علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۲۴، شماره ۱، ۱۴۰۵
صفحات ۱۲-۲۱
**مقاله پژوهشی**

# برون‌سپاری آگاه از تأخیر وظایف در رایانش مه خودرویی

محمد پورعاشوری[۱]، محسن انصاری[۲*]

*نویسنده مسئول، دریافت: ۱۴۰۴/۰۳/۰۵، بازنگری: ۱۴۰۴/۰۵/۰۱، پذیرش: ۱۴۰۴/۰۶/۲۰

[۱] مهندسی کامپیوتر، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران
[۲] استادیار، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

## چکیده

زمان اجرای وظایف مختلف در اینترنت اشیاء همواره به عنوان یک مسئله مهم و تعیین‌کننده مطرح است. از این رو تأخیر اجرای وظایف در این ساختار از اهمیت بالایی برخوردار هستند. در همین راستا، رایانش مه به عنوان یک روش مناسب برای نزدیک‌تر کردن مراکز پردازش داده به لبه و کاهش تأخیر معرفی گردید. ازجمله کاربردهای رایانش مه می‌توان به رایانش مه خودرویی اشاره کرد که با هدف کمک‌رسانی به محدودیت‌های اجرایی خودروهای متصل به این ساختار و عدم آگاهی کامل محیطی خودروها مورد استفاده قرار می‌گیرد. همچنین با گسترش شبکه‌های ارتباطی و معرفی فناوری G5 به عنوان یک روش ارتباطی مناسب، بخشی از چالش‌های مرتبط با تأخیر ارتباطی رفع گردیده‌است. با این حال، پیاده‌سازی گسترده رایانش مه خودرویی همچنان با چالش‌های مختلفی مواجه است. از جمله این چالش‌ها می‌توان به افزایش فاصله خودروها از محل اجرای وظایف برون‌سپاری‌شده و مواجه‌شدن با پدیده مهاجرت وظایف اشاره کرد. در این پژوهش، یک روش برون‌سپاری وظایف مبتنی بر یادگیری تقویتی چندعاملی عمیق پیشنهاد شده است که با در نظر گرفتن خصوصیات ویژه هر وظیفه، به مدیریت بهینه مهاجرت وظایف و کاهش تأخیر در محیط پویا و متحرک خودرویی می‌پردازد. نوآوری اصلی روش پیشنهادی، طراحی مدل ارتباطی جدید برای مدیریت پدیده مهاجرت و ارائه یک معماری غیرمتمرکز است که امکان یادگیری مستقل و موازی عامل‌ها را فراهم می‌کند. نتایج شبیه‌سازی انجام شده نشان می‌دهد که روش پیشنهادی نسبت به روش‌های جدید تا ۱۲ درصد کاهش تأخیر را به دنبال دارد.

**کلمات کلیدی:** اینترنت اشیاء، رایانش مه، رایانش مه خودرویی، برون‌سپاری وظایف

## ۱- مقدمه

با گسترش اینترنت اشیا[۱] (IoT) و فناوری‌های ارتباطی، هم تقاضای رایانش و هم نرخ داده به‌صورت نمایی افزایش می‌یابد که یک مسئله حیاتی محسوب می‌شود [۱][۲][۳][۴]. کاربردهای نوظهور فناوری 5G مانند سامانه‌های اطلاعات و سرگرمی، بازی‌های تعاملی، ردیابی ماهواره‌ای، بلاکچین[۲] و پردازش زبان طبیعی، این مسئله را به‌وضوح نشان می‌دهند [۵]. تضمین الزامات تأخیر به دلیل فاصله طولانی بین مراکز داده و گره‌های مشتری[۳] (CNs) دشوار است و این چالش جدیدی برای الگوی رایانش ابری پایه ایجاد می‌کند. برای حل این مسئله، الگوی جدیدی به نام رایانش مه[۴] معرفی شده که امکانات مبتنی بر ابر را برای گره‌های مشتری در شبکه‌های لبه ارائه می‌دهد [۶][۷][۸]. با این حال، برای پوشش یک منطقه جغرافیایی وسیع، نیاز به استقرار تعداد زیادی سرویس‌دهنده وجود دارد که هزینه‌های نگهداری و مصرف انرژی را افزایش می‌دهد. برای رفع این مشکل، یک راهکار استفاده از منابع رایانش بلااستفاده در خودروهای نزدیک یعنی گره‌های خودرویی ۵ (VNs) است. خودروهای هوشمند با امکانات ارتباطی اختصاصی و رایانه‌های پرسرعت داخلی به عنوان گره‌های خودرویی شناخته می‌شوند. بنابراین، یک گروه بزرگ از خودروهای نزدیک می‌توانند در زمان اوج بار کاری، بدون نیاز به استقرار گره‌های رایانشی اضافی، مقدار عظیمی از منابع رایانش را فراهم‌کنند.

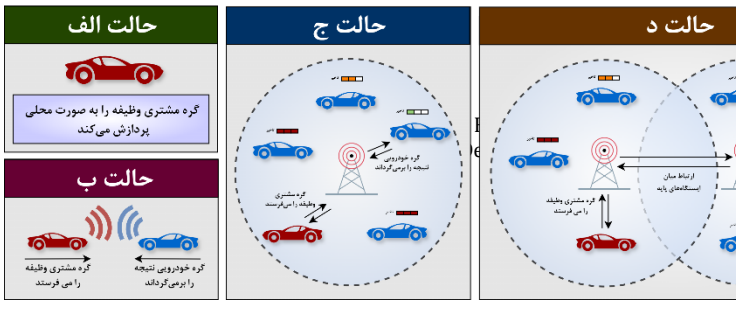
---

۱ Internet of Things
۲ Blockchain
۳ Client Nodes
۴ Fog Computing
۵ Vehicular Nodes

تقاضای رایانش گره‌های مشتری می‌تواند به گره‌های خودرویی منتقل شود تا تأخیر خدمات به حداقل برسد. این الگوی جدید رایانش که رایانش خودرویی و رایانش مه را ترکیب می‌کند، با نام رایانش مه خودرویی[۱] (VFC) شناخته می‌شود [۹][۱۰]. با این حال، علیرغم مزایای VFC، پیاده‌سازی گسترده آن هنوز با چندین چالش حیاتی روبرو است. برخی مطالعات پیشین ایده انتقال وظایف گره‌های مشتری را بررسی کرده‌اند. توسعه یک سیاست مؤثر برای انتخاب وظایف گره‌های مشتری که به گره‌های خودرویی منتقل شوند، مسئله‌ای مهم است. با وجود ویژگی‌های مختلف وظایف (مانند وظایف نیازمند پردازش سنگین و وظایف حساس به تأخیر)، چالش حیاتی این است که چگونه منابع رایانش گره‌های خودرویی را به وظایف گره‌های مشتری اختصاص دهیم تا تأخیر خدمات کاهش یابد. از آنجایی که ویژگی‌های مربوط به هر کدام از وظایف گره‌های مشتری متفاوت است، هر وظیفه به منابع رایانش و تأخیر خدمات مطابق با ویژگی‌های خود نیاز دارد. همچنین، این روش باید نیازهای رایانش وظایف و دسترس‌پذیری منابع رایانش در میان گره‌های خودرویی مختلف را در نظر بگیرد. در نتیجه، یک الگوریتم مؤثر برای برون‌سپاری وظایف لازم است که تأخیر خدمات را به حداقل برساند. چالش دیگر زمانی ایجاد می‌شود که گره‌های خودرویی یا مشتری از منطقه سرویس قبلی خود خارج شوند و دیگر در یک منطقه سرویس مشترک نباشند. این امر مشکلات جدیدی در ارتباط بین گره‌ها ایجاد می‌کند. راهکارهای مختلفی برای این وضعیت ارائه شده است، مانند مهاجرت وظایف، که باید با یک مدل ارتباطی مناسب مدیریت شود تا از ایجاد مشکل جلوگیری شود. با توجه به این چالش‌ها، در این پژوهش یک رویکرد مبتنی بر الگوریتم یادگیری تقویتی با نام شبکه Q عمیق[۲] (DQN) ارائه می‌گردد [۱۱][۱۲]. با استفاده از این رویکرد، یک روش مناسب برای برون‌سپاری وظایف ارائه می‌شود که مشکل تأخیر خدمات را حل می‌کند. استفاده از DQN در محیط رایانش مه خودرویی مزایای متعددی دارد که آن را به گزینه‌ای مناسب برای این سناریوی پویا و پیچیده تبدیل می‌کند. در این محیط‌ها که گره‌های مشتری و خودرویی به‌طور مداوم در حال حرکت هستند، تصمیم‌گیری درباره برون‌سپاری وظایف و مدیریت شبکه وابستگی زیادی به زمان دارد. قابلیت تطبیق با محیط‌های پویا، که شرایط و حالات آن‌ها به‌دلیل عواملی مانند الگوهای ترافیکی و تحرک گره‌ها در طول زمان تغییر می‌کند، موجب برتری روش DQN نسبت به دیگر الگوریتم‌های یادگیری تقویتی می‌گردد. معماری MADQN به کار رفته در این پژوهش از نوع چند عاملی و غیرمتمرکز بوده که هر عامل به‌طور مستقل و موازی یادگیری و تصمیم‌گیری را انجام می‌دهد. مزیت اصلی این روش نسبت به روش‌های بازیگر-منتقد چندعاملی[۳] اخیر مانند MADDPG، کاهش پیچیدگی محاسباتی و افزایش مقیاس‌پذیری به دلیل حذف نیاز به ارتباطات مکرر بین عامل‌ها در حین یادگیری است. با وجود اینکه الگوریتم‌های یادگیری تقویتی عمیق معمولاً هزینه رایانش بالاتری نسبت به الگوریتم‌های سبک‌تر مانند هیوریستیک‌ها دارند، اما در محیط‌های بسیار پویا و پیچیده مانند محاسبات مه خودرویی، این افزایش هزینه با مزیت‌هایی نظیر انعطاف‌پذیری، قابلیت تطبیق‌پذیری سریع با شرایط جدید و دقت بالاتر در تصمیم‌گیری‌ها جبران می‌شود. همچنین باید تأکید شود که مرحله یادگیری عامل‌ها به صورت آفلاین و پیش از استفاده عملی انجام می‌شود. بنابراین، زمان اجرا در فاز بهره‌برداری به‌شدت کاهش می‌یابد و عامل‌ها می‌توانند در به طور آنی و به سرعت تصمیم‌گیری نمایند. در نتیجه، استفاده از این روش برای وظایف حساس به تأخیر در این محیط پویا توجیه‌پذیر است، زیرا ضمن حفظ کیفیت و بهینگی تصمیم‌ها، تأخیر عملیاتی را در فاز اجرا به حداقل می‌رساند. این رویکرد همچنین تصمیم‌گیری در برون‌سپاری وظایف را با پیش‌بینی مؤثر حالت‌های آینده بر اساس داده‌های پیشین بهبود می‌بخشد، که منجر به کاهش تأخیر و استفاده بهینه‌تر از منابع می‌شود.

