مقاله پژوهشی

ارائه روشی برای مدیریت منابع در شبکه‌های Fog-DSDN با بهره‌گیری از معماری میکروسرویس و شبکه‌های ESN

عباس پیامنی، احسان متین‌فر، الهام مقامی و محمد جانبزرگی

چکیده: در شبکه‌های معماری‌های رایانش مه نرم‌افزار محور (Fog–DSDN) تخصیص بهینه‌ی منابع میان‌گره‌های متعدد با محدودیت‌های زمانی، پردازشی و حافظه‌ای، یکی از چالش‌های اساسی محسوب می‌شود. روش‌های موجود غالباً فاقد سازوکار تطبیقی در مواجهه با تغییرات سریع با شبکه هستند و در نتیجه، موجب افزایش تأخیر سرویس و افت کارایی منابع می‌گردند. در این پژوهش، روشی نوآورانه مبتنی بر معماری میکروسرویس و شبکه حالت پژواک (Echo State Network - ESN) برای مدیریت و بهینه‌سازی تخصیص منابع در محیط Fog–DSDN ارائه شده است. در این روش، هر میکروسرویس به‌صورت مستقل وظیفه جمع‌آوری و پردازش داده‌های محلی را بر عهده دارد و خروجی آن در قالب نقشه اطلاعاتی (Information Map) در سطح کنترلرهای مه تجمیع می‌شود. سپس، مدل ESN با یادگیری الگوهای زمانی ترافیک، بار پردازشی آینده‌ی گره‌ها را پیش‌بینی کرده و تصمیم تخصیص منابع را به‌صورت تطبیقی اتخاذ می‌کند. طراحی دولایه‌ی صف‌های ورودی و خروجی در هر سلول اطلاعاتی نیز موجب کاهش تراکم پردازشی و بهبود زمان پاسخ سیستم گردیده است. برای ارزیابی عملکرد روش پیشنهادی، شبیه‌سازی‌هایی در بستر OMNeT++ در سناریوهای مختلف ترافیکی انجام شد و نتایج با دو بستر پایه شامل معماری میکروسرویس بدون پیش‌بینی بار و مدل پایه‌ی غیر سلسله‌مراتبی Fog-DSDN و روش مدیریت منابع TFS مقایسه گردید. نتایج تجربی نشان می‌دهد روش پیشنهادی به طور میانگین نسبت به دو بستر معماری میکروسرویس و معماری پایه Fog-DSDN و روش مدیریت منابع TFS موجب بهبود ۱۲٫۵۷٪ در بهره‌وری منابع پردازشی، ۱۸٫۸۵٪ در استفاده بهینه از حافظه و ۳۹٫۱۳٪ در کاهش تأخیر سرویس شده است. این دستاوردها بیانگر آن است که ترکیب ساختار ماژولار میکروسرویس‌ها با پیش‌بینی هوشمند بار توسط ESN می‌تواند راهکاری کارا، مقیاس‌پذیر و سبک‌وزن برای مدیریت منابع در شبکه‌های Fog–DSDN فراهم آورد.

