فیزیکی خود می‌تواند دارای یک یا چند ماشین مجازی (VM)[4] نیز باشد. بر اساس محل ارسال وظایف درخواست شده، اولویت، مقدار ظرفیت پردازشی و میزان حافظه مورد نیاز، این وظائف به ماشین‌های مجازی در شبکه IoT تخصیص‌دهی می‌شوند. در سیستم مورد نظر، درخواست‌ها از طریق شبکه IoT به گره‌های مه ارسال و آنها وظیفه کنترل، مدیریت و ارسال/دریافت داده‌ها را به ماشین‌های مجازی بر عهده دارند.

در زمان تخصیص دهی منابع در فضای ابری چالش های متعددی نمایان می‌گردد که از آن جمله می‌توان به عدم امکان استفاده از فضاهای ابری بدلیل حجم بالای درخواست‌ها، تغییرات پیوسته منابع موجود در گره‌های مه، لزوم پاسخ‌دهی سریع به درخواست‌های حساس به تأخیر و تنوع بالای وظائف مطرح شده اشاره کرد. از سویی دیگر با اینکه رویکردهای متعددی تاکنون برای تخصیص‌دهی فعالیت‌ها ارائه شده اما یا بر روی یک مدل خاصی از وظائف تمرکز داشته یا اهداف درنظرگرفته شده برای آنها تنها کاهش مدت زمان پاسخگویی یا حداقل‌سازی میزان انرژی مصرفی درنظرگرفته شده و یا اینکه فقط از دیدگاه بهره‌بردار به مساله تخصیص‌دهی پرداخته شده است [4-3]. لذا تخصیص

[1] Internet of Things (IoT)
[2] Fog Node (FN)
[3] Physical Machine (PM)
[4] Virtual Machine (VM)

وظائف برای یک مجموعه‌ای از وظائف متعدد و بادرنظرگرفتن اهداف متعددی همچون حداقل‌سازی بازه زمانی اتمام فعالیت و خطای تخصیص‌دهی، سطح پوشش و تکمیل درخواست‌های حداکثری و ارتقای سطح کیفیت سرویس‌دهی در بستر اینترنت اشیا هنوز به طور جامعی تحت مطالعه قرار نگرفته است.

در این مقاله از یک رویکرد نوینی برای غلبه بر چالش‌های مطرح شده بهره گرفته شده است. روش پیشنهادی قادر خواهد بود به صورت خودکار و بر اساس تجربیات کسب شده قبلی خود، به یک استراتژی مؤثر جهت تخصیص‌دهی دست یابد. بدین منظور از روش یادگیری تقویتی عمیق (DRL)[1] به کمک شبکه q دوگانه (DDQN) جهت تخصیص وظائف استفاده شده است. از آنجایی که درخواست‌ها به صورت سلسله وار به سیستم ارسال می‌گردند، لذا تخصیص‌دهی آن‌ها بایستی در بازه زمانی بسیار کوتاه صورت پذیرد تا از بروز صف طولانی برای فعالیت‌های در انتظار پردازش جلوگیری گردد. از سویی دیگر، طیف درخواست‌های ارسال شده دارای تنوع بالایی بوده و این ناهمگن بودن در برنامه‌ریزی‌ها نیز لازم است مدنظر قرار گیرد. به همین منظور از الگوریتم ازدحام ذرات (PSO) برای به روزرسانی هایپرپارامترهای شبکه q دوگانه بهره گرفته شده است. تا ساختار مورد استفاده به صورت مداوم و با توجه به اطلاعات گذشته سیستم به روزرسانی گردد.

در ادامه مهم‌ترین دستاوردهای مقاله را می‌توان به صورت زیر نام برد:

- بکارگیری الگوریتم یادگیری تقویتی دوگانه (DDQN) به منظور تخصیص ماشین‌های مجازی در دسترس برای پردازش درخواست‌های دریافت شده
- بروزرسانی هایپرپارامترهای DDQN به کمک بکارگیری الگوریتم ازدحام ذرات (PSO) به منظور بهبود نرخ همگرایی
- آموزش عامل از طریق تجربیات گذشته و به روز رسانی خط‌مشی‌های پیش رو در جهت بهینه سازی عملکرد
- بررسی عملکرد عامل به صورت همزمان در چندین اپیزود جهت کاهش نوسانات نتایج خروجی
- ارزیابی همزمان معیارهایی همچون تابع خطا (Loss)، نرخ کاوش (Epsilon)، میرایی نرخ کاوش (Epsilon decay)، نرخ مصرف انرژی، نرخ تکمیل پردازش درخواست‌های دریافت شده، میانگین مدت زمان پاسخ‌دهی (Makespan) و سطح بکارگیری ماشین‌های مجازی برای بررسی سطح کارآیی رویکرد پیشنهادی

در ادامه ابتدا در مورد الگوریتم یادگیری تقویتی عمیق، نحوه آموزش شبکه q و الگوریتم ازدحام ذرات توضیحاتی ارائه شده و سپس مقالات مرتبط در این حوزه مرور شده است. در بخش سوم، اجزای رویکرد پیشنهادی به صورت مرحله به مرحله معرفی و مدلسازی شده و در انتهای این بخش تابع هدف به صورت ریاضیاتی مدلسازی شده است.

[1] Deep Reinforcement Learning (DRL)

در بخش بعد، نتایج شبیه‌سازی در قالب معیارهای مختلف بررسی و نتایج در قالب شکل‌های متعدد نمایش و سپس نتایج با دیگر مقالات مقایسه شده است. در انتها نیز نتیجه‌گیری کلی از رویکرد طراحی شده ارائه شده است.

۲- مفاهیم پایه و مرور ادبیات

۲-۱- الگوریتم یادگیری تقویتی عمیق

در الگوریتم یادگیری تقویتی (RL)[۱]، عامل با انتخاب عملی از یک مجموعه تعریف شده سعی می‌کند پاداش جذب نماید. هدف اصلی در اجرای چنین رویکردی، کسب بیشترین پاداش در یک بازه زمانی طولانی است. در این شرایط، عوامل مختلفی مثل تابع پاداش و نوع خط مشی در نظر گرفته شده بر روی عامل اثر گذاشته و به کمک اجرای آنها تعامل با محیط برقرار می‌گردد. در این شرایط، یادگیری خط مشی به عامل کمک می‌کند تا نحوه اعمال خود را مدیریت نماید. در واقع یادگیری خط مشی به عامل کمک می‌کند تا نحوه اعمال خود را مدیریت کند. لذا از طریق کاوش در محیط و دریافت بازخورد، سیستم RL یک مدل تطبیقی را بدون نیاز به دریافت تعداد زیادی از داده‌ها تشکیل می‌دهد. به بیان دیگر، RL فرایندی را شامل می‌شود که در آن عامل‌ها به صورت پیوسته و مداوم با محیط تبادل برقرار می‌نمایند تا یک توالی از تصمیمات را اتخاذ و ظرفیت‌های تصمیم‌گیری خود را بهبود بخشند.

یادگیری تقویتی عمیق یک راهکاری مناسب برای سیستم‌های مبتنی بر یادگیری ماشین است که می‌خواهند یادگیری به صورت لحظه‌ای و به صورت کاملا خودکار صورت پذیرد. یادگیری تقویتی عمیق (DRL) در واقع یکپارچه‌کننده روش‌های یادگیری عمیق و یادگیری تقویتی است که این امر باعث ادغام قابلیت روش یادگیری عمیق با توانایی اتخاذ تصمیم در روش یادگیری تقویتی می‌گردد. رایج‌ترین رویکرد DRL، روش شبکه q عمیق (DQN)[۲] است. در روش‌های یادگیری سنتی، از جدول q برای ذخیره مقادیر عمل بهره گرفته می‌شود، در حالی که برای مسائلی که دارای ابعاد بزرگی هستند، امکان ذخیره‌سازی تمام اعمال در جداول و جستجوی مکرر برای انتخاب عمل مناسب در هر حالت، زمان‌بر و غیرکاربردی است. لذا شبکه DQN از شبکه عصبی جهت تقریب تابع Q و تولید اعمال بهره می‌گیرد. در این شرایط، برنامه‌ریز خط مشی ($\pi(a|s)$) که وظیفه نگاشت حالت‌ها به عمل‌ها را بر عهده دارد، هر وظیفه‌ای را به یک ماشین مجازی تخصیص می‌دهد. پاداش لحظه‌ای چنین عملی تحت عنوان (r_t) محاسبه می‌گردد. در واقع با هر پاداشی که عامل دریافت می‌کند، حالت آینده عامل ممکن است دستخوش تغییر گردد. بر این اساس ابتدا فرض می‌گردد برای هر

[۱] Reinforcement Learning (RL)

[۲] Deep Q-Network (DQN)

حالت، تابع حالت – ارزش عامل (V^{π}) به صورت رابطه (۱) تعریف گردد. تابع مقدار حالت – عمل (Q^{π}) نیز طبق رابطه (۲) محاسبه می‌گردد. در این شرایط هدف اصلی عامل به صورت یافتن خط مشی بهینه‌ای است که بر اساس آن، مقدار مورد انتظار پاداش بهینه گردد. این موضوع در روابط (۳) و (۴) نمایش داده شده است [5].

یادگیری توابع ارزش به صورت یک چالش در حوزه RL تعریف شده که بسیاری آن را مورد مطالعه قرار داده و رویکردهای متعددی هم برای آن ارائه شده است. رایج‌ترین تکنیک مورد استفاده در این حوزه، روش یادگیری Q است که ردگیری مقادیر تخمین تابع حالت – عمل را حفظ کرده و توسط تابع هدف بروزرسانی می‌گردد. در این شرایط، ضریب تنزیل γ باعث ایجاد تعادل بین پاداش‌های لحظه‌ای و پاداش‌های درازمدت می‌گردد. در حالی که در پروژه‌های متعدد، مقدار این پارامتر ثابت و تا حد امکان نزدیک به یک در نظر گرفته شده، اما تحقیقات اخیر نشان می‌دهد که این رویکرد نمی‌تواند لزوما به عنوان بهترین راهکار انتخاب گردد [6]. در واقع یک مقدار ثابت برای ضریب تنزیل، می‌تواند منجر به بروز رفتارهای ناسازگار در زمان، ایجاد خطا در مدل‌سازی اولویت‌های عامل و جستجوی غیر بهینه گردد [7].

$$V^{\pi}(s) = \mathrm{E}\left[\sum_{k=0}^{\infty}\gamma^{k}.r_{t+k}|\, s_t = s\right] \tag{1}$$

$$Q^{\pi}(s,a) = \mathrm{E}\left[\sum_{k=0}^{\infty}\gamma^{k}.r_{t+k}|\, s_t = s, a_t = a\right] \tag{2}$$

$$V^{*}(s) = \max_{\pi} V^{\pi}(s) \tag{3}$$

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{4}$$

۲-۲- آموزش شبکه Q

ایده اصلی در روش یادگیری تقویتی، آموزش عامل بر اساس تغییراتی است که در محیط رخ می‌دهد. یکی از رایج‌ترین تکنیک‌های مورد استفاده برای تعیین خط مشی بهینه در روش‌های یادگیری تقویتی، استفاده از الگوریتم یادگیری Q است. این الگوریتم یکی از روش‌های یادگیری تقویتی است که نیازی به مدل نداشته و به عنوان یک روشی کارامد در حل مسائل یادگیری تقویتی شناخته می‌شود. این الگوریتم تابحال در حوزه‌های متنوعی از جمله فرایندهای

صنعتی پیشرفته، کنترل شبکه، نظریه بازی، رباتیک، جستجوی عملیاتی، تئوری کنترل و تشخیص تصویر مورد استفاده قرار گرفته است.