شکل ۱ یک مثال از عملکرد روش پیشنهادی بوده که به درک بهتر مسئله حل شده کمک می‌کند. مطابق شکل، پنج حالت مختلف نشان داده شده‌اند که در هر کدام، شرایط ارتباطی متفاوت است و وضعیت گره‌های خودرویی از نظر تحرک و تأخیر متغیر است. هیچکدام از کارهای پیشین چنین سطح کاملی از تنوع گزینه‌های پردازشی ارائه‌شده در روش ما را ارائه‌نداده‌است. در حالت الف، گره مشتری با استفاده از ظرفیت پردازشی خود، وظیفه‌اش را به‌صورت محلی اجرا می‌کند. حالت ب تأثیر تحرک را در شرایطی نشان می‌دهد که گره خودرویی و گره مشتری در محدوده مستقیم ارتباطی یکدیگر قراردارند و می‌توانند مستقیماً ارتباط برقرارکنند. در حالت ج، گره مشتری می‌تواند وظیفه خود را به‌صورت بیسیم از طریق ارتباط با ایستگاه پایه به گره خودرویی ارسال کند. نتایج نیز به همین روش می‌توانند به گره مشتری بازگردانده‌شوند. در حالت د، گره خودرویی مناسب در منطقه سرویس مجاور قراردارد. گره مشتری می‌تواند وظیفه خود را از طریق ارتباط با ایستگاه پایه ارسال کند. ایستگاه پایه مربوطه با ایستگاه پایه منطقه مجاور ارتباط برقرار می‌کند. در نهایت، ایستگاه پایه مربوط به منطقه مجاور، وظیفه را به گره خودرویی ارسال می‌کند. نتایج نیز به همین روش می‌توانند بازگردانده‌شوند. در حالت ه، هیچ گره خودرویی مناسبی برای برون‌سپاری وظیفه در لایه مه در دسترس نیست. در این شرایط، وظیفه می‌تواند از طریق ایستگاه پایه به سرویس‌دهنده ابری ارسال شود.

به طور کلی، نوآوری‌های اصلی این پژوهش بدین شرح است: ۱) ارائه یک روش ارتباطی برای رفع چالش‌های ناشی از جابجایی خودروها بین مناطق سرویس؛ ۲) معرفی الگوریتم جدید که برای کارایی در چارچوب پیشنهادی طراحی شده است. این الگوریتم بر اساس DQN عمل کرده، چند عاملی بوده، و در کاهش تأخیر نسبت به روش‌های دیگر عملکرد بهتری دارد؛ ۳) ارائه نتایج بر اساس مدل تحرک مبتنی بر شرایط دنیای واقعی با استفاده از شبیه ساز SUMO [۱۳] برای روشن کردن عملکرد طرح پیشنهادی




[۱] Vehicular Fog Computing
[۲] Deep Q Network
[۳] Multi Agent

جدول ۱: مقایسه با کارهای پیشین

| پژوهش | تاخیر | لایه ابر | لایه مه | تحرک | مهاجرت | ارتباط میان مناطق سرویس | اجرای محلی |
|---|---|---|---|---|---|---|---|
| [۱۴] | ✓ | × | ✓ | محدود | حل شده | ✓ | × |
| [۱۵] | ✓ | ✓ | ✓ | کامل | × | × | ✓ |
| [۱۶] | ✓ | ✓ | ✓ | محدود | × | محدود | ✓ |
| [۱۷] | ✓ | ✓ | ✓ | محدود | × | محدود | ✓ |
| [۱۸] | ✓ | ✓ | ✓ | کامل | × | محدود | ✓ |
| [۱۹] | ✓ | ✓ | ✓ | محدود | × | محدود | کامل |
| [۲۰] | ✓ | ✓ | ✓ | محدود | حل شده | ✓ | × |
| [۲۱] | ✓ | × | ✓ | محدود | × | محدود | ✓ |
| [۲۲] | ✓ | × | ✓ | محدود | × | × | ✓ |
| [۲۳] | ✓ | ✓ | ✓ | محدود | حل شده | محدود | ✓ |
| [۲۴] | ✓ | ✓ | ✓ | کامل | حل شده | محدود | ✓ |
| [۲۵] | ✓ | × | ✓ | محدود | × | محدود | ✓ |
| [۲۶] | ✓ | ✓ | ✓ | کامل | × | ✓ | ✓ |
| [۲۷] | ✓ | × | ✓ | کامل | حل شده | × | ✓ |
| [۲۸] | ✓ | ✓ | ✓ | محدود | × | ✓ | ✓ |
| [۲۹] | ✓ | ✓ | ✓ | محدود | × | × | ✓ |
| [۳۰] | ✓ | ✓ | ✓ | محدود | حل شده | × | ✓ |
| [۳۱] | ✓ | ✓ | ✓ | محدود | حل شده | × | ✓ |
| [۳۲] | ✓ | × | ✓ | × | حل شده | محدود | × |

## ۲- بررسی کارهای پیشین

در رابطه با کارهای پیشین، مقالات متعددی به بررسی مسئله برون‌سپاری وظایف از دیدگاه‌های مختلف پرداخته‌اند و هدف آن‌ها ارائه راه‌حل‌های مؤثر برای چالش‌های مرتبط بوده است. در [۱۴] یک معماری سلسله‌مراتبی رایانش مه خودرویی پیشنهاد گردیده که به مناطق مجاور اجازه می‌دهد منابع رایانش بدون استفاده را به اشتراک بگذارند تا بهره‌وری کلی منابع را بهبود بخشند. برای حل پیچیدگی‌های مربوط به مهاجرت وظایف درون‌منطقه‌ای و بین‌منطقه‌ای، نویسندگان یک استراتژی توزیع‌شده برای مهاجرت وظایف مبتنی بر یادگیری تقویتی چندعاملی معرفی کردند. نویسندگان در [۱۵] به بهینه‌سازی مصرف انرژی و تأخیر سرویس پرداخته‌اند. یک طرح پویای برون‌سپاری رایانش و تخصیص منابع با بهره‌وری انرژی بهینه پیشنهاد شده است تا مسائل مربوط به موازنه‌ی انرژی و تأخیر و تخصیص بهینه‌ی منابع را برطرف کند. در کار [۱۶] یک قرارداد دوگانه برای تشویق به اشتراک‌گذاری منابع در شبکه‌های مه خودرویی معرفی گردیده است و در کنار آن، یک چارچوب یادگیری تقویتی عمیق توزیع‌شده را برای بهینه‌سازی مهاجرت وظایف و تخصیص منابع ارائه دادند. در [۱۷] یک استراتژی

مبتنی بر یادگیری تقویتی عمیق برای به حداقل رساندن تأخیر معرفی شد. این استراتژی با استفاده از فرآیند تصمیم‌گیری مارکوف و شبکه‌های عمیق Q، تصمیمات مربوط به مهاجرت وظایف برای خودروها را بهینه می‌کند و در عین حال محدودیت‌های تحرک و ارتباطات را در نظر می‌گیرد. در کار [۱۸] یک چارچوب یادگیری تقویتی چندعاملی مبتنی بر ترانسفورمر برای بهینه‌سازی مهاجرت وظایف و تخصیص منابع پیشنهاد شد. این روش با استفاده از مدل‌های ترانسفورمر برای پیش‌بینی دنباله‌های طولانی و تفکیک تطبیقی سیاست‌ها، به عدم قطعیت در بار سرویس دهنده‌ها پرداخته و با محیط‌های جدید سازگار می‌شود. نویسندگان در پژوهش [۱۹] یک رویکرد مبتنی بر گرادیان سیاست قطعی عمیق[۱] برای بهینه‌سازی برون‌سپاری رایانش و ذخیره‌سازی سرویس پیشنهاد داده‌اند. این چارچوب با در نظر گرفتن همکاری بین گره‌های هم‌نوع و مه به‌صورت همزمان، تأخیر پردازش وظایف و مصرف انرژی را کاهش داده و در عین حال تعادل استفاده از منابع را حفظ می‌کند.