***کلیدواژه:* پردازش مه، شبکه‌های توزیع‌شده نرم‌افزار محور، مدیریت منابع، یادگیری عمیق.**

۱- مقدمه

با گسترش روزافزون دستگاه‌های اینترنت اشیا (IoT) و افزایش حجم داده‌های تولیدی در لبه شبکه[1]، نیاز به پردازش سریع، تصمیم‌گیری بلادرنگ و تخصیص هوشمند منابع بیش‌ازپیش احساس می‌شود. معماری‌های سنتی نظیر رایانش ابری[2] به دلیل ماهیت متمرکز خود، پاسخ‌گوی نیازهای مرتبط با تأخیر پایین، مصرف بهینه منابع و واکنش سریع به تغییرات ترافیک نیستند. در این راستا، معماری‌های رایانش مه[3] نرم‌افزار محور[4] (Fog-DSDN) به‌عنوان رویکردی نوین برای انتقال قابلیت پردازش و کنترل به نزدیکی کاربران نهایی مطرح شده‌اند. بااین‌حال، مدیریت مؤثر منابع در این شبکه‌ها به دلیل ناهمگونی گره‌ها، پویایی ترافیک، محدودیت منابع پردازشی و حافظه‌ای، و وابستگی زیاد به کنترلر مرکزی SDN[5] همچنان یکی از چالش‌های کلیدی محسوب می‌شود. ناتوانی در تخصیص تطبیقی منابع می‌تواند منجر به ازدحام، افت کیفیت خدمات[6] (QoS) و کاهش بهره‌وری کل شبکه گردد [۱]. از این رو، توسعه‌ی راهکارهای سبک‌وزن[7] و هوشمند برای تخصیص منابع در محیط‌های Fog-DSDN پویا و ناهمگن ضرورت ویژه‌ای دارد.

مقالاتی که توسط لی و زو ارائه شده‌اند، بر به‌کارگیری یادگیری تقویتی عمیق و شبکه‌های عصبی برای مکان‌یابی بهینه کنترلر SDN، انتخاب دینامیک لبه و بهینه‌سازی مصرف انرژی تمرکز دارند. در این روش‌ها، با استفاده از مدل‌هایی نظیر LSTM[8]، GRU[9] و PPO[10] تلاش شده است تا تعادل بار، کاهش تأخیر و بهره‌وری بهینه از منابع محقق شود. با این حال، پیچیدگی بالا، نیاز به منابع پردازشی زیاد و عدم توجه کامل به مقیاس‌پذیری در سناریوهای بزرگ، از جمله محدودیت‌های این رویکردها است [۲] و [۳].

از سوی دیگر، رویکرد پیشنهادی توسط محمودی تحت عنوان CAAF-DSDN[11]، با بهره‌گیری از خوشه‌بندی تطبیقی و مدل‌سازی جبری، کاهش سربار محاسباتی، مهاجرت بهینه ماشین‌های مجازی و بهبود توازن بار را ممکن ساخته، اما به انتخاب حساس پارامترهای اولیه وابسته است که در محیط‌های نویزی یا پراکنده ممکن است منجر به افت عملکرد شود [۴]. در کنار این رویکردها، چارچوب‌های معماری چندلایه

این مقاله در تاریخ ۲۸ فروردین ماه ۱۴۰۴ دریافت و در تاریخ ۱۲ آبان ماه ۱۴۰۴ بازنگری شد.

عباس پیامنی (نویسنده مسئول)، دانشکده برق و کامپیوتر، شعبه دورود، دانشگاه ملی مهارت، دورود، ایران، (email: abbas.payamani@yahoo.com).

احسان متین‌فر، گروه کامپیوتر، دانشکده فنی و مهندسی، واحد دورود، دانشگاه آزاد اسلامی، دورود، ایران، (email: ematinfar@gmail.com).

الهام مقامی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران، (email: e.maghami@el.iut.ac.ir).

محمد جانبزرگی، دانشکده برق و کامپیوتر، شعبه دورود، دانشگاه ملی مهارت، دورود، ایران، (email: mohammad.janbozorgi@yahoo.com).

1. Edge
2. Cloud Computing
3. Fog Computing
4. Fog-Software Defined Distributed Network
5. Software Defined Network
6. Quality of Services
7. Light Weight
8. Long Short-Term Memory
9. Gated Recurrent Unit
10. Proximal Policy Optimization
11. Cluster-Aware Adaptive Factorization method for DSDN