برای مدلسازی ریاضیاتی این الگوریتم، فرض کنید π بیان‌گر خط مشی باشد. تابع Q به عنوان یک تابع عمل‌کننده بر روی زوج حالت – عمل، پاداش تجمعی عامل را از طریق رابطه (۵) تعیین می‌کند [8]. اما پیچیدگی پیاده‌سازی این رابطه با افزایش اندازه فضای حالت و فضای عملیاتی به صورت نمایی رشد می‌یابد و حل آن به صورت مستقیم امکان‌پذیر نیست. یکی از راهکارهای غلبه بر این چالش، تقریب زنی مقادیر Q با بهره‌گیری از تعدادی از متغیرهاست. به همین منظور یک ساختار ویژه‌ای به صورت حافظه بافر جهت تخمین‌زنی از طریق اموزش در DQN درنظر گرفته شده است. هدف از بکارگیری حافظه بافر، ذخیره‌سازی تعدادی از رکوردها بوده که هر رکورد خود شامل (s_t, a_t, r_t, s_{t+1}) می‌باشد. در این صورت یک بخش از حافظه به عنوان بخش پیش‌بین بکار گرفته شده و مقدار پاداش بادرنظرگرفتن حالت‌های جاری و عمل صورت پذیرفته مشخص می‌گردد. تابع پاداش‌دهی نیز مطابق رابطه (۶) تعریف می‌گردد. ضریب γ که به عنوان ضریب تنزیل در معادلات حضور یافته، به بهبود همگرایی کمک می‌نماید. در طول اموزش شبکه، همان بخش حافظه، از کل حافظه موجود استخراج شده تا شبکه Q از تجربیات گذشته خود یاد بگیرد.

$$Q^{\pi}(s_t, a) = (1-\alpha)Q(s_t, a) + \alpha.\left(r(s_t, a) + \gamma \max \acute{Q}\,(s_{t+1}, \acute{a}; \acute{\pi})\right) \tag{5}$$

$$R_t = r_{(t+1)} + \gamma r_{(t+2)} + \gamma^2 r_{(t+3)} + \cdots + \gamma^T r_{(t+T+1)} = \sum_{k=0}^{T} \gamma^k r_{(t+k+1)} \tag{6}$$

۳-۲- الگوریتم ازدحام ذرات

الگوریتم ازدحام ذرات (PSO) یک الگوریتم تکاملی جمعیت محور است که از رفتار پرندگان و حرکات گروهی حیوانات الهام گرفته شده است. در این ساختار، هر ذره مبین یک کاندید برای حل مساله بوده و موقعیت هر ذره به کمک ارتباط با بهترین موقعیت بدست آمده برای خود ذره تا گام فعلی و بهترین موقعیت بین تمام ذرات در هر گام تغییر می‌یابد. به بیان دیگر، سرعت ذره (V_i^{k+1}) و موقعیت هر ذره (X_i^{k+1}) در هر گام طبق روابط (۷) و (۸) به روزرسانی می‌گردد. در این شرایط، اگر برازندگی محاسبه شده برای هر ذره از مقادیر تخطی شده برای پارامترهای

مختلف تعریف شده در مساله تخطی نماید، لذا ان کاندید دوباره به روزرسانی شده تا یک گزینه جواب منطقی برای مساله حاصل آید. از سویی دیگر، به محض دست‌یابی به موقعیت بهتر توسط یکی از ذرات با جواب بهینه در گام فعلی، موقعیت ذره بهینه به روزرسانی شده و در هر گام ذره‌ای که دارای بهترین برازندگی باشد، به عنوان جواب بهینه تا آن گام معرفی می‌گردد.

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \left(P_i^k - X_i^k\right) + c_2 r_2 \left(G^k - X_i^k\right) \quad (7)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (8)$$

۴-۲- پیشینه موضوع

- تخصیص وظائف در فضای مه

فضای اینترنت اشیا بر پایه فضای ابری استوار است که دارای پتانسیل زیادی بوده و قادر است فضای ذخیره‌سازی و سرویس‌های پردازشی لازم را برای کاربران اینترنت اشیا فراهم آورد. در این شرایط، محاسبات مه یک ساختار و یک الگویی برای تکمیل ساختار فضای ابری است که در واقع تکمیل‌کننده فضای ابری بوده و در لبه شبکه در سمت کاربر قرار واقع می‌گردد. در چنین ساختاری فضای مه، ارائه خدمات را به کمک تبادل از طریق سیستم‌های کنترلی به کاربران برای اجرای فعالیت‌های خاص سرویس‌دهی نموده و همچنین امکان پردازش فعالیت‌هایی که به تاخیر حساس هستند را محقق می‌سازد. در [9] بیان شده که به بخاطر معایبی از جمله تاخیر زیاد، تراکم بالای شبکه و کاهش قابلیت اطمینان شرکت سیسکو قالب جدیدی به نام مه را ارایه نموده که به عنوان یک هاب در بین فضای ابری و تجهیزات مشترکین قرار می‌گیرد و می‌تواند بخشی از وظایف حافظه ابری مثل ذخیره‌سازی و پیش پرادزش اطلاعات را انجام دهد. با این‌حال، هزینه‌های بسیار زیاد بکارگیری ساختار مه پژوهشگران را مجاب می‌نماید تا به دنبال راهکارهایی برای به حداقل رساندن این هزینه‌ها باشند در حالی کیفیت سرویس‌دهی مطلوب هم حفظ گردد [10-11].

در [12] یک رویکرد مبتنی بر یادگیری تقویتی برای تخصیص منابع در شبکه‌های دسترسی به ظرفیت گره‌های مه برای اجرای برنامه‌های IoT با الزامات تأخیر متفاوت پیشنهاد شده است. این رویکرد شامل تصمیم‌گیری برای ارائه درخواست کاربر اینترنت اشیا به صورت محلی در لبه یا ارجاع آن به فضای ابری برای حفظ منابع برای کاربران در

آینده است. تابع هدف پیشنهادی از طریق فرآیند تصمیم‌گیری مارکوف (MDP)[1] و بکارگیری چندین روش یادگیری تقویتی برای حل مشکل MDP با یادگیری از خط‌مشی‌های تصمیم‌گیری بهینه طراحی شده است. نتایج شبیه‌سازی با در نظر گرفتن ۱۹ محیط مبتنی بر اینترنت اشیای متفاوت و با الزامات تأخیر ناهمگن ارایه شده که تأیید می‌کند که روش‌های مبتنی بر RL بدون توجه به محیط اینترنت اشیا دارای پتانسیل عملکرد بالایی هستند. این مقاله عملکرد و سازگاری برتر روش‌های RL را نسبت به روش‌های مدل‌سازی شبکه از طریق شبیه‌سازی‌های گسترده تأیید نموده است. روش‌های RL تعادل مناسبی بین دو هدف متضاد ایجاد نموده و میانگین کل ارائه شده را در مقابل به حداقل رساندن زمان عدم کارکرد گره‌های مه را به حداکثر رسانده که به استفاده مؤثر از منابع محدود در فضای مه کمک می‌کند. این مقاله تحلیل دقیقی از پیچیدگی محاسباتی الگوریتم پیشنهادی ارائه نداده، که می‌تواند از نظر مقیاس‌پذیری یک محدودیت باشد.

با وجود مزایای بالقوه موجود در ساختار فضای پردازشی مه، هنوز چالش‌هایی مرتبط با پویا بودن، ناهمگنی و پیچیدگی بالای درخواست‌های ارسالی توسط مشترکین وجود دارد. در این راستا، یکی از مهم‌ترین مشکلات موجود، مدیریت منابع محدود تعریف شده در این فضا است. به منظور برخورداری از یک سطح کیفیت سرویس‌دهی (QoS)[2] مطلوب، لازم است که ظرفیت موجود در منابع به خوبی مدیریت گردد تا همه معیارهای لازم برای بهبود رضایت مشترکین را با خود به همراه داشته باشد. از سویی دیگر، در چنین ساختاری درخواست‌های ارسالی به صورت یک جریان پیوسته به سیستم وارد شده و اگر تاخیری در پردازش یکی از درخواست‌ها پیش آید، برنامه‌ریزی‌ها دچار اختلال شده و بدین ترتیب مدیریت درخواست‌های ارسال شده باید به صورتی محقق گردد که با کمترین تاخیر ممکن و استفاده حداکثری از ظرفیت‌های موجود، وظائف پردازشی اجرا شود. در واقع رضایت‌مندی مشترکین سرویس‌های داده‌ای (DSS)[3] شامل عدم تاخیر از بازه تعیین شده، ارسال پاسخ‌های مطلوب و حفظ کیفیت سرویس‌دهی در سطح مناسب از مهمترین اهداف در زمان تخصیص‌دهی منابع در یک فضای ابر – مه است.

در [13] یک استراتژی تخصیص منابع و زمان‌بندی برای محاسبات مه با استفاده از یک الگوریتم جستجوی چندهدفه پیشنهاد شده که هدف آن به حداکثر رساندن نسبت موفقیت و شرایط امنیتی بوده و برای نشان دادن کارایی آن، عملکرد رویکرد با تعدادی از الگوریتم‌های موجود مقایسه شده است. در این مرجع، مزایای محاسبات در فضای مه مانند کاهش تاخیر شبکه، بهبود امنیت و کاهش هزینه‌های عملیاتی ارزیابی شده است. این مطالعه مصرف انرژی یا مقرون به صرفه

[1] Markov Decision Process (MDP)

[2] Quality of Service (QoS)

[3] Data Service Subscribers (DSS)

بودن الگوریتم پیشنهادی را در نظر نگرفته که می‌تواند در برخی سناریوها از عوامل مهم باشد. علاوه بر این، عملکرد الگوریتم پیشنهادی به پارامترها و تنظیمات خاص مورد استفاده بستگی داشته که ممکن است برای همه شرایط بهینه نباشد و در نهایت، این مطالعه تجزیه و تحلیل دقیقی از مقیاس‌پذیری و استحکام الگوریتم پیشنهادی ارائه نداده که می‌تواند برای سیستم‌های محاسباتی مه در مقیاسی بزرگتر و پیچیده‌تر مهم باشد.