[۱] Deep Deterministic Policy Gradient

در کار [۲۰] یک چارچوب برون‌سپاری وظیفه و تخصیص منابع معرفی شد که پیش‌بینی تحرک را در یک استراتژی بهینه‌سازی حداقل-حداکثر[۱] ادغام می‌کند و با این روش، تأخیر کلی وظایف را کاهش داده و کارایی سیستم را بهبود می‌بخشد. در پژوهش [۲۱] یک الگوریتم مهاجرت وظایف وابسته به تأخیر و انرژی معرفی شد که با استفاده از یادگیری تقویتی عمیق، فرآیند مهاجرت وظایف را بهینه‌سازی می‌کند. این الگوریتم اطلاعات ناقص محیطی و الزامات بلندمدت کیفیت سرویس[۲] (QoS)را در نظر گرفته و عملکرد سیستم را بهبود می‌بخشد. در مطالعه [۲۲] یک استراتژی مبتنی بر فرآیند تصمیم‌گیری نیمه-مارکوف برای بهینه‌سازی مهاجرت وظایف در محیط‌های خودرویی با پشتیبانی از رایانش مه خودرویی ارائه شد. این استراتژی عوامل مختلفی مانند ناهمگنی منابع وسایل نقلیه، ورود تصادفی وظایف، و پویایی وسایل نقلیه را در نظر می‌گیرد. روش پیشنهادی با تعیین تخصیص بهینه وظایف بین خودروها و منابع مربوط به رایانش مه خودرویی، تأخیر در مهاجرت وظایف را به حداقل رسانده و در عین حال، پاداش‌های بلندمدت را به حداکثر می‌رساند. در [۲۳] یک چارچوب مهاجرت وظایف با استفاده از ابزارهای داکر[۳]، کوبرنیتز[۴]، و روش بازگشت به عقب[۵] ارائه شد. این مطالعه استراتژی‌هایی را برای مهاجرت وظایف از دستگاه‌های کاربری، مانند وسایل نقلیه، به گره‌های مه مورد بررسی قرار می‌دهد. در مطالعه [۲۴] یک مدل سه‌لایه‌ای رایانش مه خودرویی برای کاهش تأخیر در محیط‌های شهری با بهینه‌سازی مهاجرت رایانش و تخصیص منابع در بین گره های مه خودرویی، گره های مه ثابت و سرویس دهنده‌های ابری پیشنهاد شد. این مطالعه نشان می‌دهد که با استفاده از الگوریتم‌های تصمیم‌گیری مهاجرت در یک منطقه و تخصیص منابع در چند منطقه، کاهش تأخیر بهتری نسبت به روش‌های موجود حاصل می‌شود.در پژوهش [۲۵] یک استراتژی مهاجرت وظایف در رایانش مه خودرویی ارائه شد که اولویت‌بندی وظایف را در نظر گرفته و با استفاده از یادگیری تقویتی عمیق برون‌سپاری وظایف را بهینه‌سازی می‌کند. در این سیستم، وظایف بر اساس فوریت آن‌ها اولویت‌بندی شده و تخصیص منابع به طور کارآمد مدیریت می‌شود. نویسندگان در مطالعه [۲۶] یک رویکرد مبتنی بر یادگیری تقویتی عمیق برای بهینه‌سازی تخصیص منابع در رایانش مه خودرویی معرفی کرده‌اند. نویسندگان یک معماری سه‌لایه‌ای مشارکتی شامل ابر، هماهنگ‌کننده مه[۶]، و لایه مه خودرویی پیشنهاد کردند تا به اشتراک‌گذاری کارآمد منابع بین وسایل نقلیه را تسهیل کند. این سیستم با استفاده از یادگیری عمیق برای پیش‌بینی ترافیک و یک الگوریتم جدید برای دسته‌بندی و تطبیق پویای وظایف، نرخ موفقیت تطبیق وظیفه-منبع را به حداکثر می‌رساند. در [۲۷] نویسندگان چارچوب جدیدی برای برون‌سپاری وظایف در رایانش مه خودرویی ارائه کردند که تأخیر و کیفیت سرویس (QoS) را بهینه‌سازی می‌کند. آن‌ها یک رویکرد پویای برون‌سپاری وظایف بر پایه بهینه‌سازی برنامه‌ریزی خطی[۷] و بهینه‌سازی ازدحام ذرات دودویی[۸] (BPSO) پیشنهاد کردند. در [۲۸] یک چارچوب تخصیص منابع برای رایانش مه خودرویی پیشنهاد شده چالش مدیریت منابع را به‌عنوان یک مسئله بهینه‌سازی با هدف کاهش تأخیر سرویس مطرح کرده است. برای دستیابی به این هدف، از الگوریتم بهینه‌سازی سیاست مجاورتی[۹] (PPO) همراه با یک الگوریتم اکتشافی استفاده کردند. در پژوهش [۲۹] یک چارچوب مهاجرت وظایف چندگانه در رایانش مه خودرویی با استفاده از رویکرد یادگیری تقویتی عمیق چندعاملی معرفی شده است. این چارچوب توزیع‌شده مبتنی بر ارتباط خودرو با خودرو برای مهاجرت وظایف پیشنهاد شده و از الگوریتم بازیگر-منتقد گیت‌دار چندعاملی[۱۰] بهره می‌برد تا استراتژی‌های کارآمد و مشارکتی مهاجرت وظایف را در یک محیط پویای خودرویی به صورت موثر یاد بگیرد.

در مطالعه [۳۰] یک استراتژی برون‌سپاری وظایف بی‌درنگ مبتنی بر یادگیری تقویتی برای سیستم‌های رایانش مه تحمل پذیر اشکال معرفی شد. نویسندگان به چالش تضمین اجرای مطمئن وظایف بی‌درنگ در محیط‌های رایانش مه می‌پردازند که خرابی گره‌ها یا لینک‌های ارتباطی می‌تواند تکمیل وظایف را به خطر بیاندازد. این استراتژی با استفاده از یادگیری تقویتی به‌طور پویا مناسب‌ترین گره‌های مه را برای اجرای وظایف اصلی و پشتیبان انتخاب می‌کند. در [۳۱] یک طرح پیشنهادی برون‌سپاری و مهاجرت وظایف آگاه از تحرک برای شبکه‌های رایانش مه ارائه شد. نویسندگان یک معماری شبکه رایانش مه سه‌لایه توسعه دادند و تحرک کاربران را با استفاده از زمان توقف مدل‌سازی کردند. آنها مسئله را به‌صورت یک مسئله برنامه‌ریزی غیرخطی فرموله کرده و دو الگوریتم پیشنهاد دادند: یک الگوریتم انتخاب برای بهینه‌سازی تصمیمات برون‌سپاری وظایف و یک الگوریتم بهینه‌سازی توزیع‌شده منابع بر اساس الگوریتم ژنتیک برای تخصیص بهینه منابع. نویسندگان در [۳۲] یک الگوریتم توزیع بار به‌طور خاص برای محیط‌های رایانش مه و لبه پیشنهاد کرده‌اند تا تأخیر خدمات در میان گره‌های توزیع‌شده به‌صورت یکنواخت تنظیم شود. نویسندگان یک الگوریتم کاملاً غیرمتمرکز ارائه دادند که مهاجرت وظایف بین گره‌ها را به‌گونه‌ای تنظیم می‌کند که هیچ کاربری، صرف‌نظر از گره‌ای که به آن متصل است، تجربه تأخیر خدمات بدتری نسبت به دیگران نداشته باشد.

جدول ۱ خلاصه‌ای از کارهای پیشین ارائه می‌دهد. در حالی که هدف اصلی بهینه‌سازی در این مقالات معمولاً کاهش تأخیر است، تفاوت‌های قابل توجهی در جنبه‌های دیگر وجود دارد. به عنوان مثال، ارتباط میان مناطق تحت پوشش هر ایستگاه پایه وجود نداشته یا محدود است. در رابطه با مهاجرت وظایف نیز بسیاری از روش‌های پیشین آن را در نظر نمی‌گیرند. در حالی که مدل تحرک استفاده شده در روش ارائه شده در این پژوهش کامل بوده و بر اساس یک شرایط واقعی می‌باشد، برخی از کارهای پیشین در این زمینه محدودیت هایی دارند. تمامی این موارد برتری روش ارائه شده در این کار را نسبت به کارهای پیشین نشان می‌دهند.

## ۳- مدل سامانه

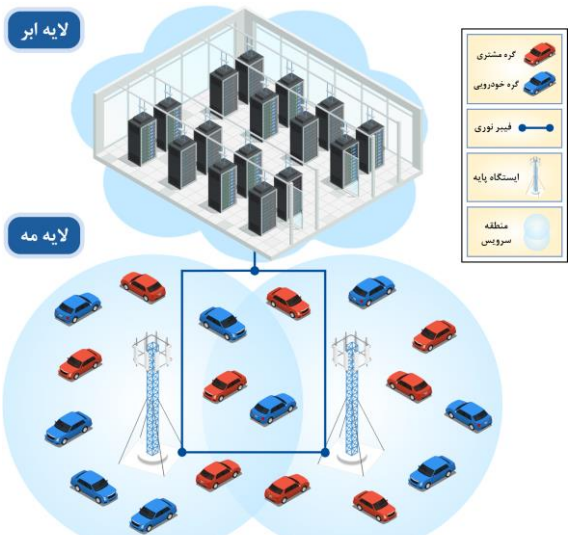
در این بخش، ابتدا معماری سامانه را شرح داده، و سپس بخش‌های مختلف مدل سامانه را تشریح می‌کنیم.

### ۳-۱- معماری سامانه

همانطور که در شکل ۲ نشان داده شده است، مدل سامانه پیشنهادی ما از دو لایه تشکیل شده است: لایه مه و لایه ابر. در لایه مه، گره‌های مشتری به رنگ قرمز نمایش داده شده‌اند. اگر این خودروها ظرفیت لازم برای پردازش وظایف خود را نداشته باشند، با ایستگاه پایه منطقه خدمات خود ارتباط برقرار می‌کنند تا وظایف خود را به یک گره خودرویی مناسب منتقل کنند. ایستگاه‌های پایه همچنین اطلاعات تمامی گره‌های مشتری و خودرویی موجود و تحت پوشش را دریافت می‌کنند تا تصمیمات قابل اجرا برای برون‌سپاری وظایف گرفته شود. اگر گره‌های مشتری به عنوان مدیران و تصمیم‌گیرندگان برای انتقال وظایف در نظر گرفته شوند، لازم است که اطلاعات گره‌های خودرویی و گزینه‌های موجود برای انتقال وظایف را در اختیار داشته باشند. اما این رویکرد منجر به ایجاد سربار بسیار بالا می‌شود. بنابراین، واگذاری فرآیند تصمیم‌گیری برای انتقال وظایف به ایستگاه‌های پایه، گزینه‌ای بسیار کارآمدتر و عملی‌تر است. در لایه مه، گره‌های مشتری و گره‌های خودرویی وجود دارند که وظایف مربوط به گره‌های مشتری را

---

[۱] Min-Max
[۲] Quality of Service
[۳] Docker
[۴] Kubernetes
[۵] Checkpointing
[۶] Fog Collaborator
[۷] Linear Programming
[۸] Binary Particle Swarm Optimization
[۹] Proximal Policy Optimization
[۱۰] Multi-Agent Gated Actor-Critic




شکل ۲: مدل سامانه

پردازش می‌کنند. لایه ابر شامل یک سرور ابری با توان پردازشی بالا است. زمانی که هیچ گره خودرویی مناسبی برای برون‌سپاری وظایف در دسترس نباشد، وظایف گره‌های مشتری از طریق ایستگاه‌های پایه به سرویس‌دهنده ابری منتقل می‌شوند. به طور کلی، ایستگاه‌های پایه دو نقش دارند: فراهم کردن قابلیت‌های مختلف ارتباطی، که در بخش‌های بعدی به طور مفصل بحث خواهد شد، و همچنین انجام تصمیم‌گیری‌های مربوط به برون‌سپاری وظایف. لازم به ذکر است که وسایل نقلیه از ارتباطات بی‌سیم استفاده می‌کنند، در حالی که ارتباط بین ایستگاه‌های پایه و همچنین بین ایستگاه‌های پایه و سرور ابری از طریق کابل‌های فیبر نوری انجام می‌شود.