مانند آنچه بریاتور معرفی کرده است، با توزیع بار میان لایه‌های لبه، مه و ابر، تلاش می‌کنند هم‌زمان به چالش‌های تأخیر، مصرف پهنای باند و سازگاری با محیط‌های صنعتی پاسخ دهند، اما اجرای پیچیده و مدیریت داده‌های توزیع‌شده از چالش‌های اصلی آن به شمار می‌رود [۵]. همچنین، ژین با بهره‌گیری از یادگیری تقویتی چندعامله[1] (MARL) چارچوبی برای زمان‌بندی وظایف در زیرساخت‌های ابری- مه ارائه کرده که با کاهش مصرف انرژی و زمان پاسخ، گامی نوین در افزایش انعطاف‌پذیری و بهره‌وری منابع برداشته است، ولی همچنان با چالش‌هایی چون پیچیدگی محاسباتی بالا و نیاز به همگرایی پایدار مواجه است [۶]. از منظر مرور سیستماتیک، رحمانی و همکاران نشان داده‌اند که الگوریتم‌های فراابتکاری، با وجود مزایای خود در شرایط پویا، در عمل نیازمند بهینه‌سازی پارامترها و کاهش پیچیدگی اجرایی هستند تا در محیط‌های واقعی رایانش توزیع‌شده کاربردی‌تر باشند [۷]. متین‌فر و همکاران با بهره‌گیری از یک روش کارامد مبتنی بر ریاضیات نوین نشان دادند که بدون توجه به محدودیت‌های شبکه‌های مبتنی بر لبه می‌توان راهکاری برای بهبود بهره‌وری منابع ارائه داد [۸]. تحلیل انتقادی این مطالعات نشان می‌دهد که هنوز شکاف علمی مهمی در توسعه‌ی یک مدل جامع، مقیاس‌پذیر و سبک برای تخصیص تطبیقی منابع در شبکه‌های Fog-DSDN وجود دارد. اغلب پژوهش‌های موجود فاقد سازوکار پیش‌بینی بار لحظه‌ای گره‌ها هستند و تصمیم‌گیری تخصیص منابع را تنها بر اساس وضعیت جاری انجام می‌دهند. این موضوع باعث تأخیر در واکنش به تغییرات ترافیکی می‌شود. از سوی دیگر، نبود معماری ماژولار و مستقل از کنترل‌کننده مرکزی منجر به افزایش سربار کنترلی و کاهش انعطاف‌پذیری سامانه می‌شود. در نتیجه، نیاز به روشی احساس می‌شود که بتواند در قالب ساختاری توزیع‌شده، همزمان پیش‌بینی مصرف منابع، تصمیم‌گیری تطبیقی و هماهنگی میان گره‌ها را با حداقل پیچیدگی محاسباتی فراهم آورد.

در این پژوهش، روشی نوآورانه برای تخصیص تطبیقی منابع در شبکه‌های Fog–DSDN ارائه شده است که با توزیع فرایند تصمیم‌گیری و پیش‌بینی بار میان‌گره‌های مه، به دنبال افزایش بهره‌وری در سه بعد اصلی منابع–حافظه، پردازنده و زمان پاسخ است. در روش پیشنهادی، از سازوکار پخش وظایف مبتنی بر میکروسرویس[2] استفاده شده است تا هر مؤلفه از سیستم، به‌صورت مستقل وظیفه‌ی جمع‌آوری داده، تحلیل وضعیت و تخصیص منابع را بر عهده گیرد. این معماری ماژولار، افزون بر کاهش سربار پردازشی کنترلر مرکزی، مقیاس‌پذیری و قابلیت به‌روزرسانی سیستم را به طور چشمگیری افزایش می‌دهد.

برای پیش‌بینی رفتار دینامیکی شبکه و مصرف آینده‌ی منابع، از شبکه حالت پژواک[3] (ESN) به‌عنوان مدل پیش‌بینی سبک‌وزن و کارا استفاده شده است. مدل ESN با تحلیل الگوهای زمانی ترافیک و یادگیری تاریخچهٔ تغییرات گره‌ها، بدون نیاز به فرایند آموزش سنگین، قادر است بار پردازشی آینده‌ی هر گره را پیش‌بینی کرده و تصمیمات تخصیص منابع را به‌صورت تطبیقی و بلادرنگ اتخاذ کند. داده‌های محلی هر گره ابتدا توسط میکروسرویس‌های سبک جمع‌آوری و پردازش شده و در قالب نقشه اطلاعاتی[4] در سطح کنترلر مه تجمیع می‌شوند. سپس، بر اساس خروجی مدل ESN، منابع پردازشی، حافظه‌ای و زمانی میان‌گره‌ها بازتخصیص می‌یابد تا از بروز تراکم، اتلاف منابع و تأخیر اضافی جلوگیری شود.