در [14] از یک تکنیک تخصیص منابع آگاه از تاخیر در شبکه‌های IoT به کمک فضای مه و با استفاده از یادگیری تقویتی استفاده شده است. هدف این تکنیک به حداقل رساندن تاخیر در روند تخصیص منابع در کانال بی‌سیم و گره مه و در عین حال رعایت محدودیت‌ها (QoS) است. در این راستا، فرمول‌بندی مسئله تخصیص منبع با استفاده از یادگیری تقویتی به یک مسئله غیر خطی عدد صحیح تبدیل شده که در آن هر دو منبع رادیویی و منبع محاسباتی در نظر گرفته شده اند. نویسندگان همچنین در مورد فرمول‌بندی مسئله تخصیص منابع و طراحی یک الگوریتم یادگیری تقویتی آنلاین برای اتخاذ تصمیم‌های بهینه در زمان واقعی بحث کرده‌اند و با برجسته کردن نتایج شبیه‌سازی، اثربخشی تکنیک پیشنهادی را نشان داده‌اند. نتایج نشان می‌دهد که تکنیک پیشنهادی می‌تواند به طور قابل توجهی تاخیر کار را کاهش دهد، در حالی که محدودیت‌های QoS برآورده شود. در این مرجع، به صراحت به هیچ محدودیتی در مورد تکنیک پیشنهادی اشاره نشده است. با این حال، یک محدودیت بالقوه می‌تواند عدم اطلاع کامل از وضعیت سیستم باشد، که ممکن است در سناریوهای دنیای واقعی عملی روی دهد. علاوه بر این، تکنیک پیشنهادی ممکن است به مقدار قابل توجهی از منابع محاسباتی برای پیاده‌سازی در یک سیستم بلادرنگ نیاز داشته باشد. در [15] یک چارچوب کنترل سلسله مراتبی برای تعادل بار و تخصیص منابع در خدمات رایانش ابری برای اطمینان از الزامات کیفیت خدمات (QoS) در برنامه‌های پایش ساختاری و در عین حال بهینه‌سازی استفاده از زیرساخت‌های موجود ارائه شده است. این چارچوب از دو لایه کنترلی تشکیل شده که یکی برای تخصیص منابع و نحوه مدیریت بکارگیری ماشین‌های مجازی و دیگری برای متعادل‌سازی بار و قرار دادن ماشین‌های مجازی در خوشه‌ای از سرورهای فیزیکی تعیین شده‌اند. نتایج عددی نشان می‌دهد که بکارگیری دو لایه کنترلی، سطح رضایت از محدودیت‌های سیستم و نیازهای کاربر را با توجه به نوسانات درخواست‌های دریافتی تضمین نموده است. از محدودیت‌های کاربرد رویکرد پیشنهادی در این مقاله، چارچوب پیشنهادی از طریق شبیه‌سازی‌های عددی ارزیابی شده و مشخص نیست که در یک سناریوی واقعی چقدر خوب عمل می‌کند. هم‌چنین، فرض شده است که هر ماشین مجازی یک برنامه را میزبانی می‌کند، که ممکن است در سناریوهای دنیای واقعی که ممکن است چندین برنامه روی یک ماشین مجازی میزبانی شوند، چنین امری میسر نباشد. در [16] از ساختار تقویتی عمیق مبتنی با یادگیری Q جهت تخصیص‌دهی بهینه منابع قرارگرفته در لبه شبکه استفاده شده است. سیستم بهینه‌ساز طراحی شده در قالب حداقل‌سازی هزینه انرژی مصرفی، هزینه محاسباتی و هزینه تاخیر

زمانی تعریف شده است. در این مرجع تخصیص فعالیت‌ها به صورت آفلاین صورت پذیرفته و بخاطر تاخیر زمانی زیاد، رویکرد ارائه شده نمی‌تواند به خوبی در مسائل برخط مورد بهره‌برداری قرار گیرد. از سویی دیگر، یک رویکرد برنامه‌ریزی خطی عدد صحیح - مرکب (MILP)[1] جهت مدل‌سازی بارگذاری وظائف مستقل با مهلت زمانی محدودشده در فضای مه توسط نویسندگان [17] ارائه شده است. در این روش تجهیزات IoT نادیده گرفته شده و ظرفیت فضای محاسباتی ابری به صورت نامحدود فرض شده است. در [18]، نویسندگان دو مدل برنامه‌ریزی شده‌ای را برای تشکیل گروه‌های ابری با استفاده از نظریه بازی‌های تکاملی و الگوریتم ژنتیک ارائه کرده‌اند. برای بهبود سود کل گروه از نسلی به نسل دیگر، الگوریتم ژنتیک فضای جستجو را بررسی کرده و مناسب‌ترین فضای محاسباتی را پیدا می‌کند، سپس مدل ژنتیکی توسط یک بازی تکاملی بهبود می‌یابد. محدودیت اصلی این روش، زمان‌بندی وظائف است که با تلاش برای بهینه‌سازی مجموعه‌ای از پارامترها در هر بار دریافت درخواست، به صورت آفلاین عمل نموده که مستلزم زمان اجرای بالایی است و برای کارهای حساس به تاخیر ناکارآمد است. در [19] یک رویکرد مبتنی بر یادگیری تقویتی عمیق به منظور زمان‌بندی وظائف در مبتنی بر لبه پیشنهاد شده است. هدف این رویکرد به حداکثر رساندن درجه رضایت از کار با تخصیص چندین کار بر روی ماشین‌های مجازی پیکربندی شده در سرور لبه است. هدف از رویکرد پیشنهادی، به حداکثر رساندن درجه رضایت کاربران در یک دوره طولانی مدت است. این چالش به عنوان یک فرآیند تصمیم مارکوف (MDP) فرمول‌بندی شده که حالت، عمل، انتقال حالت و پاداش برای آن طراحی شده است. رویکرد پیشنهادی تنوع وظایف و ناهمگونی منابع موجود را نیز در نظر گرفته است. علاوه بر این، از شبکه عصبی برای استخراج ویژگی‌ها بهره گرفته شده است. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم زمان‌بندی کار مبتنی بر DRL از روش‌های موجود از نظر میانگین درجه رضایت از پوشش‌دهی وظائف و نسبت موفقیت بهتر عمل کرده است. با اینکه رویکرد پیشنهادی می‌تواند در سایر حوزه‌های تحقیقاتی مرتبط مانند محاسبات ابری، محاسبات مه، و محاسبات لبه‌ای سیار گسترش یابد ولی در آزمایش‌های دنیای واقعی پیاده‌سازی نشده و ممکن است به طور مستقیم برای سناریوهای دیگر مانند محاسبات ابری و محاسبات در لایه مه قابل اجرا نباشد. در [17] با در نظرگرفتن مقیاس‌پذیری و وابستگی وظائف، یک طرح زمان‌بندی جدید مبتنی بر یادگیری تقویتی عمیق ارائه شده که قادر به کاهش هزینه‌های انرژی در مقیاس‌های بالا با تعداد سرور زیاد و درخواست کاربران متعدد می‌باشد. این روش در دو مرحله تامین منابع و زمان‌بندی وظائف به طور خودکار بهترین تصمیمات را در بلندمدت با توجه به تغییر محیط در رابطه با درخواست کاربران می‌گیرد.

[1] Mixed-Integer Linear Programming (MILP)

این مقاله تنها به مساله کاهش هزینه انرژی پرداخته است و دیگر پارامترهای کیفیت سرویس را در نظر نگرفته است. بطور خلاصه کارهای مرور شده در زمینه تخصیص منابع در فضای مه در جدول (۲-۱) گردآوری شده است.

جدول ۱: مقایسه روش‌های بکارگرفته شده برای تخصیص وظائف

مرجع	معیارهای ارزیابی شده	رویکرد مورد استفاده
[2]	زمان پاسخگویی، میزان هزینه و تعداد تلفات کاربری‌ها	روش بهینه‌سازی تابع لیاپانوف
[21]	کل زمان اجرا و هزینه منابع	الگوریتم برنامه‌ریزی وظائف چندهدفه با استراتژی همسایگی تطبیق شونده
[22]	میزان سود با درنظرگرفتن محدودیت منابع	بهینه‌سازی محدب برای تخصیص مناسب توان و الگوریتم ژنتیک برای برنامه‌ریزی آن
[23]	ترکیب بهره‌گیری از دو تکنیک نظریه بازی مختلف در ترکیب با شبکه عمیق Q، بهینه‌سازی خط‌مشی عامل و گرادیان نزولی	برآورد سطح مشارکت کاربران در پایش جمعی سیار
[24]	استفاده از الگوریتم تبرید فلزات جهت مدیریت درخواست‌ها	تخصیص وظائف به صورت لحظه‌ای حداقل‌سازی سطح انرژی مصرفی ارتقای کیفیت پایش
[25]	بهبود مدیریت تخصیص درخواست‌های مشترکین به کمک تخمین نوع رفتار آنها	حداقل‌سازی هزینه مشارکت افزایش سطح کیفیت اطلاعات

- کارکردهای الگوریتم یادگیری Q

در حالت کلی، الگوریتم RL یک راه حل توسعه یافته‌ای را برای فرایندهای اتخاذ تصمیمات پیچیده ارائه می‌دهد. اگرچه تاکنون روش‌های متعددی برای یادگیری از تعامل‌های صورت گرفته ارائه شده، اما در RL هایی فاقد مدل یادگیری، توابع ارزش به کمک تخمین میزان مطلوب بودن قرارگیری یک عامل در یک حالت مشخص یا انجام یک عمل مشخص در یک حالت ویژه انجام می‌پذیرد [5]. میزان کیفیت جفت حالت – عمل هم معمولا به کمک مقدار پاداش مورد انتظار برای گام‌های زمانی در آینده ارزیابی می‌گردد. تخمین دقیق مقادیر حالت – عمل به عنوان یکی از ارکان اساسی در روش‌های RL فاقد مدل قلمداد شده و بر این اساس است که توابع مقدار، تعریف‌کننده عمل‌ها بوده

و به آن‌ها اجازه می‌دهند که به صورت مناسبی با محیط اطراف خود واکنش نشان دهند. در انتخاب استراتژی عوامل متعددی تاثیرگذار هستند. مثلا قطعی بودن اعمال یا غیرقطعی بودن نتایج آنها، قابلیت تخمین حالت بعد از انجام هر عمل یا عدم توانایی پیش‌بینی وضعیت آینده، آموزش عامل توسط مربی برای انجام اعمال بهینه یا یادگیری فقط بر اساس انجام اعمال، می‌تواند در این شرایط تاثیرگذار باشد. در ادامه مرور مختصری بر کارکردهای مختلف الگوریتم یادگیری Q در جدول (۲) ارائه شده است.