## ۳-۲- مدل وظیفه

در محیط رایانش مه خودرویی تعداد n گره خودرویی به نام $F = \{vn_1, vn_2, \dots, vn_i, \dots, vn_n\}$ تشکیل شده است که به صورت جغرافیایی توزیع شده‌اند و مختصات آن‌ها در هر گام زمانی به طور پویا تغییر می‌کند. مجموعه‌ای از گره‌های مشتری نیز وجود دارد که شامل m گره به صورت $C = \{cn_1, cn_2, \dots, cn_j, \dots, cn_m\}$ است و دارای مجموعه وظایف $T = \{t_1, t_2, \dots, t_j, \dots, t_m\}$ هستند. هر وظیفه $t_j$ دارای اندازه رایانش $S_j$ است که به طور مستقیم بر تأخیر وظیفه تأثیر می‌گذارد.

## ۳-۳- مدل تحرک

در مدل تحرک مبتنی بر SUMO [۱۳]، سرعت، جهت حرکت و مختصات وسیله نقلیه $V_i$ به ترتیب با $v_i$، $\theta_i(t)$ و $(x_i, y_i)$ نشان داده می‌شوند. در این کار، تمرکز ما بر روی چندین وسیله نقلیه درون محدوده ارتباطی ایستگاه پایه است که در آن تراکم ترافیک و تحرک وسیله نقلیه به طور مشترک بر عملکرد بارگذاری تأثیر می‌گذارند. در سناریوی در نظر گرفته شده وسایل نقلیه بر روی یک جاده دوخطی دوطرفه حرکت می‌کنند. با فرض ثابت بودن سرعت و جهت در هر گام زمانی $\Delta t$، موقعیت هر وسیله نقلیه می‌تواند به صورت زیر به روزرسانی شود:

$$x_i(t+\Delta t) = x_i(t) + v_i(t) \cdot cos\left(\frac{\theta_i(t)\pi}{180}\right) \cdot \Delta t \qquad (۱)$$

$$y_i(t+\Delta t) = y_i(t) + v_i(t) \cdot sin\left(\frac{\theta_i(t)\pi}{180}\right) \cdot \Delta t \qquad (۲)$$

فاصله بین گره مشتری i و گره خودرویی j باید محاسبه شود تا اتصال‌پذیری تعیین گردد. برای هر دو وسیله نقلیه، فاصله در زمان t به صورت زیر است:

$$d_{i,j}(t) = \sqrt{\left(x_i(t) - x_j(t)\right)^2 + \left(y_i(t) - y_j(t)\right)^2} \qquad (۳)$$

که در آن، $(x_i, y_i)$ مختصات گره مشتری i و $(x_j, y_j)$ مختصات گره خودرویی j است.

## ۳-۴- مدل ارتباطی

بر اساس فاصله‌ها و شعاع پوشش، اتصالات به صورت V2V، V2B، B2C یا B2B برقرار می‌شوند که در زیر آمده است:

۱) V2V: گره‌های مشتری (CN) و گره‌های خودرویی (VN) می‌توانند به طور مستقیم با استفاده از پروتکل DSRC ارتباط برقرار کنند.

۲) V2B: گره‌های مشتری (CN) با استفاده از پروتکل 5G با ایستگاه پایه (BS) ارتباط برقرار می‌کنند.

۳) B2C: ایستگاه‌های پایه (BS) می‌توانند با استفاده از ارتباط فیبر نوری با سرویس دهنده ابری ارتباط برقرار کنند.

۴) B2B: ایستگاه‌های پایه (BS) می‌توانند با استفاده از ارتباط فیبر نوری با یکدیگر ارتباط برقرار کنند.

جزئیات مدل تأخیر برای هر استراتژی در زیربخش بعدی مورد بحث قرار خواهد گرفت. ارتباط V2V بین وسایل نقلیه در صورتی وجود دارد که هر دو در شعاع پوشش یکدیگر باشند:

$$\exists\, V2V_{i,j} \Leftrightarrow d_{i,j}(t) \leq (r_i, r_j) \qquad (۴)$$

که در آن $r_i$ و $r_j$ به ترتیب شعاع پوشش خودروی i و j است. ارتباط V2B بین وسیله نقلیه i و ایستگاه پایه f وجود دارد اگر وسیله نقلیه در شعاع پوشش $r_f$ ایستگاه پایه قرار داشته باشد:

$$\exists\, V2B_{i,f} \Leftrightarrow d_{i,f}(t) \leq r_f \qquad (۵)$$

که در آن $d_{i,f}(t)$ فاصله بین وسیله نقلیه i و ایستگاه پایه f است. ارتباط B2B بین ایستگاه‌های پایه f و g می‌تواند وجود داشته باشد اگر مجاور یکدیگر باشند:

$$\exists\, B2B_{f,g} \Leftrightarrow \left(x_f - x_g\right)^2 + \left(y_f - y_g\right)^2 \leq r_f + r_g \qquad (۶)$$

که در آن $r_f$ و $r_g$ به ترتیب شعاع پوشش ایستگاه پایه f و g، $(x_f, y_f)$ مختصات ایستگاه پایه f، و $(x_g, y_g)$ مختصات ایستگاه پایه g است. به عبارت دیگر، ایستگاه‌های پایه i و j مجاور یکدیگر هستند، اگر فاصله بین آن‌ها کمتر از یا برابر با مجموع شعاع پوشش آن‌ها باشد.

## ۳-۵- مدل تاخیر

در این بخش جزئیات مدل تأخیر مورد استفاده را توضیح خواهیم‌داد. فرمول‌ها بر اساس روابط موجود در [۱۴] و [۱۵] پیاده‌سازی شده‌اند.

### ۳-۵-۱- تاخیر رایانش

زمان اجرای وظیفه زمانی که به گره خودرویی $v_i$ برون‌سپاری می‌شود، به اندازه رایانش $S_j$ وظیفه $t_j$ و ظرفیت رایانش $U_i$ بستگی دارد. بنابراین تأخیر رایانش $l_{i,j}^c$ وظیفه به صورت زیر قابل بیان است:

$$l_{i,j}^c = \frac{\lambda S_j}{U_i} \qquad (۷)$$

که در آن $\lambda$ تعداد چرخه‌های مورد نیاز برای پردازش یک بیت داده می‌باشد.

### ۲-۵-۳- تاخیر انتقال

در هر بازه زمانی، مسیر ارتباطی از یک وسیله نقلیه به وسیله نقلیه دیگر در فرایند ارسال درخواست‌های رایانش ثابت می‌ماند. نرخ انتقال از وسیله نقلیه $V_i$ به وسیله نقلیه $V_j$ به صورت زیر نمایش داده می‌شود:

$$\tau_{i,j} = B_{i,j}\left(1 + SINR_{i,j}\right) \qquad (۸)$$

که در آن $B_{i,j}$ نشان‌دهنده پهنای باند اشغال‌شده مسیر ارتباطی است و $SINR_{i,j}$ نسبت سیگنال به تداخل به علاوه نویز مسیر ارتباطی است که به صورت زیر محاسبه می‌شود:

$$SINR_{i,j} = \frac{P_i d_{i,j}^{-\alpha} h_{i,j}^2}{N_0 + \sum_{k \in F \cup C, k \neq i} P_k d_{k,j}^{-\alpha} h_{k,j}^2} \qquad (۹)$$

که در آن $P_i$ توان انتقال فرستنده، $d_{i,j}$ فاصله بین دو وسیله نقلیه $V_i$ و $V_j$، $\alpha$ نمایانگر توان تلفات مسیر[۱] است، $h_{i,j}^2$ بخش تضعیف کوچک مقیاس[۲] است، $N_0$ قدرت نویز گاوسی است و $\sum_{k \in F \cup C, k \neq i} P_k d_{k,j}^{-\alpha} h_{k,j}^2$ نشان‌دهنده تداخل از مسیرهای ارتباطی دیگر است. مشابه مدل ارتباط V2V، فرض بر این است که مسیر ارتباطی بین وسایل نقلیه و ایستگاه پایه در فرایند انتقال ثابت می‌ماند. نرخ انتقال بین وسیله نقلیه و ایستگاه پایه به صورت زیر نمایش داده می‌شود:

$$\tau_{i,B} = B_{i,B}\left(1 + SINR_{i,B}\right) \qquad (۱۰)$$

که در آن $B_{i,B}$ پهنای باند تخصیص‌یافته به کانال V2B بین وسیله نقلیه و ایستگاه پایه است، و $SINR_{i,B}$ نسبت سیگنال به تداخل به علاوه نویز مسیر ارتباطی است که به صورت زیر محاسبه می‌شود:

$$SINR_{i,B} = \frac{P_i d_{i,B}^{-\alpha} h_{i,B}^2}{N_0 + \sum_{k \in F \cup C, k \neq i} P_k d_{k,B}^{-\alpha} h_{k,B}^2} \qquad (۱۱)$$

همچنین، فرض می‌کنیم که مسیر ارتباطی سیمی در فرایند انتقال بین ایستگاه‌های پایه (B2B) ثابت و پایدار می‌ماند. برای اجرای یک وظیفه در وسیله نقلیه، پنج روش وجود دارد: ۱) اجرای محلی بر روی گره مشتری، ۲) برون‌سپاری به یک گره خودرویی که در محدوده ارتباط مستقیم گره مشتری قرار دارد، ۳) برون‌سپاری به یک گره خودرویی که در همان ناحیه خدماتی است با کمک ایستگاه پایه، ۴) برون‌سپاری به گره خودرویی که در منطقه سرویس مجاور قرار دارد، و ۵) برون‌سپاری به ابر. همچنین در زیربخش قبلی ذکر کردیم که چهار استراتژی بارگذاری مختلف داریم: ۱) V2V، ۲) V2B، ۳) B2C و ۴) B2B. برای V2V، تأخیر انتقال بین خودروها به صورت زیر محاسبه می‌شود:

$$l_{t,i,j}^{t_{V2V}} = l_{t,i,j}^{up} + l_{t,i,j}^{down} = \frac{S_i}{\tau_{i,j}} + \frac{\eta S_i}{\tau_{j,i}} \qquad (۱۲)$$