نوآوری اصلی این پژوهش در ترکیب هم‌افزای دو مفهوم کلیدی، یعنی معماری میکروسرویس و مدل پیش‌بینی ESN نهفته است. این ترکیب امکان تصمیم‌گیری توزیع‌شده و یادگیری تطبیقی رفتار شبکه را فراهم می‌کند، درحالی‌که پیچیدگی محاسباتی سیستم را نسبت به مدل‌های یادگیری عمیق متداول به میزان قابل‌توجهی کاهش می‌دهد. بدین ترتیب، روش پیشنهادی قادر است ضمن حفظ سبک‌وزنی و سرعت تصمیم‌گیری، پایداری عملکرد، کاهش تأخیر سرویس و بهره‌وری انرژی در محیط‌های پویا و ناهمگن Fog–DSDN را تضمین کند.

ضرورت انجام این پژوهش از آنجا ناشی می‌شود که هیچیک از رویکردهای پیشین نتوانسته‌اند به‌صورت هم‌زمان سه ویژگی حیاتی سبک‌وزنی، پیش‌بینی‌پذیری و استقلال کنترلی را در چارچوبی واحد و قابل‌پیاده‌سازی فراهم آورند. در نتیجه، روش پیشنهادی با ارائه‌ی مدلی پیش‌بینی محور، مقیاس‌پذیر و مستقل از کنترلر مرکزی، گامی مؤثر در جهت پرکردن شکاف علمی موجود و ارتقای کارایی سیستم‌های Fog-DSDN در شرایط واقعی شبکه برداشته است.

این مقاله در ۷ بخش تدوین شده است. در بخش مقدمه تلاش شده است به صورت خلاصه مبانی پژوهش بررسی شده و چالش‌های پیش روی آن بیان شود. در بخش دوم کارهای مرتبط مورد بررسی قرار گرفته‌اند. در بخش سوم اهمیت و ضرورت پژوهش توضیح داده شده است. بخش چهارم به بیان روش پیشنهادی اختصاص دارد. در بخش پنجم نتایج و شبیه‌سازی‌ها آورده شده‌اند و بخش ششم نتایج کلی کار با یکدیگر مقایسه شده یک نتیجه‌گیری کلی از نتایج حاصل شده، انجام شده است. در نهایت در بخش پایانی پیشنهادهایی برای پژوهشگران و علاقمندان به مبحث مدیریت منابع در شبکه‌های Fog-DSDN آورده شده است.

۲- پیشینه پژوهش

Fog-DSDN یکی از فناوری‌هایی است که به واسطه سیاست‌های درست، در سراسر جهان و در جوامع علمی به طور گسترده پذیرفته شده است. با قدرت و توانایی بی‌شمار Fog-DSDN، این قابلیت وجود دارد در هر زمان و مکان به هر شبکه یا سرویسی متصل شد. Fog-DSDN در حال تبدیل‌شدن به یک ابرقدرت برای ماشین‌آلات نسل بعدی است و تأثیر آن در تجارت فعلی غیرقابل‌انکار است. با کمک Fog-DSDN، صنایع یا محققان درحال‌توسعه راه‌حل‌هایی برای چالش‌های مختلف زندگی روزمره انسان‌ها هستند. در Fog-DSDN، دستگاه‌ها الکترونیکی هوشمند (با یکپارچه‌سازی زیرساخت اینترنتی موجود) برای استفاده مناسب از منابع تعامل خواهند داشت. همچنین این دستگاه‌ها قادر به گسترش خدمات و مزایا برای سیستم‌های هوشمند می‌باشند. Fog-DSDN فراتر از سناریوهای M2M[5] است، از جمله اتصال سریال بین شبکه‌ها و دستگاه‌ها برای ارائه خدمات تحت استرس و... در این بخش باسیاست انتخاب مقالات مرتبط با پژوهش، برخی از مقالات انجام شده در زمینه معماری‌های نوین در Fog-DSDN انتخاب شده‌اند [۹].