جدول ۲: مقایسه کارکردهای مختلف الگوریتم Q

مرجع	کارهای انجام شده	محدودیت ها
[9]	استفاده از معادله بلمن برای توصیف RL- بکارگیری روش های افلاین و انلاین برای تعیین خط مشی جهت حل مسائل کنترلی	عدم دسته‌بندی الگوریتم‌ها و مقایسه عملکرد آنها در مسائل بیان شده
[10]	بررسی جامع کارکردهای استفاده کننده از یادگیری Q دسته‌بندی روش‌ها بر اساس رفتار عامل‌ها	عدم ارائه جزئیات بکارگیری تکنیکی و عملی روش‌های یادگیری Q فقدان بررسی روش های بکارگیری چندعامل و کارکردهای آنها
[26]	طبقه‌بندی یادگیری‌های تقویتی متعدد با تمرکز ویژه بر روی یادگیری Q عمیق ارزیابی ویژگی‌ها، چالش‌ها و الگوریتم‌های یادگیری تقویتی عمیق	عدم ارائه یک پیش زمینه ریاضیاتی جامع فقدان بررسی کامل کارکردهای معرفی شده
[27]	استفاده از الگوریتم یادگیری Q در جهت همگرایی مشارکت کاربران و بهبود شرایط امنیتی در پایش جمعی سیار	عدم مدلسازی حرکت کاربران سیار و تغییرات شرایط دینامیکی آنها در گذر زمان

۳- مشخصات رویکرد پیشنهادی

در رویکرد پیشنهادی، ماشین‌های مجازی درنظرگرفته شده برای اجرای وظائف همان عامل‌ها در روش یادگیری تقویتی عمیق می‌باشند. خط مشی طراحی شده همان الگوی تخصیص‌دهی منابع بوده که درخواست‌های ارسال شده را بر روی ماشین‌های مجازی نگاشت می‌دهد. پردازش درخواست‌های تخصیص یافته به عنوان عمل فرض شده و میزان سود دریافت شده هم برابر میزان پاداش در یادگیری تقویتی است. در این راستا، برای بدست آوردن مقادیر پاداش یادگیری تقویتی (RL) از یادگیری تقویتی عمیق دوگانه (DDQN) بهره گرفته شده که از ویژگی‌های تابع تقریب زننده مؤلفه‌های جدول Q به کمک شبکه عصبی بهره می‌برد. این تقریب‌زنی به این خاطر اجرا می‌شود که اندازه فضای حالت – عمل تعریف شده برای پروژه دارای ابعاد بالایی بوده و دارای دینامیک تغییرات شدیدی است. به بیان دیگر، بر اساس وضعیت حالت‌های ورودی، مقادیر جدول Q جهت تعیین عمل مربوطه تخمین زده می‌شود.

شبکه عصبی مورد استفاده شامل یک شبکه عصبی پیش رو با یک لایه ورودی، دو لایه پنهان و یک لایه خروجی بوده که تمام لایه‌های پنهان از تابع خطی اصلاح شده برای فعال‌سازی بهره گرفته شده است. به منظور استفاده از داده‌های ذخیره‌شده از جفت حالت – عمل از یک حافظه بازاجرا استفاده شده که هم از انحراف فرایند یادگیری جلوگیری می‌نماید و هم سرعت آموزش شبکه را افزایش می‌دهد. برای پیاده‌سازی رویکرد پیشنهادی فرضیات زیر درنظر گرفته شده است:

- شیوه زمان‌بندی به صورتی است که تا قبل از اتمام وظیفه درنظرگرفته شده برای هر ماشین مجازی، نمی‌توان تغییری در شرایط منبع اختصاص یافته ایجاد نمود.
- برای هر ماشین مجازی و در هر گام، فقط یک وظیفه قابل اجرا است.
- مقادیر پاداش لحظه‌ای و حالت بعدی پس از اتمام اجرای وظیفه محاسبه و به همراه مقادیر حالت اولیه در حافظه سیستم برنامه‌ریز ذخیره و از این اطلاعات برای تعیین استراتژی مناسب استفاده می‌شود

۳-۱- یادگیری عامل

در این ساختار عامل از روش DDQN برای تعیین عمل‌ها به منظور دست‌یابی به هدف مورد نظر استفاده می‌کند. این هدف به صورت دست‌یابی به مقادیر بهینه برای معیارهای درنظرگرفته شده تعریف می‌شود. این شبکه از داده‌های ورودی (شامل مشخصات درخواست‌های پذیرش شده) به منظور تولید خروجی (طراحی نگاشت وظائف بر روی ماشین‌های مجازی مشارکت کننده) بهره گرفته و برای پیاده‌سازی شبکه عصبی مورد نظر از یک شبکه عصبی متشکل از یک مدل خطی از لایه‌ها استفاده شده است. لایه ورودی دارای ۲۴ نرون (برابر با فضای حالت) بوده که از واحد خطی اصلاح شده (ReLU)[۱] به عنوان عملگر فعال‌ساز بهره گرفته و تعداد نرون‌ها در لایه مخفی ۲۴ عدد در نظر گرفته شده است. لازم به ذکر است که تمام لایه‌های پنهان نیز از تابع خطی اصلاح شده برای فعال‌سازی نرون‌های طراحی شده برای انها بهره می‌برند. تعداد نرون ها در لایه خروجی نیز برابر اندازه فضای عمل بوده که در این لایه از تابع فعال‌ساز خطی (Linear) بهره گرفته شده است.

۳-۲- حافظه بافر

[۱] Rectified Linear Unit (ReLU)

در شبکه Deep Q به منظور جلوگیری از بروز انحراف در توزیع داده‌ها از تکنیکی به نام حافظه بافر استفاده می‌شود. دلیل این امر به خاطر یادگیری بر اساس نمونه‌های کوچک بدست آمده می‌باشد زیرا بدلیل همبستگی موجود در بین توالی حالت‌ها، فرایند یادگیری در شبکه با چالش‌های جدی روبرو می‌گردد. در چنین ساختاری، یادگیری Q بر روی جفت‌های حالت – عمل به صورت مستقیم و در هنگام شبیه‌سازی اعمال نمی‌گردد، بلکه به جای آن داده‌های استخراج شده در فرایند شبیه‌سازی در بافر مورد نظر ذخیره می‌گردد. در این شرایط عامل با یک بافر خالی شروع نموده و سپس هنگام آموزش و اجرا آن را پر می‌کند. در هر گام زمانی یک تاپل چهارگانه شامل حالت فعلی، عمل فعلی، پاداش و حالت بعدی در حافظه بازاجرا ذخیره می‌گردد. اگر حالتی از قبل در حافظه بافر وجود داشته باشد، برای کسب پاداش و عمل بعدی دیگر نیازی به محاسبه نمی‌باشد.

۳-۳- شبکه هدف

به این خاطر که در یادگیری تقویتی، داده‌ها بر چسب‌گذاری نشده‌اند، لذا از یک شبکه هدف به منظور محاسبه مقدار Q برای هر جفت حالت – عمل استفاده می‌شود. در این راستا استفاده از یک شبکه عصبی برای محاسبه مقادیر هدف و مقادیر تخمین زده شده ممکن است باعث بروز واگرایی گردد. به همین منظور، در ابتدا پارامترهای شبکه هدف با پارامترهای شبکه اصلی برابر انتخاب شده و بعد در هر مرحله شبکه هدف بروزرسانی نشده و فقط با برداشت پارامترهای شبکه اصلی، به روزرسانی می‌گردد. این ساختار به درهم شکستن همبستگی کمک نموده و از بروز نوسانات جلوگیری می‌نماید. از سویی دیگر، هایپرپارامترهای شبکه به کمک بهره‌گیری از الگوریتم PSO به روزرسانی می‌شوند. در واقع این رویکرد باعث می‌شود که در زمان مشاهده نتایج ضعیف‌تر در چند گام متوالی، مقادیر جدول Q توسط تغییر ضرائب شبکه، دوباره به روزرسانی گردد تا از عدم واگرایی تخصیص منابع موجود اطمینان حاصل گردد.

۳-۴- فرآیند آموزش در DDQN

در رویکرد پیشنهادی، از دو شبکه عصبی برای پیاده‌سازی فرآیند آموزش بهره گرفته شده است. در مرحله اول، اوزان شبکه اصلی (ω) به طور تصادفی مقداردهی شده و سپس در شبکه هدف از اوزان ($\acute{\omega}$) مشابه شبکه اصلی بهره گرفته شده و یک حافظه باز اجرای خالی برای ذخیره نمونه‌ها تعریف می‌شود. استفاده از دو شبکه به طور همزمان منجر به ایجاد ثبات در فرایند یادگیری شده و به بهبود تاثیر کارایی الگوریتم کمک می‌نماید. در هنگام اجرای فرآیند یادگیری، از شبکه هدف برای بازیابی مقادیر آزمایش شده استفاده شده در حالی که در شبکه اصلی، تمام به روزرسانی‌ها

در مرحله آموزش اجرا می‌گردد. با پیشرفت فرایند یادگیری، هایپرپارامترهای شبکه با پارامترهای شبکه هدف هماهنگ می‌گردد.

۳-۵- تابع هدف

به منظور طراحی تابع هدف، فرض کنید مجموعه $T = \{t_1, t_2, \dots, t_m\}$ برابر با مجموعه وظائف در نظر گرفته شده باشد. همچنین، درخواست‌های ارائه شده به سیستم بایستی توسط n تا ماشین مجازی v_j مورد پردازش قرار گیرد. بدین ترتیب تابع هدف نهایی به صورت رابطه (۹) قابل بیان است. در این رابطه، $E_i(\mathrm{X})$ مبین میزان احتمال تکمیل درخواست t_i داده شده به سیستم از طریق ارزیابی ماتریس X است. این ماتریس از نوع باینری و دارای ابعاد $m \times n$ بوده که نگاشت صورت گرفته بین وظائف با کاربران تعریف شده را مشخص می‌نماید. در این شرایط، برای هر وظیفه تعیین شده (t_i)، اگر ماشین مجازی (v_j) برای آن تخصیص یابد، آنگاه مقدار x_{ij}آن برابر یک و در غیر این صورت مقدار x_{ij} برابر صفر می‌گردد.

$$\max \sum_{i=1}^{m} E_i(\mathrm{X}) \tag{9}$$

$$E_i(\mathrm{X}) = \begin{cases} \dfrac{\sum_{j=1}^{n} P(t_i\ v_j) \times x_{ij}}{\sum_{j=1}^{n} x_{ij}} & if \sum_{j=1}^{n} x_{ij} \neq 0 \\ 0 & if \sum_{j=1}^{n} x_{ij} = 0 \end{cases} \tag{10}$$

در این رویکرد، خروجی شبکه هدف توسط پاداش‌دهی بروزرسانی می‌گردد و به عنوان یک معیار در ارزیابی سطح عملکرد مورد استفاده قرار می‌گیرد (رابطه (۱۱)). در ادامه، از الگوریتم ازدحام ذرات (PSO) به منظور تعیین مقادیر هایپرپارامترهای الگوریتم یادگیری تقویتی استفاده شده تا اختلاف بین خروجی شبکه هدف و خروجی شبکه تخمین زده شده طبق رابطه (۱۲) حداقل گردد. در ساختار رویکرد تخصیص منابع، نیاز است که احتمال تکمیل درخواست t_i توسط ماشین مجازی v_j به خوبی برآورد گردد. در این راستا، میزان احتمال تکمیل وظائف $P(t_i\ v_j)$ توسط رابطه

(۱۳) محاسبه می گردد. اگر برای تعدادی از ماشین‌های مجازی، اطلاعات کاملی از آنها در دسترس نباشد، این احتمال به صورت تصادفی محاسبه می‌گردد.