که در آن $l_{t,i,j}^{up}$ و $l_{t,i,j}^{down}$ به ترتیب تأخیر بارگذاری و بارگیری برای برون‌سپاری وظیفه $t_j$ در گام زمانی $\Delta t$ هستند. $\eta$ نسبت اندازه داده‌های خروجی به اندازه داده‌های ورودی است که معمولاً کوچکتر از ۱ است، یعنی $\eta \ll 1$. اگر یک وظیفه به خودرویی که در همان منطقه سرویس قرار دارد اما خارج از محدوده ارتباطی V2V است، برون‌سپاری شود، آنگاه وظیفه ابتدا باید به ایستگاه پایه با استفاده از استراتژی V2B ارسال شود. تأخیر انتقال برای V2B به صورت زیر محاسبه می‌شود:

$$l_{t,i,B}^{t_{V2B}} = l_{t,i,B}^{up} + l_{t,i,B}^{down} = \frac{S_i}{\tau_{i,B}} + \frac{\eta S_i}{\tau_{B,i}} \qquad (۱۳)$$

اگر یک وظیفه به سرویس‌دهنده ابری برون‌سپاری شود، باید از ارتباط B2C استفاده شود. تأخیر استراتژی B2C به صورت زیر محاسبه می‌شود:

$$l_{t,B,C}^{t_{B2C}} = l_{t,B,C}^{up} + l_{t,B,C}^{down} = \frac{S_i}{\tau_{B,C}} + \frac{\eta S_i}{\tau_{C,B}} \qquad (۱۴)$$

اگر یک وظیفه به گره خودرویی که در منطقه سرویس مجاور قرار دارد برون‌سپاری شود، باید از مسیر ارتباطی B2B استفاده شود. تأخیر استراتژی B2B به صورت زیر محاسبه می‌شود:

$$l_{t,B,B}^{t_{B2B}} = l_{t,B,B}^{up} + l_{t,B,B}^{down} = \frac{S_i}{\tau_{B,B}} + \frac{\eta S_i}{\tau_{B,B}} \qquad (۱۵)$$

بنابراین، زمانی که می‌خواهیم وظیفه را به طور محلی بر روی خود گره مشتری پردازش کنیم، تأخیر کل تنها شامل تأخیر رایانش $l_i^c$ است و می‌تواند به صورت زیر محاسبه شود:

$$l_{i_{local}}^{total} = l_i^c \qquad (۱۶)$$

اگر وظیفه به یک گره خودرویی در محدوده ارتباط مستقیم برون‌سپاری شود، تأخیر کل اجرای وظیفه به صورت زیر محاسبه می‌شود:

$$l_{i,j_{direct}}^{total} = l_j^c + l_{t,i,j}^{t_{V2V}} \qquad (۱۷)$$

اگر گره خودرویی که برای برون‌سپاری انتخاب شده است، در محدوده ارتباط مستقیم نبوده، اما در همان منطقه سرویس قرار داشته باشد، تأخیر کل اجرای وظیفه به صورت زیر محاسبه می‌شود:

$$l_{i,j,B_{samezone}}^{total} = l_j^c + l_{t,i,B}^{t_{V2B}} + l_{t,j,B}^{t_{V2B}} \qquad (۱۸)$$

زمانی که گره خودرویی که برای برون‌سپاری انتخاب شده است در منطقه سرویس مجاور قرار داشته باشد، تأخیر کل اجرای وظیفه به صورت زیر محاسبه می‌شود:

$$l_{i,j,B_{adjacent}}^{total} = l_j^c + l_{t,i,B}^{t_{V2B}} + l_{t,j,B}^{t_{V2B}} + l_{t,B,B}^{t_{B2B}} \qquad (۱۹)$$

و در نهایت، اگر سرویس‌دهنده ابری برای برون‌سپاری انتخاب شود، تأخیر کل اجرای وظیفه به صورت زیر محاسبه می‌شود:

$$l_{i,B,C_{cloud}}^{total} = l_j^c + l_{t,i,B}^{t_{V2B}} + l_{t,B,C}^{t_{B2C}} \qquad (۲۰)$$

## ۴- ارائه‌ی روش پیشنهادی

در این بخش، رویکرد خود را برای حل مسئله برون‌سپاری وظایف معرفی می‌کنیم. ساختار کلی روش پیشنهادی در شکل ۳ قابل مشاهده است. هدف اصلی کاهش تأخیر در برون‌سپاری وظایف بین گره های خودرویی است. رویکرد ما از یادگیری تقویتی استفاده می‌کند و با به‌کارگیری عامل‌ها، استراتژی‌های بهینه برون‌سپاری را از طریق تعامل با محیط می‌آموزد. با فرض اینکه $J$ مجموعه وظایف و $O$ مجموعه گزینه‌های پردازش است که شامل N گره خودرویی، سرویس‌دهنده ابری و اجرای محلی بر روی خود مشتری می‌شود. برای هر وظیفه $j \in J$، یک متغیر دودویی $A_{j,o}$ معرفی می‌کنیم که نشان می‌دهد آیا وظیفه j به گزینه $o \in O$ اختصاص داده شده است یا خیر. هر گره مه o دارای ظرفیتی به اندازه $C_o$ است.

---

[۱] Path Loss Exponent

[۲] Small Scale Fading

اگر وظیفه j بر روی گزینه پردازش o اجرا شود، $L_{j,o}$ را به عنوان تأخیر کل در نظر می‌گیریم. مسئله بهینه سازی به صورت زیر است:

$$P1: \quad \min_{A} \left( \sum_{j \in J} \sum_{o \in O} A_{j,o}.L_{j,o} \right)$$

$$s.t.$$

$$C1: \quad \sum_{o \in O} A_{j,o} = 1, \quad \forall j \in J, \qquad (۲۱)$$

$$C2: \quad \sum_{j \in J} A_{j,o} \leq C_o, \quad \forall o \in O,$$

$$C3: \quad A_{j,o} \in \{0,1\}, \quad \forall j \in J, \forall o \in O$$

C1 اطمینان حاصل می‌کند که هر وظیفه دقیقاً به یک گزینه پردازش اختصاص داده شده است، C2 اطمینان حاصل می‌کند که مجموع وظایف اختصاص داده شده به هر گره از ظرفیت آن تجاوز نکند، و C3 به صورت دودویی وضعیت تخصیص وظایف را معین می‌کند. این مسئله با یک نسخه از مسئله چند کوله‌پشتی[۱] هم‌راستا است که NP-سخت[۲] است. بنابراین، طراحی یک الگوریتم مؤثر برای حل مسئله در مقیاس بزرگ ضروری است. برای سازگاری با ماهیت پویا و ترتیبی مسئله برون‌سپاری وظایف، آن را به عنوان یک فرایند تصمیم‌گیری مارکوف[۳] (MDP) مدل‌سازی می‌کنیم. چارچوب MDP مسئله را از نظر حالات، اقدامات، احتمالات انتقال، پاداش‌ها و سیاست‌ها فرمول‌بندی می‌کند.

## ۴-۱- نمایش حالت

یک حالت[۴] $s_t \in S$ در زمان t وضعیت فعلی سیستم را نمایش می‌دهد که شامل این موارد است: موقعیت‌ها و وضعیت‌های تمام گره‌های مشتری، موقعیت‌ها و ظرفیت گره‌های خودرویی، بار فعلی گره‌ها و تخصیص وظایف و تأخیرهای مربوط به هر کدام. برای هر ایستگاه پایه b، بردار وضعیت $S_b$ به صورت زیر تعریف می‌شود:

$$S_b = [P_c, P_f, P_k, N_p, T_s] \qquad (۲۲)$$

که در آن $P_c = (x_c, y_c)$ موقعیت نرمالیزه‌شده گره مشتری مرتبط با ایستگاه پایه b، $P_f = (x_{f1}, y_{f1}, x_{f2}, y_{f2}, \dots, x_{fn}, y_{fn})$ موقعیت نرمالیزه شده گره‌های خودرویی، $P_k = (x_k, y_k)$ موقعیت نرمالیزه‌شده سرویس‌دهنده ابری، $N_p$ تعداد وظایف گره مشتری در ایستگاه پایه b، و $T_s$ اندازه تجمعی نرمال‌شده وظایف برای گره مشتری را نمایش می‌دهد.

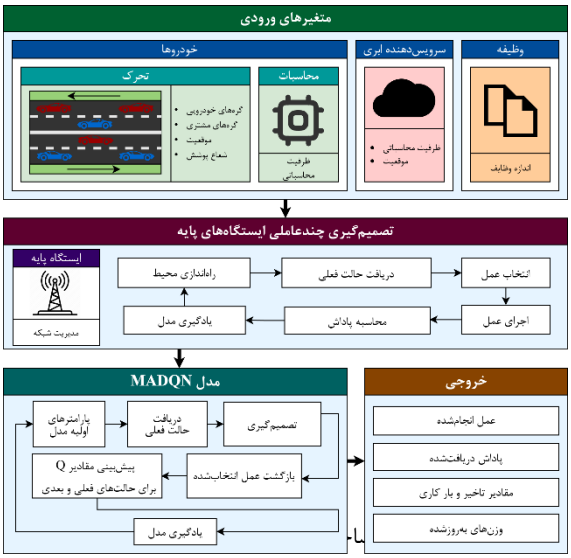
## ۴-۲- فضای عمل

یک عمل $a_t \in \mathcal{A}$ معادل تخصیص یک وظیفه به یک گزینه پردازش خاص است:

$$\mathcal{A} = O \qquad (۲۳)$$

که در آن O شامل N گره خودرویی، سرویس‌دهنده ابری و اجرای محلی است. به طور مشخص، برای هر ایستگاه پایه:

$$\mathcal{A}_b = \{a_1, a_2, \dots, a_N, a_{N+1}, a_{N+2}\} \qquad (۲۴)$$




## ۴-۳- تابع پاداش

پاداش $\mathcal{R}(s_t, a_t)$ بر اساس تأخیر تعریف می‌شود. تأخیر بالا مورد جریمه واقع خواهدشد. به طور خاص:

$$\mathcal{R}(s_t, a_t) = -\sum_{j \in J} A_{j,o} . \frac{L_{j,o} - L_{min}}{L_{max} - L_{min}} \qquad (۲۵)$$

که در آن $L_{j,o}$ تأخیر تخصیص وظیفه j به گزینه پردازش o، و $L_{min}, L_{max}$ حداقل و حداکثر تأخیرهایی هستند که برای نرمال‌سازی استفاده می‌شوند.