احمد و همکاران در پژوهش [۱۰] ترکیب محاسبات مه و شبکه‌های نرم‌افزار محور برای بهبود کیفیت خدمات در سامانه‌های اینترنت اشیا ارائه

1. Fog-Software Defined Distributed Network
2. Microservice-Based Task Distribution
3. Echo State Network
4. Fog-Software Defined Distributed Network
5. Machine-to-Machine

را دادند. این روش با تفکیک منطقی کنترل‌کننده SDN، بهینه‌سازی انتخاب نقاط دسترسی و مکان پردازش داده‌ها را هدف قرار داده و منجر به کاهش تأخیر، استفاده از شبکه و مصرف انرژی شده است. نتایج شبیه‌سازی در محیط iFogSim نشان‌دهنده کاهش سه‌برابری تأخیر و بهبود چشمگیر پارامترهای QoS (شامل هزینه، مصرف توان و تأخیر) است. با وجود پیچیدگی زیرساخت و وابستگی به کنترل‌کننده مرکزی، این رویکرد در کاربردهایی نظیر IoMT[1]، صنعت ۴٫۰، Green IoT و شبکه‌های 5G مؤثر ارزیابی شده است.

از سوی دیگر، اوگاندوین و کمیل در مطالعه‌ای جامع، تکنیک‌های بهینه‌سازی در رایانش مه را دسته‌بندی و کاربرد آن‌ها را در مسائل مختلفی همچون تخصیص منابع، کش‌سازی داده و کشف خدمات بررسی کرده‌اند [۱۱]. این پژوهش با ارائه معیارهای ارزیابی مانند تأخیر، مصرف انرژی و زمان پاسخ، به شکاف‌های پژوهشی موجود اشاره کرده و مسیرهای تحقیقاتی آینده را مشخص ساخته است. در ادامه، المهتادی و همکاران، مدلی مبتنی بر منطق ذهنی برای ارزیابی اعتماد در شبکه‌های مه ارائه دادند که با محاسبه اعتماد پویا، امنیت ارتباطات را در محیط‌های ناهمگن افزایش می‌دهد [۱۲]. همچنین، کادیم و همکاران در [۱۳] با ارائه الگوریتم DMPFS[2] در شبکه‌های خودرویی با استفاده از رایانش مه، SDN و خودروهای پارک‌شده، موفق به کاهش چشمگیر تأخیر، افزایش نسبت تحویل بسته و کاهش سربار ارتباطی شدند. با این حال، پیچیدگی محاسباتی و وابستگی به کنترل‌کننده مرکزی از چالش‌های اصلی این روش به شمار می‌روند.

داراخ و همکاران پروتکل مسیریابی نوآورانه CDRP-F-SDVN[3] را برای شبکه‌های خودرویی طراحی کرده‌اند که با ترکیب محاسبات مه و شبکه‌های نرم‌افزار محور وسایل نقلیه و بهره‌گیری از معماری سلسله‌مراتبی، هوشمندی و انعطاف‌پذیری بالایی در مدیریت ترافیک ایجاد می‌کند. این پروتکل با خوشه‌بندی مبتنی بر سرعت و موقعیت، فاز دومسیره جهت رفع خرابی کنترلر SDN، و کاهش بار کنترلی عملکرد چشم‌گیری نسبت به پروتکل‌های سنتی داشته است، از جمله بهبود ۹۶٫۵۳٪ در توان عملیاتی و کاهش ۹۵٫۶٪ در تأخیر انتها به انتها [۱۴]. در ادامه، حسینی و همکاران با ارائه رویکرد PQFAHP[4] که ترکیبی از AHP[5]، منطق فازی و صف‌های اولویت‌دار است، زمان‌بندی وظایف در محیط رایانش مه‌آلود سیار را بهینه کرده‌اند؛ این روش تا ۴۹٪ تأخیر، ۲۶٪ انرژی و ۱۰٫۸٪ سطح خدمات را بهبود داده و با در نظر گرفتن چندین معیار کلیدی، دقت تصمیم‌گیری را بالا برده است [۱۵].