$$Q^{lab}(s_t, a) = r(s_t, a) + \gamma . max Q^{tar}(s_{t+1}, \acute{a}, \acute{w}) \quad (11)$$

$$Loss = \left(Q^{lab}(s_t, a) - Q^{pre}(s_t, a, w)\right)^2 \quad (12)$$

$$P(t_i\ v_j) = \frac{\sum_{k=1}^{|R_i^j|} P_{ijk}^*}{|R^j|} \quad (13)$$

مراحل الگوریتم پیشنهادی بر اساس یادگیری Q

1:	Initialize the Q-network with random parameters (ω) (Main netwok)
2:	Initialize the Q'-network with random parameters ($\acute{\omega}$) (Target netwok)
3:	For step = 1…TS do
4:	With probability of ε select $a_t = Max\ Q^{\pi}(s_t, a)$
5:	Otherwise choose a random action a_t
6:	Obtain reward $R_t = \sum_{k=0}^{T} \gamma^k r_{(t+k+1)}$ and reach new state s_{t+1}
7:	Store (s_t, a_t, r_t, s_{t+1}) in Buffer
8:	Perform a gradient descent step L = (yi-$Q^{\pi}(s_t, a)$)2
9:	With respect to the network parameters (ω), calculate loss function
10:	Every k step respect $\acute{\omega} = \omega$ (Update the target network)

یکی از معیارهای مهم در زمینه تخصیص منابع، حداقل‌سازی بازه زمانی اتمام فعالیت است. این بازه زمانی شامل مدت زمان ارسال درخواست تا دریافت نتایج توسط ماشین مجازی می‌شود. برای محاسبه این شاخص، می‌توان میزان تاخیر در انجام وظائف توسط کاربران را به عنوان معیار محاسبه نمود (رابطه ۱۴). برای این منظور از مدل تاخیر زمانی در پردازش درخواست‌های مورد نیاز ارائه شده در [17] بهره گرفته شده و منابع اولیه از بروکر انتخاب شده است. از آنجایی که پردازش کامل تمام درخواست های ارائه شده در پایگاه داده Borg فراتر از توان محاسباتی معمول و در دسترس می باشد، لذا برای این منظور ۵ نوع ماشین مجازی و ۲۱ درخواست به صورت تصادفی از بین مجموعه انتخاب شده است. عملیات اموزش نیز در زمانی آغاز می گردد که حافظه بازاجرا پر شده باشد (معمولا بعد از اتمام ۶ اپیزود). سطح کیفیت سرویس‌دهی توسط رابطه (۱۵) اندازه‌گیری شده است.

$$T_n^{loc} = \sum_{m=1}^{M} T_{nm}^{loc}(1 - x_{nm}) \tag{14}$$

$$E_k^i = \frac{1}{T_i} \times \frac{N_{k,}^i \times T_{k,}^i}{\sum_{i=1}^{n} N_{k,g}^i \times Ts_{k,g}^i} \times \delta_i \tag{15}$$

۴- مشخصات رویکرد و ارزیابی نتایج

در این مقاله، برای انجام شبیه‌سازی‌ها از مجموعه داده Borg ارائه شده برای سال ۲۰۱۹ شرکت گوگل بهره گرفته شده است. اطلاعات مجموعه داده انتخاب شده در جدول (۳) درج شده است. کل مجموعه در یک بازه زمانی یک ماهه محاسبه شده و ۴۰ میلیون وظیفه را شامل می‌شود که در میان بیش از ۱۲۰۰۰ ماشین توزیع شده است [27]. از آن‌جایی که کار بر روی مجموعه داده ارائه شده توسط امکانات در دسترس به صورت کاربردی تحقق‌پذیر نمی‌باشد، لذا بازه زمانی یک ساعته بین ۶:۳۰ صبح تا ۷:۳۰ صبح به صورت تصادفی انتخاب شده و حداکثر مدت زمان پایش لازم برای وظائف درخواستی نیز برابر ۱۵ دقیقه منظور شده است. در جدول (۴) مشخصات ۴ نوع مختلف از وظائف بیان شده است. البته در زمان دریافت درخواست‌ها، لازم است اولویت اجرای آنها نیز مشخص گردد.

جدول ۳: اطلاعات مجموعه داده Borg [28]

پارامتر	مقدار
تعداد ماشین‌های مجازی	۳۵۰ _ ۵۰
بازه زمانی مشارکت	۶:۳۰ صبح تا ۷:۳۰ صبح
تعداد وظائف درخواست شده	۱۶۰۰ _ ۴۰۰
حداکثر مدت زمان پردازش	۱۵ دقیقه

در تئوری، مجموع محدودیت‌های تمام فعالیت‌های در حال اجرا نبایستی از ظرفیت ماشین مورد نظر تخطی نماید. در این حالت، تعداد وظائف درخواستی از ۴۰۰ تا ۱۶۰۰ درخواست متغیر بوده و تعداد ماشین‌های مجازی در دسترس نیز در بازه ۵۰ تا ۳۵۰ تعریف گردیده است. مرحله آموزش انتخاب خط مشی بهینه در روش پیشنهادی توسط اجرای اپیزودها صورت پذیرفته است. بایستی توجه شود که تعداد ماشین‌های مجازی و تعداد درخواست‌های تعیین شده در مجموعه داده مورد مطالعه بسیار فراتر از ظرفیت‌های پردازشی و امکانات در دسترس نویسنده بوده و لذا مطابق با

رویکردهای مورد استفاده در مقالات مرجع، تعداد محدودی از رکوردهای ثبت شده از داده‌های موجود برای پیاده‌سازی رویکرد پیشنهادی مورد استفاده قرار گرفته است.

جدول ۴: مشخصات ۴ مدل از درخواست های مجموعه Borg [29]

Type 0	37	6	212	1260
Type 1	36	0	67	2280
Type 2	90	28	75	1276
Type 3	21	10	0	2

این وظائف دارای اولویت‌بندی‌های متفاوتی می‌باشند که کمترین اولویت به دسته ردیف آزاد (۰ تا ۹۹) و بیشترین اولویت به ردیف مانیتورینگ (بالاتر از ۳۶۰) تعلق دارد. مشخصات پارامترهای بکاررفته در الگوریتم PSO نیز در جدول (۵) ارائه شده است. لازم به ذکر است که بازه تغییرات طوری طراحی می‌گردد که مقادیر هایپرپارامترها از محدوده مجاز تعیین شده فراتر نرفته و اگر موقعیت ذره از موقعیت بهترین ذره برتر باشد، آنگاه جایگاه نقطه بهینه تعویض می‌گردد.

جدول ۵: مشخصات الگوریتم ازدحام ذرات (PSO)

پارامتر	مقدار
تعداد ذرات	۵۰
تعداد تکرارها برای بهینه‌سازی هایپرپارامترها	۱۰
تعداد مراحل در هر اپیزود	۵
اندازه حالت	۸
مقدار ضریب اینرسی (ω)	۰,۵
وزن بهترین موقعیت هر ذره (c_1)	۲
وزن بهترین موقعیت سراسری (c_2)	۲

۴-۱- تنظیم هایپرپارامترهای الگوریتم DDQN

برای مدلسازی منابع از کلاس بروکر (Broker) استفاده شده که نه تنها برای هر بروکر یک شناسه (ID) اختصاص می‌یابد، بلکه میزان ظرفیت و حافظه در دسترس نیز از قبل مشخص می‌باشد. برای ایجاد مدل یادگیر عمیق از مدل Sequential در کتابخانه Keras بهره گرفته شده که یک مدل خطی از لایه‌ها تشکیل می‌دهد و از تابع خطای حداقل

مربعات (MSE)[1] و بهینه‌ساز Adam نیز برای کامپایل کردن کمک گرفته شده است. برای بهبود سطح کارآیی رویکرد DDQN طراحی شده، از هایپرپارامترها بهره گرفته شده که شامل نرخ یادگیری (Learning rate)، نرخ تنزیل (Discount rate)، نرخ کاوش (Epsilon)، نرخ میرای کاوش (Epsilon decay) و تعداد مراحل در هر اپیزود می‌باشد. خروجی این الگوریتم هم به عنوان میانگین پاداش پس از اتمام اپیزودها تعیین شده است. در ادامه بخش‌های مختلف به روز رسانی هایپرپارامترهای الگوریتم DDQN برای مسئله تخصیص‌دهی منابع در محیط مه ارزیابی می‌شود.

- **نرخ یادگیری**: در مسئله تخصیص‌دهی وظائف در محیط مه، مقدار نرخ یادگیری (Learning rate) در بازه ۰,۰۰۰۱ تا ۰,۰۰۱ تنظیم می گردد. به بیان دیگر، این ضریب میزان تاثیرگذاری پاداش فعلی بر به‌روزرسانی تخمین مقدار Q را نشان می‌دهد. با افزایش مقدار ضریب یادگیری، تاثیر پاداش فعلی در به‌روزرسانی تخمین مقدار Q افزایش می‌یابد و الگوریتم سریع‌تر به بهینه‌سازی هدف خود نزدیک می‌شود. با این حال، مقدار بالای ضریب یادگیری ممکن است باعث شود که الگوریتم به نقطه شکست برسد یا بیش‌برازش روی دهد. به‌طور کلی، باید ضریب یادگیری را به‌گونه‌ای تنظیم کرد که الگوریتم به بهینه‌سازی هدف خود برسد و در عین حال دچار بیش‌برازش یا شکست نشود. بعد از اجرای تعداد ثابتی از گام‌های یادگیری، وزن‌های مدل هدف با وزن‌های مدل اصلی یکسان می‌گردد تا نرخ یادگیری بهبود یابد.
- **نرخ تنزیل**: این پارامتر نشان می‌دهد که چه مقدار از پاداش آینده باید در محاسبه پاداش کل مورد استفاده قرار گیرد. برای مثال، اگر ضریب تنزیل (Discount rate) برابر با ۰,۹ باشد، پاداش آینده با فاصله زمانی یک مرحله، ۰,۹ بار ارزش پاداش کنونی را دارد. در مسئله تخصیص‌دهی وظائف در محیط مه، مقدار ضریب تنزیل در بازه ۰,۹ تا ۰,۹۹ محدود شده و با افزایش مقدار این ضریب، تاثیر پاداش‌های آینده در به‌روزرسانی تخمین مقدار Q افزایش می‌یابد. برای تعیین مقدار ضریب تخفیف، می‌توان از اطلاعات موجود در مسئله و هدف اصلی الگوریتم استفاده کرد. اگر هدف بیشتر به بهبود عملکرد در آینده باشد، ضریب تخفیف باید بیشتر باشد، چرا که پاداش‌های آینده بیشتری باید مد نظر باشند. تعیین مقدار مناسب برای ضریب تنزیل، با توجه به ویژگی‌های مسئله و هدف الگوریتم، به‌عنوان یک فرضیه، از روش‌های تجربی و تحلیلی استفاده می‌شود و در این مساله به کمک الگوریتم PSO به روزرسانی می‌گردد.
- **تعداد قدم‌های آموزش**: این پارامتر نشان می‌دهد که عامل باید در هر دور آموزش چند قدم بردارد. در مسئله تخصیص‌دهی وظائف در محیط مه، تعداد گام‌های آموزش بین ۱۰۰ تا ۱۰۰۰ تنظیم می‌گردد.