## ۴-۴- بهینه‌سازی سیاست

در این مرحله، عامل یاد می‌گیرد که سیاست بهینه‌ای را برای بارگذاری وظایف پیدا کند، به‌گونه‌ای که تأخیر را به حداقل برساند:

$$\pi^* = \arg\max_{\pi} E[\sum_{t=0}^{\infty} \gamma^t \ \mathcal{R}(s_t, a_t) \mid \pi] \qquad (۲۶)$$

که در آن عامل تنزیل[۵] $\gamma \in [۰,۱]$ اهمیت پاداش‌های فوری در مقابل پاداش‌های آینده را متعادل می‌کند.

الگوریتم 1 چارچوب کلی برون‌سپاری وظایف را توصیف می‌کند که از رویکرد شبکه عمیق Q چند عاملی[۶] (MADQN) برای بهینه‌سازی توزیع وظایف در محیط رایانش مه استفاده می‌کند. فرآیند با مرحله مقداردهی اولیه آغاز می‌شود و محیط $F$ با استفاده از پارامترهای تحرک از طریق تابع FogEnvInit(mobility_params) تنظیم می‌شود (خط ۲). سپس، یک عامل جداگانه برای هر ایستگاه پایه $b \in \mathcal{B}$ با استفاده از $\text{MADQNInit}(S, A)$ ایجاد می‌شود (خط ۴)، جایی که هر عامل با اندازه وضعیت S و اندازه اقدام A پیکربندی می‌شود. این پیکربندی به هر عامل این امکان را می‌دهد که به طور مستقل بیاموزد. چارچوب بر اساس یک سری از قسمت‌ها[۷] که با e از ۱ تا E شماره‌گذاری شده‌اند عمل می‌کند و در هر قسمت، از طریق یک دنباله از گام‌های زمانی s از ۱ تا $S_p$ پیش می‌رود (خطوط ۶-۲۰). در هر گام زمانی، زمان شبیه‌سازی جاری $t_{\text{current}}$ تعیین می‌شود و موقعیت‌های گره‌ها در محیط با استفاده از $\text{update\_node\_positions}(\mathcal{F}, t_{\text{current}})$ به‌روزرسانی می‌شود. سپس محیط از طریق $\text{process\_time\_step}(\mathcal{F}, t_{\text{current}})$ پیش می‌رود و هرگونه ورود وظایف جدید از طریق $\text{process\_task\_arrivals}(\mathcal{F})$ پردازش می‌شود. تمامی وظایف معوقه با استفاده از $\mathcal{T}\text{pending} \leftarrow \text{get\_pending\_tasks}(\mathcal{F})$ بازیابی

---

۱ Multiple Knapsacks
۲ NP-Hard
۳ Markov Decision Process
۴ State
۵ Discount Factor
۶ Multi Agent Deep Q Network
۷ Episodes

**Algorithm 1** Task Offloading Framework

**Inputs:**
$F$, {MADQN[b] | $b \in B$}
**Outputs:**
π_b, s
**Initialization:**
$F \leftarrow$ FogEnvInit(mobility_params)
**for all** $b \in B$ **do**
    MADQN[$b$] $\leftarrow$ MADQNInit($S$, $A$)
**end for**
**for** $e \leftarrow 1$ **to** $E$ **do**
    **for** $s \leftarrow 1$ **to** $Sp$ **do**
        *t_current* $\leftarrow f$ *(e, s) // process episode and step*
        update_node_positions($F$, *t_current*)
        process_time_step($F$, *t_current*)
        process_task_arrivals($F$)
        *T_pending* $\leftarrow$ get_pending_tasks($F$)
        **for all** (*bs, client,* $\tau$ ) $\in$ T_pending **do**
            *S_e^((bs))*$\leftarrow$ get_state($F$, *bs*)
            M $\leftarrow$ valid_actions_mask(*S_e^*((*bs*)))
            $a \leftarrow$ MADQN[*bs*].act(*S_e^*((*bs*)), M)
            execute_action($F$, $a$, $\tau$ )
            $r \leftarrow R(\tau, a)$ *// reward from latency*
        **end for**
    **end for**
**end for**

می‌شوند. برای هر وظیفه معوقه $(bs, client, \tau) \in \mathcal{T}$pending بردار حالت جاری $s_e^{(bs)}$ ایستگاه پایه مربوطه bs را با فراخوانی get_state$(\mathcal{F}, bs)$ دریافت می‌کند (خطوط ۱۳ و ۱۴). سپس لیست اقدامات معتبر $M$ از طریق valid_actions_mask$\left(s_e^{(bs)}\right)$ تعیین می‌شود که اقدامات معتبر برای برون‌سپاری و پردازش وظیفه را شناسایی می‌کند (خط ۱۵). عامل مربوط به ایستگاه پایه bs یک اقدام a را با استفاده از دستور act خود بر اساس حالت دریافت‌شده و اقدامات معتبر انتخاب می‌کند (خط ۱۶). اقدام انتخاب‌شده a وظیفه τ را به یک گره انتخابی با اجرای execute_action$(\mathcal{F}, a, \tau)$ اختصاص می‌دهد و پاداش r بر اساس تأخیر ایجاد شده با استفاده از $\mathcal{R}(\tau, a)$ محاسبه می‌شود (خطوط ۱۷ و ۱۸). پس از اتمام تمامی قسمت‌ها و گام‌های زمانی، چارچوب، سیاست‌های آموخته‌شده را برای هر ایستگاه پایه و وضعیت‌های نهایی به‌روزرسانی‌شده محیط را به عنوان خروجی ارائه می‌دهد.

در این پیاده‌سازی، شبکه Q به صورت یک شبکه عصبی با چهار لایه‌ی مخفی طراحی شده است. اندازه ورودی برابر با ابعاد فضای حالت و خروجی برابر با تعداد اقدامات ممکن است. لایه‌های مخفی شامل دو لایه‌ی Dense با ۲۵۶ نورون و دو لایه‌ی دیگر با ۱۲۸ نورون هستند که همگی از تابع فعال‌سازی ReLU استفاده می‌کنند. خروجی شبکه نیز دارای تابع فعال‌سازی خطی است. به‌روزرسانی وزن‌ها با استفاده از الگوریتم Adam و نرخ یادگیری ۰.۰۰۰۵ انجام می‌شود. از حافظه‌ی تجربه تکراری[۱] با ظرفیت ۱۰۰۰۰ تجربه برای نمونه‌گیری تصادفی در یادگیری استفاده شده است. در هر تکرار یادگیری، یک دسته‌ی تصادفی از این حافظه نمونه‌گیری شده و با استفاده از آن، مقادیر هدف Q محاسبه می‌گردد. سپس وزن‌های شبکه Q با استفاده از تابع هزینه‌ی MSE به‌روزرسانی می‌شوند.

همچنین در شرایطی که هیچ عمل معتبری در دسترس نباشد الگوریتم به‌صورت پیش‌فرض یک عمل تصادفی از بین کل فضای عمل انتخاب می‌کند. در پایان فرایند آموزش، مدل آموزش‌دیده ذخیره می‌شود. سیاست بهینه که در قالب وزن‌های آموخته‌شده‌ی شبکه Q هستند، به‌صورت مشخص در فایل‌های خروجی ذخیره می‌گردد. این قابلیت امکان استفاده مجدد از مدل آموزش‌دیده برای اجرا در محیط واقعی یا ارزیابی‌های مستقل را فراهم می‌کند.

وضعیت محیط در هر بازه زمانی به شکلی کاملاً جامع به مدل ارائه می‌شود که شامل موقعیت فعلی گره‌ها و ویژگی مربوط به وظایف می‌باشد. این اطلاعات کامل باعث می‌شود مدل بتواند تصمیم‌گیری بهینه را صرفاً بر اساس وضعیت فعلی انجام دهد و نیاز کمتری به حافظه ضمنی مانند لایه‌های بازگشتی یا مکانیزم‌های توجه برای یادگیری از تاریخچه طولانی‌تر داشته باشد. علاوه بر این، به‌دلیل پله‌های زمانی نسبتاً کوتاه در سناریوی برون‌سپاری وظایف در این پژوهش، مثلاً در حدود چند ثانیه برای هر تصمیم، افزودن لایه‌های بازگشتی یا توجه منجر به افزایش پیچیدگی محاسباتی چشمگیری می‌شود، بدون آنکه بهبود قابل‌توجهی در عملکرد مدل داشته باشد. نتایج تجربی نیز نشان‌دهنده موفقیت روش پیشنهادی در کنترل مؤثر تأخیر و مدیریت پویایی‌های محیط بوده است.

## ۵- نتایج شبیه‌سازی

شبیه‌سازی بر اساس زبان پایتون و مدل تحرک بر اساس شبیه‌ساز SUMO پیاده شده‌اند. محیط شبیه‌سازی SUMO بر اساس شهر برلین است، منطقه‌ای با عرض ۱۵۶۲ متر و ارتفاع ۱۰۵۱ متر. خروجی SUMO شامل مختصات وسایل نقلیه در هر گام زمانی است. وسایل نقلیه برای ۷۲۰ قسمت تحت نظر قرارگرفتند، که هر قسمت شامل ۵ گام زمانی بود. هر گام زمانی برابر با ۰.۵ ثانیه است؛ بنابراین، زمان کل شبیه‌سازی SUMO برابر با ۳۰ دقیقه (۱۸۰۰ ثانیه) است. بالغ بر ۱۷۰۰ وسیله نقلیه در شرایط مختلف تحت نظر قرارگرفتند.

الگوریتم پیشنهادی با چهار روش پایه مقایسه می‌شود:

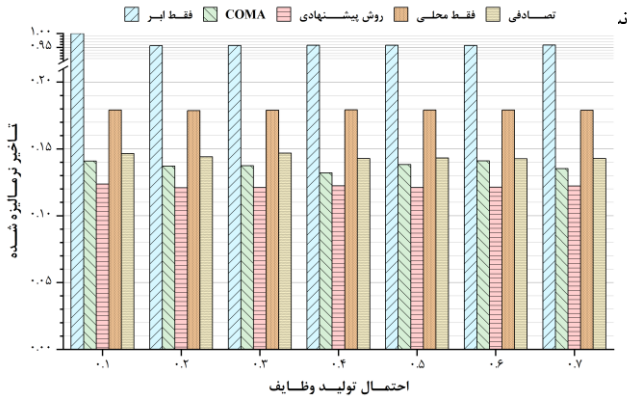
- **COMA**[۲]: که بر اساس روش یادگیری تقویتی بازیگر-انتقاد است که در کار [۱۴] توصیف شده است.
- **الگوریتم تصادفی:** که در آن تمام تصمیمات پردازش به صورت تصادفی گرفته می‌شود.
- **فقط پردازش محلی:** که در آن تمام وظایف به طور محلی بر روی خود گره مشتری پردازش می‌شوند.
- **فقط برون‌سپاری به ابر:** که در آن تمام وظایف برای پردازش به سرویس‌دهنده ابری ارسال می‌شوند.