عزیزی و همکاران در [۱۶] با مدل‌سازی زمان‌بندی وظایف به صورت یک برنامه‌ریزی غیرخطی عدد صحیح مختلط و ارائه دو الگوریتم نیمه‌حریصانه PSG[6] و PSG-M[7]، توانسته‌اند نرخ انجام وظایف در مهلت مهلت مقرر را ۱٫۳۵ برابر افزایش و نقض مهلت‌ها را تا ۹۷٫۶٪ کاهش دهند. این الگوریتم‌ها با ترکیب مدل دقیق و روش‌های سریع، بهره‌وری منابع و کیفیت خدمات را در محیط‌های IoT بهبود داده‌اند. در همین راستا، سرما و همکاران در [۱۷] چارچوبی ترکیبی برای مدیریت منابع در محیط‌های Fog مبتنی بر SDN با استفاده از یادگیری ماشین ارائه داده‌اند که با استفاده از الگوریتم‌هایی مانند گرادیان کاهشی[8]، دیکسترا، گاوسی نیو بیز[9] و K-Means++ موفق به بهبود تخصیص منابع و کاهش نرخ شکست شده است. در نهایت، اسلام و همکاران با مرور نظام‌مند مطالعات مرتبط با جای‌گذاری بهینه برنامه‌ها در رایانش مه، یک چارچوب ساخت‌یافته برای تحلیل الگوریتم‌ها و فناوری‌های مورد استفاده ارائه کرده‌اند و به کمبود روش‌های تطبیقی حساس به تغییرات محیطی و تفاوت معماری مه و ابر در تحقیقات گذشته اشاره نموده‌اند [۱۸].

علی و همکاران در [۱۹] چارچوبی نوین با عنوان SDN-REMR[10] برای بهبود انتقال پیام‌های اضطراری در شبکه‌های بین خودرویی پیشنهاد داده‌اند. این چارچوب با ترکیب فناوری‌های شبکه نرم‌افزار محور، دوقلوی دیجیتال[11] (DT) و رایانش مه، به بهینه‌سازی انتخاب گره‌های پایدار و مسیریابی پیام‌ها در محیط‌هایی با تحرک بالا می‌پردازد. از قابلیت SCF[12] برای افزایش نرخ تحویل در شرایط نامطلوب شبکه استفاده شده و با پیش‌بینی پایداری گره‌ها، ارتباطات قابل اعتمادتری برقرار می‌گردد. به‌کارگیری DT نیز امکان پیش‌بینی رفتار گره‌ها بر اساس سرعت و مکان را فراهم کرده است. نتایج نشان‌دهنده بهبود چشم‌گیر در توان عملیاتی، نرخ تحویل بسته‌ها و کاهش سربار مسیریابی هستند. با این حال، روش پیشنهادی با چالش‌هایی نظیر پیچیدگی محاسباتی بالا، وابستگی به زیرساخت SDN، و کاهش کارایی در سناریوهای واقعی مواجه است. متین فر و همکاران در [۲۰] روشی نوین برای مدیریت منابع در شبکه‌های SDVN به نام FABLSM[13] ارائه دادند. در این روش با بهره‌گیری از یک از یک BLSTM[14] مبتنی بر لایه توجه چند سر، متغیرهای تصمیم‌گیری در مورد اعطای منابع به گره‌ها از طریق توجه ویژگی‌های زمینه‌ای شبکه انجام می‌گیرد. این کار موجب می‌شود بهره‌وری اعطای منابع به میزان قابل توجهی افزایش یابد.