[1] Mean Square Error (MSE)

- **اندازه حافظه بافر**: هدف از بکارگیری این بخش، یادگیری از تجربیات گذشته می‌باشد. به همین منظور، یک زیر مجموعه تصادفی از حافظه (mini batch) انتخاب شده که برای محاسبه مقدار تابع هدف و به روزرسانی ضرائب پیش‌بینی مدل بکار می‌رود. اندازه دسته (batch) نشان می‌دهد که در هر دور آموزش، چند تجربه باید برای به‌روزرسانی تابع Q استفاده شود. در مسئله تخصیص‌دهی وظائف در محیط مه، اندازه دسته برابر ۳۲ تنظیم شده است.
- **نرخ کاوش**: این پارامتر (Epsilon) نشان می‌دهد که در هر مرحله از آموزش، عامل به چه احتمالی باید عمل تصادفی انجام دهد. این پارامتر باید در ابتدای آموزش بالا باشد تا عامل بتواند به طور کامل فضای عمل را بررسی کند. در ادامه، مقدار این پارامتر باید کاهش یابد تا عامل بتواند به سرعت به یک حالت بهینه برسد. در مسئله تخصیص‌دهی وظائف در محیط مه، مقدار احتمالی برای انتخاب عمل در بازه ۰,۹تا ۱,۰ تنظیم شده است. در این شرایط، انتخاب عمل بعدی بر اساس وضعیت فعلی صورت گرفته و بر اساس حداکثر پاداش ممکن در درازمدت این به روز رسانی محقق می‌گردد.
- **تابع پاداش**: این پارامتر نشان می‌دهد که چگونه پاداش برای هر حالت و عمل محاسبه شود. در مسئله تخصیص منابع در محیط فاگ، می‌توان تابع پاداش را بر اساس معیارهایی مانند تعداد منابع اختصاص داده شده به هر فرآیند، میزان کیفیت سرویس ارائه شده توسط هر فرآیند و هزینه منابع اختصاص داده شده توسط هر فرآیند تعریف نمود. در واقع بعد از انجام هر عمل، حالت بدست آمده (state)، مقدار پاداش (reward)، عمل انجام شده (action) و علامت نشان دهنده به اتمام رسیدن اپیزود ذخیره می گردد تا در مراحل بعدی استفاده گردد. با توجه به این دسته از پارامترها، می‌توان الگوریتم Q-Learning دوگانه را برای مسئله تخصیص‌دهی منابع در محیط مه پیاده‌سازی نمود.

به‌طور کلی، الگوریتم Q-Learning دوگانه شامل دو تابع Q و S می‌باشد. تابع Q برای محاسبه ارزش عمل در حالت‌های مختلف و تابع S برای انتخاب عمل براساس ارزش عمل در حالت‌های مختلف استفاده می‌شود. الگوریتم Q-Learning دوگانه از طریق مراحل زیر می‌تواند پیاده‌سازی شود:

- **مقداردهی اولیه**: ابتدا باید مقادیر اولیه برای تابع Q مشخص شوند. می‌توان این مقادیر را به صورت تصادفی یا با استفاده از یک روش مشخص مانند مقداردهی صفر اولیه تنظیم نمود.
- **تعریف تابع پاداش**: در ادامه لازم است تابع پاداش برای مسئله تخصیص‌دهی تعریف گردد. برای مثال، تابع پاداش ممکن است به صورت زیر تعریف نمود: برای هر فرآیندی که منابع به آن اختصاص داده می‌شود، یک پاداش مثبت

برابر با کیفیت سرویس ارائه شده توسط آن فرآیند منتسب شود و برای هر فرآیندی که به آن منابع اختصاص داده نمی‌شود، یک پاداش منفی برابر با هزینه صورت گرفته لحاظ گردد.

در مسائلی با تنوع بالا، بکارگیری رویکرد تخمین مقادیر هایپرپارامترها در فرایند یادگیری عامل می‌تواند نقش تعیین‌کننده‌ای داشته باشد زیرا شرایطی پیش می‌آید که بر اساس تکنیک‌های مبتنی بر گرادیان، خط‌مشی بهینه براحتی قابل شناسایی نیست و به کمک تقریب‌زنی ضرائب می‌توان روند شناسایی خط مشی مناسب را تسریع بخشید. برای پیاده‌سازی رویکرد پیشنهادی از نرم‌افزارهای MATLAB و Python استفاده شده و از کتابخانه‌های NumPy (انجام محاسبات عددی)، TensorFlow (یادگیری زبان ماشین و آموزش مدل‌های یادگیری عمیق)، Pandas (تحلیل داده‌ها) و الگوریتم Adam از ماژول Keras نیز برای به روزرسانی اوزان شبکه عصبی نیز بهره گرفته شده است. مشخصات شبکه عصبی مورد استفاده در جدول (۶) بیان شده است.

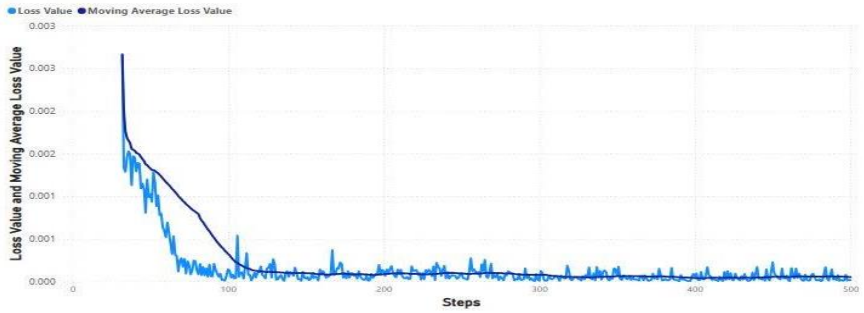
جدول ۶: مشخصات شبکه عصبی و هایپرپارامترهای مورد استفاده

Neural Network	Layer	No. of neurons	Activation Function
	Input	24	ReLU
	Hidden	24	ReLU
	Output	5	Linear
Hyper Parameters	Learning rate	0.0001 – 0.001	
	Discount rate	0.9 – 0.99	
	Epsilon	0.9 – 1.0	
	Epsilon decay	0.95 – 0.99	
No. of steps for Episode		5	

۴-۲- نتایج شبیه‌سازی

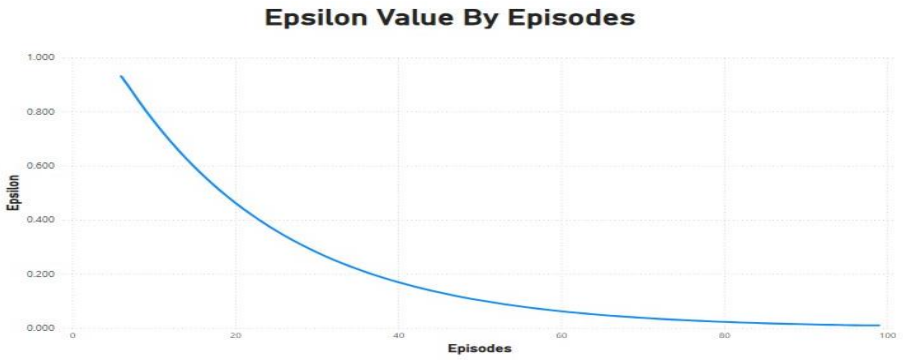
همانطور که از شکل (۱) مشخص است، مقدار تابع خطا ($Loss$) با گذشت زمان دارای یک روند نزولی بوده که نشان‌دهنده موفقیت آمیز بودن عملیات اموزش الگوریتم است. به بیان دیگر هرچه مقدار تابع خطا کمتر شود، یادگیری مدل دقیق‌تر انجام پذیرفته است. این بدان معناست که الگوریتم قادر است ارتباط بین ورودی‌ها و خروجی‌های مورد انتظار را به خوبی درک کند. کاهش میانگین تابع خطا به میزان ۰,۰۰۰۵ در هر مرحله، نمایانگر پیشرفت مستمر مدل در فرآیند یادگیری بوده و تأیید می‌کند که شبکه به‌طور پیوسته به سمت بهبود در حال پیشروی است. همچنین با توجه به

شکل (۲)، حالات تصادفی میرایی اپسیلون نیز رفته رفته کاهش یافته که نشان دهنده افت فعالیت کاوش در فضاهای جدید و افزایش یادگیری از محیط موجود بوده و این امر باعث همگرایی اپسیلون نیز می‌گردد.



شکل ۱: مقدار تابع خطا و میانگین متحرک آن (۵۰۰ گام)

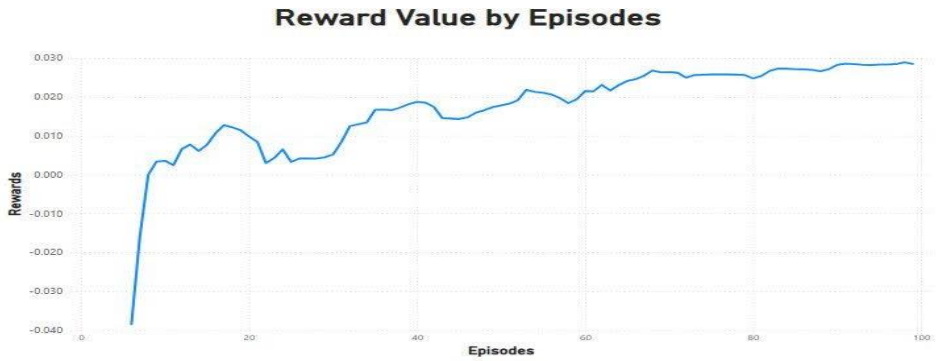
در واقع روند به ثبات رسیدن اپسیلون نیز به عنوان یکی دیگر از معیارهای همگرایی الگوریتم قلمداد می گردد که نشان می دهد شبکه به سمت یادگیری از تجربیات گذشته هدایت شده است. باید توجه داشت که کاهش نرخ کاوش و میرای آن به صورت همزمان از اهمیت بالایی برخوردار است. نرخ کاوش (Epsilon) نمایانگر میزان تصادفی بودن انتخاب‌ها است و کاهش تدریجی آن موجب می‌شود که مدل بیشتر از تجارب گذشته‌اش آموخته و به تصمیم‌گیری‌های مناسب‌تری دست یابد.