تمام مقادیر شبیه‌سازی نرمالیزه شده اند تا نتایج مربوطه خواناتر باشند و تفاوت بین روش‌های مختلف به وضوح در شکل‌ها قابل مشاهده باشند. در شکل ۴ (الف) عملکرد روش‌های مختلف با تعداد متغیر گره‌های مشتری (از ۲۰ تا ۸۰) از نظر تأخیر قابل مشاهده است. در تمام حالات، تعداد گره‌های خودرویی ثابت و برابر با ۲۵ بود. مشاهده می‌شود که روش پیشنهادی ما توانسته است کمترین تأخیر متوسط را بدست آورد، با کاهش ۱۲٪ نسبت به COMA که در رتبه بعدی قرار دارد، در حالی که روش برون‌سپاری به ابر بیشترین تأخیر را دارد که نشان‌دهنده بار زیاد ناشی از پردازش متمرکز است. روش‌های Local Only و Random تأخیرهای متوسطی را نشان می‌دهند.

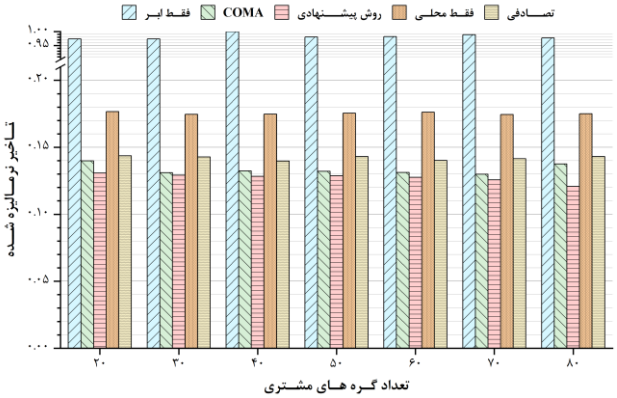
شکل ۴ (ب) عملکرد روش‌های مختلف را در زمانی که تعداد وظایف افزایش می‌یابد، نشان می‌دهد. یک متغیر به نام احتمال تولید وظیفه[۳] تعریف شده که احتمال ایجاد یک وظیفه جدید برای پردازش توسط گره مشتری را در هر گام زمانی تعیین می‌کند. هرچه این عدد بیشتر باشد، احتمال بیشتری برای داشتن وظایف جدید برای گره مشتری وجود دارد. در این شکل متوسط تأخیر پنج روش برای

[۱] Experience Replay
[۲] Counterfactual Multi-Agent
[۳] Task Generation Probability

احتمال‌های مختلف تولید وظیفه (از ۰.۱ تا ۰.۷) و تعداد ثابت گره مشتری و گره‌های خود

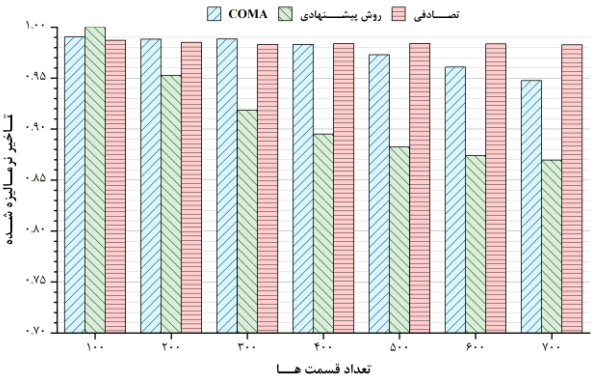


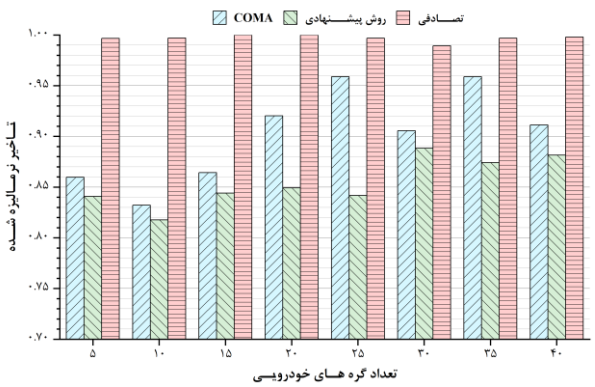

مشاهده می‌شود که عملکرد روش پیشنهادی با افزایش تعداد گره‌های خودرویی نه




ب الف

شکل ۴: الف) تغییرات تاخیر وظایف با تغییر تعداد گره های مشتری؛ ب) تغییرات تاخیر وظایف با تغییر احتمال تولید وظایف




ب الف

شکل ۵: الف) تاخیر نرمالیزه شده با تغییر تعداد قسمت‌ها؛ ب) تاخیر نرمالیزه شده با تغییر تعداد گره‌های خودرویی

می‌شود که روش پیشنهادی کمترین سطوح تأخیر را در حالت‌های مختلف ارائه می‌دهد. COMA نیز دوم است که تأخیرهای بالاتری نسبت به

روش پیشنهادی دارد. روش فقط ابر بیشترین تأخیر را دارد که نشان‌دهنده ناکارآمدی در پردازش وظایف است. روش‌های فقط محلی و تصادفی تأخیرهای متوسطی دارند و به طور کلی روش تصادفی نسبت به روش فقط محلی عملکرد بهتری دارد.

نمودار میله‌ای در شکل ۵ (الف) تأخیر متوسط روش‌های مختلف را برای تعداد مختلف قسمت‌ها (از ۱۰۰ تا ۷۰۰) و تعداد ثابت گره مشتری و گره‌های خودرویی به ترتیب ۸۰ و ۲۵ مقایسه می‌کند. روش پیشنهادی تأخیر کمتری نسبت به سایر روش‌ها دارد و این شکاف با افزایش تعداد قسمت‌ها و مدت زمان شبیه‌سازی بیشتر می‌شود، به طوری که در ازای ۷۰۰ قسمت، تأخیر آن تا ۱۱٪ کمتر از COMA است.

شکل ۵ (ب) تأخیر متوسط سه روش مختلف را برای تعداد متغیر گره‌های خودرویی (از ۵ تا ۴۰) مقایسه می‌کند، در حالتی که تعداد گره‌های مشتری ثابت و برابر با ۸۰ است. تعداد گره‌های خودرویی تأثیر بیشتری بر تأخیر متوسط در تعداد پایین‌تر دارند (تعداد ۵ و ۱۰ گره)، که مورد انتظار است، زیرا گره‌های خودرویی سطوح تأخیر کمتری نسبت به سایر گزینه‌ها ارائه می‌دهند. روش پیشنهادی کمترین تأخیر را ارائه می‌دهد COMA با تأخیرهای کمی بالاتر در رتبه بعدی قرار دارد و روش تصادفی در تمامی تعداد گره‌های خودرویی بیشترین تأخیر را دارد. همچنین

به طور کلی، روش ارائه شده با توزیع بهتر وظایف و استفاده از بهینه از گزینه‌های ارتباطی مختلف، می‌تواند عملکرد بهتری نسبت به روش‌های دیگر داشته باشد. جدول ۲ توزیع پردازش‌های انجام شده برای گزینه‌های مختلف را نشان می‌دهد:

۱) مستقیم: وظیفه مستقیماً از گره کاربر به گره خودرویی ارسال می‌شود، بدون اینکه ایستگاه پایه واسطه باشد.
۲) منطقه مشابه: وظیفه با واسطه‌گری ایستگاه پایه به یک گره خودرویی در همان منطقه ارسال می‌شود.
۳) منطقه مجاور: وظیفه از طریق ارتباط بین ایستگاه‌های پایه به یک گره خودرویی مناسب در منطقه مجاور برون‌سپاری می‌شود.
۴) محلی: وظیفه به صورت محلی روی خود گره کاربر پردازش می‌شود.
۵) ابر: وظیفه به فضای ابری ارسال می‌شود.

توزیع وظایف برای مقادیر مختلف احتمال تولید وظایف (از ۰.۱ تا ۰.۷) بررسی شده است، در حالی که تعداد گره‌های کاربر و خودرویی به‌ترتیب ثابت و برابر با ۸۰ و ۲۵ در نظر گرفته شده‌اند. با افزایش احتمال تولید وظایف، تعداد وظایف در همه گزینه‌های پردازشی افزایش می‌یابد، که نشان‌دهنده مقیاس‌پذیری سیستم و توانایی آن برای سازگاری با بار کاری بیشتر است. در میان این گزینه‌ها، برون‌سپاری به منطقه مجاور پرکاربردترین روش است؛ به طوری که تعداد برون‌سپاری‌ها از ۳۴,۴۵۰ در احتمال ۰.۱ شروع شده و تا ۲۴۹,۹۴۶ در احتمال ۰.۷ افزایش می‌یابد. این

موضوع نشان می‌دهد سیستم دارد.

جدول ۲: توزیع پردازش های انجام شده برای مقادیر مختلف احتمال تولید وظایف

| احتمال تولید وظیفه | ۰.۱ | ۰.۲ | ۰.۳ | ۰.۴ | ۰.۵ | ۰.۶ | ۰.۷ |
|---|---|---|---|---|---|---|---|
| مستقیم | ۴۸۵۲ | ۶۹۸۰ | ۱۲۴۳۹ | ۱۵۰۵۱ | ۲۱۷۳۱ | ۲۳۹۳۷ | ۲۶۵۹۱ |
| منطقه مشابه | ۱۵۹۴۰ | ۲۵۲۸۲ | ۴۵۸۷۱ | ۵۹۶۲۲ | ۷۸۸۷۱ | ۱۰۱۲۳۴ | ۱۱۱۷۱۱ |
| منطقه مجاور | ۳۴۴۵۰ | ۷۸۶۴۸ | ۱۰۷۳۵۶ | ۱۴۶۸۴۳ | ۱۷۶۱۵۵ | ۲۰۷۴۶۵ | ۲۴۹۹۴۶ |
| محلی | ۷۵ | ۹۱ | ۱۵۳ | ۲۱۹ | ۲۹۵ | ۲۷۸ | ۳۶۸ |
| ابر | ۴۹ | ۷۲ | ۹۶ | ۱۱۵ | ۱۶۱ | ۱۸۰ | ۱۹۵ |

گزینه منطقه مشابه دومین روش پرکاربرد است، با تعداد برون‌سپاری بین ۱۵,۹۴۰ تا ۱۱۱,۷۱۱ که اهمیت آن را به‌عنوان جایگزینی برای گزینه برون‌سپاری به منطقه مجاور نشان می‌دهد.