در مقابل، لی و همکاران در [۲] با بهره‌گیری از یادگیری تقویتی عمیق و شبکه‌های CNN-LSTM، روشی برای مکان‌یابی بهینه کنترلر SDN و انتخاب دینامیک لبه در شبکه‌های MEC[15] پیشنهاد داده‌اند که منجر به کاهش مصرف انرژی و تأخیر ارتباطی شده است. همچنین، ژو و همکاران در [۳] با استفاده از الگوریتم PPO[16] و مدل‌سازی مارکوف، تخصیص منابع را بهینه کرده و تعادل بار را در شبکه‌های SDN مبتنی بر محاسبات لبه بهبود داده‌اند. در نهایت، محمودی و همکاران در [۴] با معرفی روش CAAF-DSDN، توازن بار را از طریق خوشه‌بندی تطبیقی و مدل‌سازی جبری در مراکز داده بهبود بخشیده‌اند. روش آن‌ها با کاهش تأخیر و زمان پاسخ و همچنین ساده‌سازی محاسبات نسبت به الگوریتم‌های RL، کارایی بهتری نشان داده است، اگرچه عملکرد آن وابسته به دقت انتخاب پارامترهای اولیه خوشه‌بندی است.

1. Internet of Medical Things
2. Delay Efficient Multicasting Approach Depending on Parked Vehicles, FC
3. Innovative Cluster-Based Dual-Phase Routing Protocol for SDVN
4. Priority Queue, Fuzzy and Analytical Hierarchy Process
5. Analytical Hierarchy Process
6. Priority-Aware Semi-Greedy
7. PSG with Multi-Start Procedure
8. Gradient Descent
9. Gaussian Naive Bayes
10. SDN Based on Reliable Emergency Message Routing Schema
11. Digital Twin
12. Store-Carry-Forward
13. Federated Adaptive Attention-Based BLSTM
14. Bidirectional Long Short-Term Memory
15. Multi-Access Edge Computing
16. Proximal Policy Optimization

جدول ۱: طراحی معماری در Fog-DSDN

مرجع	مدیریت مصرف حافظه	مدیریت مصرف منابع پردازشی	افزایش سرعت دسترسی	کاهش پیچیدگی طراحی
[۱۰]	☑	☒	☑	☒
[۱۲]	☒	☒	☑	☑
[۱۱]	☒	☒	☒	☑
[۱۶]	☒	☒	☒	☒
[۱۴]	☑	☑	☑	☒
[۱۳]	☑	☑	☒	☒
[۱۸]	☒	☑	☑	☑
[۱۹]	☑	☑	☑	☒
[۳]	☒	☒	☑	☑
[۴]	☑	☑	☑	☒
روش پیشنهادی	☑	☑	☑	☑

یک مقایسه موردی باتوجه‌به اهداف پژوهش در جدول ۱ بین پژوهش‌های بررسی‌شده انجام گرفته است.

در یک جمع‌بندی کلی در این بخش مبانی شبکه‌های Fog-DSDN بررسی و چالش‌های موجود بر سر راه بهبود کارایی این شبکه‌ها بررسی شد. حل چالش‌های ذکر شده این بخش یک افق تحقیقاتی پیشروی محققان در صنعت، کاربردهای نظامی، بهداشتی و ... گشوده است. اهمیت موضوع پژوهش در زمینه Fog-DSDN محققان را بر این داشته تا راه‌حل‌های متنوعی برای چالش‌های بیان شده ارائه دهند.