شکل ۲: میزان تغییرات نرخ کاوش (۱۰۰ اپیزود)

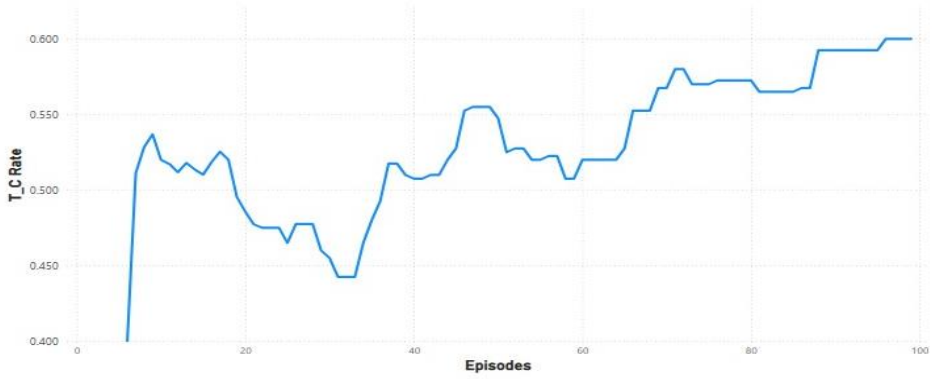
برای هر ماشین مجازی مقدار CPU و مقدار حافظه از قبل در دسترس می‌باشد. در زمان انتخاب ماشین‌های مجازی دو حالت ممکن است پیش آید. اگر منابع در دسترس برای پردازش درخواست‌ها کافی باشند، مجموع CPU و حافظه به پاداش اضافه شده و درخواست ارسال شده با موفقیت انجام می‌گیرد. اما اگر منابع کافی نباشد، میزان CPU و حافظه از پاداش کلی کسر می‌گردد و بدین ترتیب تابع پاداش به روز رسانی می‌شود. در نهایت پاداش کلی و تعداد درخواست‌های انجام شده ثبت می‌گردد. در ادامه نتایج بدست آمده برای پاداش تجمعی حاصل از ۱۰۰ اپیزود اول در

شکل (۳) نمایش داده شده است. محور افقی نمایانگر تعداد دوره‌ها و محور عمودی میزان پاداش‌ها را مشخص می‌کند. در اپیزودهای اولیه (۰ تا ۲۰) کاهش مقادیر پاداش‌ها بیانگر این است که الگوریتم هنوز در فرآیند یادگیری قرار دارد. این نوسانات اولیه نشان‌دهنده چالش‌های موجود در شناسایی استراتژی‌ها برای تخصیص‌دهی منابع است. در اپیزودهای ۲۰ تا ۶۰، افزایش تدریجی پاداش‌ها مشاهده می‌شود که نشان‌دهنده پیشرفت عملکرد الگوریتم در تخصیص منابع و بهبود کارآیی آن است. این تغییر مثبت به معنای بهبود توانایی الگوریتم در تطبیق با شرایط متغیر محیطی است. در مرحله نهایی (اپیزودهای ۶۰ تا ۱۰۰)، پاداش‌ها به طور نسبی پایدار شده و الگوریتم به حالت بهینه خود نزدیک می‌شود. این ثبات و افزایش تدریجی پاداش‌ها نمایانگر بهبود عملکرد سیستم در تخصیص منابع و توانایی آن در مدیریت درخواست‌ها است. به بیان دیگر الگوریتم با یادگیری از تجربیات گذشته و تطبیق با شرایط جدید، روند آموزشی خود را بهبود بخشیده و قادر است پس از کسب آموزش لازم، راهکارهای مؤثری را ارائه دهد.



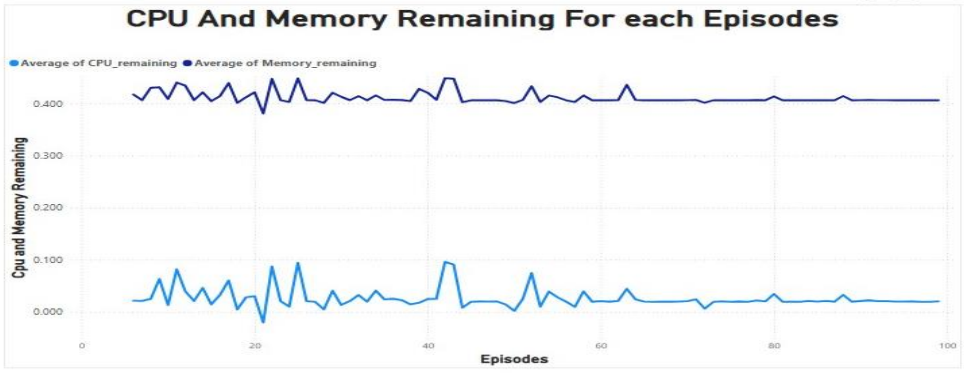
شکل ۳: مقدار پاداش تجمعی در ۱۰۰ اپیزود اول

در گام بعدی، نرخ تکمیل درخواست‌ها مورد بررسی قرار گرفته که معیاری برای سطح کارآیی تخصیص‌دهی قلمداد می گردد. همانطور که از شکل (۴) مشخص است، این نرخ به صورت پیوسته افزایش یافته که نشان دهنده افزایش توانایی مدل در پردازش درخواست های دریافت شده می باشد. با اینکه در مراحل ابتدایی آموزش، نرخ تکمیل درخواست‌ها بین ۴۰٪ تا ۶۰٪ نوسان داشته که مبین نیازمندی مدل به یادگیری بیشتر است. با افزایش سطح آموزش، این نرخ به تدریج به ۱۰۰٪ نزدیک می‌شود. روند صعودی دیده شده در این شرایط، نشان‌دهنده بهره‌برداری مفید مدل از تجربیات گذشته و بهبود در تصمیم‌گیری‌ها است. میزان ظرفیت CPU و حافظه باقی مانده در هر اپیزود نیز در شکل (۵) به تصویر کشیده شده است.

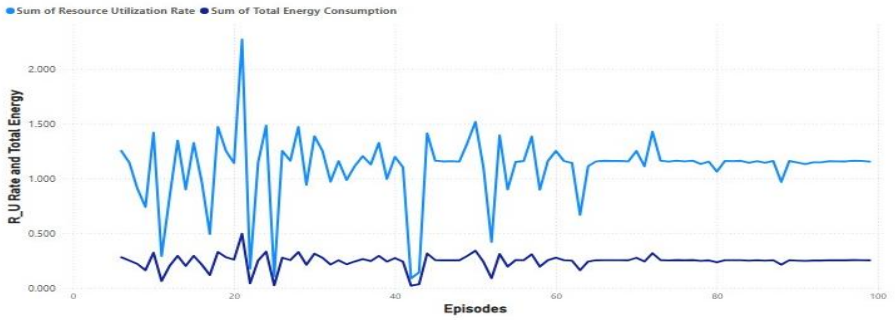


شکل ۴: ارزیابی نرخ تکمیل پردازش درخواست ها در هر مرحله

در ادامه نرخ استفاده از منابع و میزان مصرف انرژی کل مورد بررسی قرار می گیرد. نرخ استفاده از منابع (R_U Rate) نشان‌دهنده سطح کارایی تخصیص‌دهی منابع در مراحل مختلف است. با اینکه در برخی اپیزودها مقدار این نرخ به‌طور قابل توجهی افزایش یافته ولی در شرایطی نوساناتی را نیز تجربه کرده که به تغییرات شرایط کاری مربوط می‌شود. به ثبات رسیدن این شاخص مبین بهبود نرخ استفاده از ماشین‌های مجازی در دسترس است. معیار مهم دیگر، میزان انرژی مصرف شده است. تغییرات مقدار این شاخص نیز همزمان با تغییرات نرخ بکارگیری منابع نوسان داشته که در مراحل نهایی به ثبات می رسد. نتایج بدست آمده در این شرایط در شکل (۶) ارائه شده است. مقایسه دو سری داده نشان می‌دهد که در برخی اپیزودها، افزایش نرخ استفاده از منابع با افزایش مصرف انرژی همراه بوده است. برای مثال، در اپیزودهای ۲۰ تا ۳۰، افزایش نرخ استفاده از منابع منجر به افزایش مصرف انرژی شده است. نتایج بدست آمده برای معیارهای مهم در جدول (۷) بیان شده است. کاهش نوسانات دیده شده در نتایج و نزدیک شدن به یک روند ثابت در اپیزودهای نهایی، نشانه ثبات مدل طراحی شده و عملکرد مناسب رویکرد است. متوسط زمان پاسخ دهی برای ۵ نوع ماشین مجازی مختلف در شکل (۷) به نمایش درآمده است.



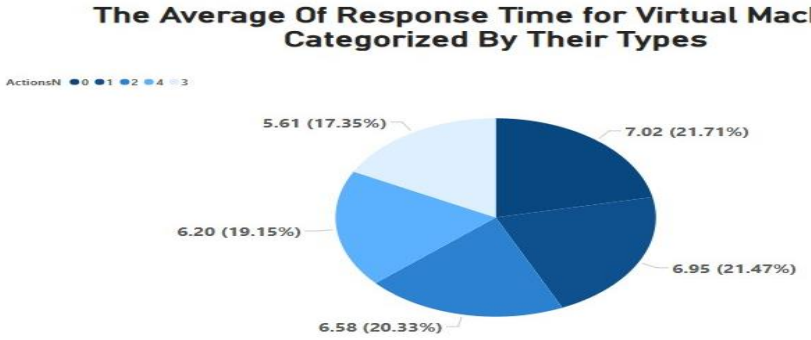
شکل ۵: مقدار ظرفیت CPU و حافظه باقی‌مانده آزاد در هر اپیزود



شکل ۶: ارزیابی نرخ انرژی مصرفی و بکارگیری منابع و در ۱۰۰ اپیزود اول

جدول ۷: مقادیر متوسط شاخص‌های مهم

Index	Value
Avg. of Make span	19.97
Avg. of Response time	6.69
Avg. of Total Energy	0.0493
Avg. of Rewards	0.0192

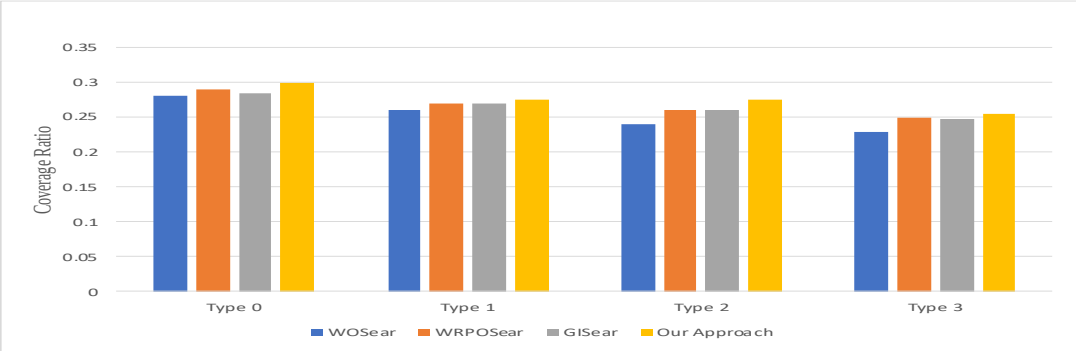


شکل ۷: متوسط زمان پاسخ‌دهی ماشین‌های مجازی

۴-۳- مقایسه نتایج

به منظور مقایسه سطح کارآیی رویکرد و سنجش کیفیت روش تخصیص‌دهی طراحی شده، از داده‌های ارائه شده در [۳۰] بهره گرفته شده است. این مجموعه داده متشکل از ۵۰۰۰۰ درخواست متشکل از ۴ نوع مختلف در یک بازه زمانی ۱۰ هفته‌ای در منطقه Ivory Coast می‌باشد. در راستای دست‌یابی به نتایج عملیاتی، تعداد درخواست‌ها به ۲۵۰

عدد کاهش یافته و از جنبه‌های نرخ پوشش‌دهی و زمان اجرا با روش‌های سنتی مقایسه شده است. نتایج بدست آمده در این شرایط، در شکل (۸) ارائه شده است.