روش برون‌سپاری مستقیم استفاده متوسطی دارد و تعداد برون‌سپاری از ۴,۸۵۲ تا ۲۶,۵۹۱ افزایش می‌یابد.

در مقابل، استفاده از پردازش محلی و فضای ابری کمتر است. پردازش محلی تنها با ۷۵ پردازش در احتمال ۰.۱ آغاز می‌شود و به ۳۶۸ در احتمال ۰.۷ می‌رسد، که نشان‌دهنده اتکای کم سیستم به اجرای محلی است. برون‌سپاری به فضای ابری کمترین تعداد را دارد، بین ۴۹ تا ۱۹۵، که بیانگر ترجیح سیستم به راهکارهای برون‌سپاری غیرمتمرکز و نزدیک برای کاهش تأخیر و سربار شبکه است.

## ۶- جمع‌بندی

در این پژوهش، یک روش برون‌سپاری وظایف مبتنی بر یادگیری تقویتی به منظور حل مشکل تأخیر در محیط‌های رایانش مه خودرویی ارائه گردید. برای چالش مهاجرت وظایف با استفاده از یک مدل ارتباطی جامع رسیدگی شده است. تحرک خودروها نیز با استفاده از شبیه‌ساز SUMO با استفاده از یک سناریو واقعی و دقیق شبیه‌سازی شده است. نتایج اولیه نشان می‌دهند که روش پیشنهادی ما در زمینه تاخیر توانسته است تا ۱۲ درصد بهتر از روش های پیشین عمل نماید. مطالعات آینده می‌توانند پارامترهای دیگری همانند انرژی و تعادل بار کاری را نیز به عنوان اهداف مهم در نظر بگیرند.

## ۷- مراجع

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities," in *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, Feb. 2014, doi: 10.1109/JIOT.2014.2306328.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015, doi: 10.1109/COMST.2015.2444095.

[3] Y. -K. Chen, "Challenges and opportunities of internet of things," *17th Asia and South Pacific Design Automation Conference*, Sydney, NSW, Australia, 2012, pp. 383-388, doi: 10.1109/ASPDAC.2012.6164978.

[4] M. Elkhodr, S. Shahrestani and H. Cheung, "The Internet of Things: Vision & challenges," *IEEE 2013 Tencon - Spring*, Sydney, NSW, Australia, 2013, pp. 218-222, doi: 10.1109/TENCONSpring.2013.6584443.

[5] L. Chettri and R. Bera, "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems," in *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16-32, Jan. 2020, doi: 10.1109/JIOT.2019.2948888.

[6] N. Kumari, A. Yadav, P. K. Jana, "Task Offloading in Fog Computing: A Survey of Algorithms and Optimization Techniques," *Computer Networks*, vol. 214, pp. 109137, 2022, ISSN 1389-1286.

[7] M. Mukherjee, L. Shu and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826-1857, thirdquarter 2018, doi: 10.1109/COMST.2018.2814571.

[8] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416-464, Firstquarter 2018, doi: 10.1109/COMST.2017.2771153.

[9] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin and S. Chen, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," in *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860-3873, June 2016, doi: 10.1109/TVT.2016.2532863.

[10] Aisha Muhammad A. Hamdi, Farookh Khadeer Hussain, Omar K. Hussain, Task offloading in vehicular fog computing: State-of-the-art and open issues, *Future Generation Computer Systems*, Volume 133, 2022, Pages 201 212, ISSN 0167-739X, https://doi.org/10.1016/j.future.2022.03.019.

[11] N. C. Luong et al., "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133-3174, Fourthquarter 2019, doi: 10.1109/COMST.2019.2916583.

[12] W. Chen, X. Qiu, T. Cai, H. -N. Dai, Z. Zheng and Y. Zhang, "Deep Reinforcement Learning for Internet of Things: A Comprehensive Survey," in *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1659-1692, thirdquarter 2021, doi: 10.1109/COMST.2021.3073036.

[13] Krajzewicz, D, "Traffic Simulation with SUMO – Simulation of Urban Mobility," In: Barceló, J. (eds) Fundamentals of Traffic Simulation, *International Series in Operations Research & Management Science*, vol 145, Springer, New York, NY, 2010

[14] Y. Hou, Z. Wei, R. Zhang, X. Cheng and L. Yang, "Hierarchical Task Offloading for Vehicular Fog Computing Based on Multi-Agent Deep Reinforcement Learning," in *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 3074-3085, April 2024, doi: 10.1109/TWC.2023.3305321.

[15] R. Yadav, W. Zhang, O. Kaiwartya, H. Song and S. Yu, "Energy-Latency Tradeoff for Dynamic Computation Offloading in Vehicular Fog Computing," in *IEEE Transactions on Vehicular Technology*, vol. 69,

no. 12, pp. 14198-14211, Dec. 2020, doi: 10.1109/TVT.2020.3040596.

[16] J. Zhao, M. Kong, Q. Li and X. Sun, "Contract-Based Computing Resource Management via Deep Reinforcement Learning in Vehicular Fog Computing," in *IEEE Access*, vol. 8, pp. 3319-3329, 2020, doi: 10.1109/ACCESS.2019.2963051.

[17] I. Sarkar and S. Kumar, "Delay-aware Intelligent Task Offloading Strategy in Vehicular Fog Computing," 2022 *International Conference on Connected Systems & Intelligence (CSI)*, Trivandrum, India, 2022, pp. 1-6, doi: 10.1109/CSI54720.2022.9924066.

[18] Z. Gao, L. Yang and Y. Dai, "Fast Adaptive Task Offloading and Resource Allocation via Multiagent Reinforcement Learning in Heterogeneous Vehicular Fog Computing," in *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 6818-6835, 15 April15, 2023, doi: 10.1109/JIOT.2022.3228246.

[19] D. Lan, A. Taherkordi, F. Eliassen and L. Liu, "Deep Reinforcement Learning for Computation Offloading and Caching in Fog-Based Vehicular Networks, "*2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Delhi, India, 2020, pp. 622-630, doi: 10.1109/MASS50613.2020.00081.

[20] I. Sarkar and A. Taherkordi, "Enhancing Task Efficiency in Vehicular Fog Computing: Leveraging Mobility Prediction and Min-Max Optimization for Reduced Latency, "*2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*", Hong Kong, Hong Kong, 2023, pp. 1-7, doi: 10.1109/VTC2023-Fall60731.2023.10333621.

[21] G. Zhao and H. Cui, "DRL-Based Dependent Task Offloading in Collaborative Vehicular Networks,"*2024 6th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI)*, Guangzhou, China, 2024, pp. 227-231, doi: 10.1109/IoTAAI62601.2024.10692789.

[22] Q. Wu, S. Wang, H. Ge, P. Fan, Q. Fan and K. B. Letaief, "Delay-Sensitive Task Offloading in Vehicular Fog Computing-Assisted Platoons," in *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 2012-2026, April 2024, doi: 10.1109/TNSM.2023.3322881.

[23] A. Chebaane, M. K. Arshad, F. Burger and A. Khelil, "A Layered Strategy for Reducing Offloading Latency in Fog Computing, "*2024 9th International Conference on Fog and Mobile Edge Computing (FMEC)*, Malmö, Sweden, 2024, pp. 176-182, doi: 10.1109/FMEC62297.2024.10710234.

[24] Y. Yang, W. Cheng and J. Wang, "Computation Offloading for Latency Reduction in Regionalized Hierarchical Vehicular Fog Network, "*GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, Rio de Janeiro, Brazil, 2022, pp. 5807-5812, doi: 10.1109/GLOBECOM48099.2022.10001703.

[25] J. Shi, J. Du, J. Wang, J. Wang and J. Yuan, "Priority-Aware Task Offloading in Vehicular Fog Computing Based on Deep Reinforcement Learning," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16067-16081, Dec. 2020, doi: 10.1109/TVT.2020.3041929.

[26] L. Sun, M. Liu, J. Guo, X. Yu and S. Wang, "Deep Reinforcement Learning Empowered Resource Allocation in Vehicular Fog Computing," in *IEEE Transactions on Vehicular Technology*, vol. 73, no. 5, pp. 7066-7076, May 2024, doi: 10.1109/TVT.2023.3338578.

[27] C. Zhu et al., "Folo: Latency and Quality Optimized Task Allocation in Vehicular Fog Computing," in *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4150-4161, June 2019, doi: 10.1109/JIOT.2018.2875520.

[28] S. -S. Lee and S. Lee, "Resource Allocation for Vehicular Fog Computing Using Reinforcement Learning Combined With Heuristic Information," in *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10450-10464, Oct. 2020, doi: 10.1109/JIOT.2020.2996213.

[29] Z. Wei, B. Li, R. Zhang, X. Cheng and L. Yang, "Many-to-Many Task Offloading in Vehicular Fog Computing: A Multi-Agent Deep Reinforcement Learning Approach," in *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2107-2122, March 2024, doi: 10.1109/TMC.2023.3250495.

[30] R. Siyadatzadeh et al., "ReLIEF: A Reinforcement-Learning-Based Real-Time Task Assignment Strategy in Emerging Fault-Tolerant Fog Computing," in *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10752-10763, 15 June15, 2023, doi: 10.1109/JIOT.2023.3240007.

[31] D. Wang, Z. Liu, X. Wang and Y. Lan, "Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks," in *IEEE Access*, vol. 7, pp. 43356-43368, 2019, doi: 10.1109/ACCESS.2019.2908263.

[31] G. P. Mattia, A. Pietrabissa and R. Beraldi, "A Load Balancing Algorithm for Equalising Latency Across Fog or Edge Computing Nodes," in *IEEE Transactions on Services Computing*, vol. 16, no. 5, pp. 3129-3140, Sept.-Oct. 2023, doi: 10.1109/TSC.2023.3265883.