شکل ۸: مقایسه میزان نرخ پوشش‌دهی در روش‌های گوناگون

۵- نتیجه‌گیری

استفاده از سامانه فضای ابری برای ساختارهایی که از تکنولوژی IoT بهره می‌برند، بدلیل حجم بالای درخواست‌ها و تنوع آنها به صورت لحظه‌ای امکان‌پذیر نمی‌باشد. در این شرایط بهره‌گیری از ماشین‌های مجازی قرارگرفته در لبه شبکه می‌تواند راهگشا باشد. با این حال، منابع موجود در گره‌های مه دارای ظرفیت محدودی بوده و بکارگیری آنها نیازمند اجرای یک سیستم کنترل یکپارچه و مقیاس‌پذیری است. لذا هدف اصلی در این مقاله، تخصیص‌دهی وظائف درخواست شده به ماشین‌های مجازی موجود و در دسترس با درنظرگرفتن محدودیت‌های حاکم بر شبکه بوده است. در این مقاله از الگوریتم یادگیری تقویتی عمیق مبتنی بر شبکه q دوگانه (DDQN) بهره گرفته شده که در هر مرحله هایپرپارامترهای آن به کمک الگوریتم ازدحام ذرات (PSO) به روزرسانی شده‌اند. برای این منظور از مجموعه داده Borg معرفی شده توسط شرکت گوگل در سال ۲۰۱۹ بهره گرفته شده است. با توجه به نتایج بدست آمده، میانگین کاهش مقدارتابع خطا به میزان ۰,۰۰۰۵ در هر مرحله، افزایش نرخ تکمیل پردازش درخواست‌ها، ایجاد ثبات در میزان مصرف انرژی، کاهش بازه زمانی تخصیص درخواست‌ها به ماشین‌های مجازی، کاهش نرخ کاوش و بکارگیری منابع به صورت متعادل از جمله دستاوردهای رویکرد طراحی شده است.

جدول ۱: فهرست پارامترها

γ : ضریب تنزیل

$\pi(a|s)$: برنامه‌ریز خط مشی

r_t : پاداش لحظه‌ای

$V^{\pi}(s)$: تابع حالت – ارزش عامل

S : فضای حالت

A : فضای عمل

P : ماتریس توزیع گذر حالت

r : سیگنال پاداش

P^{*}_{ijk} : احتمال تکمیل وظیفه t_i توسط ماشین مجازی v_j

R^{j} : پیشینه وظائف تخصیص داده شده به ماشین مجازی j ام

α : نرخ یادگیری

$N^{i}_{k,}$: عدد باینری معرف تکمیل وظیفه t_i در سیکل زمانی k

T_i : آخرین مهلت زمانی جهت تکمیل درخواست t_i

$T^{i}_{k,}$: مدت زمان پردازش درخواست t_i در سیکل زمانی k

δ_i : سطح اولویت درخواست t_i

T^{loc}_{nm} : زمان لازم برای پردازش درخواستm ام توسط ماشین مجازی n ام

x_{nm} : متغیر باینری معرف تایید/رد درخواستmام توسط ماشین n ام

P^{k}_{i} : بهترین موقعیت ذره i ام در گام k ام

G^{k} : بهترین موقعیت در بین تمام ذرات در گام k ام

ω : ضریب اینرسی

r_1 ، r_2 : مقادیر تصادفی با توزیع نرمال

c_2 ، c_1 : ضرائب شتاب

منابع

[1] Zheng, Haotian, Kangming Xu, Mingxuan Zhang, Hao Tan, and Hanzhe Li. "Efficient resource allocation in cloud computing environments using AI-driven predictive analytics." Applied and Computational Engineering 82 (2024): 17-23.

[2] Lu, An-qi, and Jing-hua Zhu. "Worker recruitment with cost and time constraints in mobile crowd sensing." Future Generation Computer Systems 112 (2020): 819-831.

[3] Zhang, Yifan, Bo Liu, Yulu Gong, Jiaxin Huang, Jingyu Xu, and Weixiang Wan. "Application of machine learning optimization in cloud computing resource scheduling and

management." In Proceedings of the 5th International Conference on Computer Information and Big Data Applications, pp. 171-175. 2024.

[4] Alizadeh Javaheri, S.D., Ghaemi, R. & Monshizadeh Naeen, H. An autonomous architecture based on reinforcement deep neural network for resource allocation in cloud computing. Computing 106, 371–403 (2024). https://doi.org/10.1007/s00607-023-01220-7

[5] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[6] Vahedi, Zohreh, Seyyed Javad Mahdavi Chabok, and Gelareh Veisi. "Heterogeneous task allocation in mobile crowd sensing using a modified approximate policy approach." International Journal of Nonlinear Analysis and Applications 15, no. 4 (2024): 251-264.

[7] Cheng, Yunli, A. Vijayaraj, Kiran Sree Pokkuluri, Taybeh Salehnia, Ahmadreza Montazerolghaem, and Roqia Rateb. "Vehicular fog resource allocation approach for VANETs based on deep adaptive reinforcement learning combined with heuristic information." IEEE Access (2024).

[8] Zhang, Daqing, Haoyi Xiong, Leye Wang, and Guanling Chen. "CrowdRecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint." In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 703-714. 2014.

[9] Jie, Yingmo, Mingchu Li, Cheng Guo, and Ling Chen. "Game-theoretic online resource allocation scheme on fog computing for mobile multimedia users." China Communications 16, no. 3 (2019): 22-31.

[10] Zolghadri, Mohammad, Parvaneh Asghari, Seyed Ebrahim Dashti, and Alireza Hedayati. "Resource allocation in fog–cloud environments: state of the art." Journal of Network and Computer Applications 227 (2024): 103891.

[11] Afzali, Mahboubeh, Amin Mohammad Vali Samani, and Hamid Reza Naji. "An efficient resource allocation of IoT requests in hybrid fog–cloud environment." The Journal of Supercomputing 80, no. 4 (2024): 4600-4624.

[12] Nassar, Almuthanna, and Yasin Yilmaz. "Reinforcement learning for adaptive resource allocation in fog RAN for IoT with heterogeneous latency requirements." IEEE Access 7 (2019): 128014-128025.

[13] Subbaraj, Saroja, Revathi Thiyagarajan, and Madavan Rengaraj. "A smart fog computing based real-time secure resource allocation and scheduling strategy using multi-objective crow search algorithm." Journal of Ambient Intelligence and Humanized Computing 14, no. 2 (2023): 1003-1015.

[14] Fan, Qiang, Jianan Bai, Hongxia Zhang, Yang Yi, and Lingjia Liu. "Delay-aware resource allocation in fog-assisted IoT networks through reinforcement learning." IEEE Internet of Things Journal 9, no. 7 (2021): 5189-5199.

[15] Leontiou, Nikolaos, Dimitrios Dechouniotis, Spyros Denazis, and Symeon Papavassiliou. "A hierarchical control framework of load balancing and resource allocation of cloud computing services." Computers & Electrical Engineering 67 (2018): 235-251.

[16] Huang, Liang, Xu Feng, Cheng Zhang, Liping Qian, and Yuan Wu. "Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing." Digital Communications and Networks 5, no. 1 (2019): 10-17

[17] Sundar, Sowndarya, and Ben Liang. "Offloading dependent tasks with communication delay and deadline constraint." In IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 37-45. IEEE, 2018.

[18] Hammoud, Ahmad, Azzam Mourad, Hadi Otrok, Omar Abdel Wahab, and Haidar Harmanani. "Cloud federation formation using genetic and evolutionary game theoretical models." Future Generation Computer Systems 104 (2020): 92-104.

[19] Sheng, Shuran, Peng Chen, Zhimin Chen, Lenan Wu, and Yuxuan Yao. "Deep reinforcement learning-based task scheduling in iot edge computing." Sensors 21, no. 5 (2021): 1666.

[20] Song, Zheng, Chi Harold Liu, Jie Wu, Jian Ma, and Wendong Wang. "QoI-aware multitask-oriented dynamic participant selection with budget constraints." IEEE Transactions on Vehicular Technology 63, no. 9 (2014): 4618-4632.

[21] Zhu, Xiaoyu, Yueyi Luo, Anfeng Liu, Wenjuan Tang, and Md Zakirul Alam Bhuiyan. "A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility." IEEE Transactions on Intelligent Transportation Systems (2020).

[22] Li, Hanshang, Ting Li, and Yu Wang. "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks." In 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems, pp. 136-144. IEEE, 2015.

[23] Kalui, Dorothy Mwongeli, Dezheng Zhang, Geoffrey Muchiri Muketha, and Jared Okoyo Onsomu. "Simulation of trust-based mechanism for enhancing user confidence in mobile crowdsensing systems." IEEE Access 8 (2020): 20870-20883.

[24] Wu, Chu-ge, Wei Li, Ling Wang, and Albert Y. Zomaya. "Hybrid evolutionary scheduling for energy-efficient fog-enhanced internet of things." IEEE Transactions on Cloud Computing 9, no. 2 (2018): 641-653.

[25] Ding, Xuyang, Ruizhao Lv, Xiaoyi Pang, Jiahui Hu, Zhibo Wang, Xu Yang, and Xiong Li. "Privacy-preserving task allocation for edge computing-based mobile crowdsensing." Computers & Electrical Engineering 97 (2022): 107528.

[26] Buhussain, A. A., R. E. D. Grande, and A. Boukerche. "Performance analysis of Bio-Inspired scheduling algorithms for cloud." In IEEE International parallel and distributed processing symposium workshops, pp. 776-785. 2016.

[27] Verma, Abhishek, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. "Large-scale cluster management at Google with Borg." In Proceedings of the tenth european conference on computer systems, pp. 1-17. 2015.

[28] Tirmazi, Muhammad, Adam Barker, Nan Deng, Md E. Haque, Zhijing Gene Qin, Steven Hand, Mor Harchol-Balter, and John Wilkes. "Borg: the next generation." In Proceedings of the fifteenth European conference on computer systems, pp. 1-14. 2020.

[29] Chen, Yanpei, Archana Sulochana Ganapathi, Rean Griffith, and Randy H. Katz. "Analysis and lessons from a publicly available google cluster trace." EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-95 94 (2010).

[30] Wang, Liang, Zhiwen Yu, Daqing Zhang, Bin Guo, and Chi Harold Liu. "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation." IEEE Transactions on Mobile Computing 18, no. 1 (2018): 84-97.