۳- روش پیشنهادی و ساختار سیستم

روش پیشنهادی یک نقشه اطلاعاتی پیشنهاد می‌دهد که سعی دارد با درنظرگرفتن مسائل مربوط به مدیریت منابع، راهکارهای لازم را برای گره‌های Fog-DSDN ارسال کند. در این روش برای ایجاد این منبع میکروسرویس‌ها به‌کارگرفته‌شده تا منبع اطلاعات را در یک مجتمع بسازند، سپس اطلاعات از طریق یک ESN دوبعدی پردازش شده و در هنگام درخواست گره، اطلاعات نقشه اطلاعاتی بروز خواهد شد؛ تا بتوان از این طریق دستورات لازم، برای گره Fog-DSDN ایجاد نمود. سپس اشیا شبکه، با استفاده از بخش مربوط به خود اطلاعات لازم برای اقدامات بعدی را از روی نقشه جامع خود برخواهند داشت. از منظر امنیت و حریم خصوصی، تفکیک وظایف در قالب میکروسرویس‌ها باعث می‌شود که هر سرویس فقط به داده‌های مرتبط با نقش خود دسترسی داشته باشد. ارتباط میان سرویس‌ها از طریق درگاه‌های امن[1] انجام می‌شود و امکان اعمال سیاست‌های کنترل دسترسی و رمزگذاری داده در سطح سرویس فراهم است.

در همین راستا این بخش در دو قسمت تدوین شده است:

- بیان ریاضی کارایی در Fog-DSDN
- تشریح روش پیشنهادی

در جدول ۲ برای راحتی در استفاده، تمام متغیرهای مورد استفاده در این بخش معرفی شده است.

جدول ۲: جدول متغیرهای مورد استفاده.

عنوان متغیر	توضیحات
P_n	کارایی سیستم مدیریت منابع
R_M	میزان منابع حافظه‌ای
R_P	میزان منابع پردازشی
R_T	میزان زمان مصرفی
$d_{n(ij)}$	داده نرمال شده از سطر i ام و ویژگی j ام
d_{ij}	میزان واقعی داده از سطر i ام و ویژگی j ام (داده غیرنرمال)
$\min_j$	کمینه داده‌ها از ستون j ام
$\max_j$	بیشینه داده‌ها از ستون j ام
W_1, W_2, W_3	ضرایب هر کدام از داده‌های R_M، R_P و R_T
A	محیط عملیاتی روش پیشنهادی
M	طول محیط عملیاتی
n	عرض محیط عملیاتی
I_{ij}	گره Fog-DSDN موجود در طول i و عرض j
p	تعداد مسائل موردبررسی در محیط عملیاتی
a_k	واحد داده ارسالی به مرکز داده‌ها
a_{out}	خروجی در صف خروجی در زمان t
a_{in}	ورودی در صف ورودی در زمان t
$a^t_{mean\ out}$	میانگین صف خروجی در زمان t
$a^{t-1}_{last\ out}$	مقدار قبلی موجود در سلول اطلاعات ij در زمان $t-1$
w^t_k	ماتریس ضریب اهمیت داده ها در مجموعه خروجی a_{out} در زمان t (وابسته به شرایط محیط)
$E^{ij}(t-1)$	میزان خطای قبلی گره در زمان $t-1$

۳-۱ بیان ریاضی کارایی در Fog-DSDN

کارایی یک شبکه Fog-DSDN خصوصاً در مدیریت منابع، به محیط عملیاتی آن وابسته است، به این مفهوم که گاهی یک سیستم کارا سیستمی است که می‌تواند با نرخ ارسال بالا و با کمترین مقدار تأخیر ممکن به درخواست‌ها پاسخ دهد؛ یا در تعریف دیگر سیستمی کارا خواهد بود که بتواند مقدمات لازم را برای ارائه خدمات هرچه بیشتر و کیفیت سرویس هرچه بالاتر فراهم کند. بنا بر همین منطق در این پژوهش مفهوم کارایی، باتوجه‌به میزان اهداف مشخص شده در بخش اول به‌صورت تابعی از سه متغیر تعریف می‌گردد:

- منابع حافظه‌ای R_M: میزان حافظه‌ای که سیستم برای اجرا به آن نیازمند است. واحد سنجش این متغیر MB است.
- منابع پردازشی R_P: میزانی از منابع پردازشی که در مجموع، یک روش برای اجرای کامل به آن نیازمند است. این متغیر با واحد تعداد دستور پردازش شده سنجیده می‌شود.
- منابع زمانی R_T: مدت زمانی که طول می‌کشد تا روش مدیریت منابع، به اهداف خود دست پیدا کند. بدیهی است واحد سنجش این متغیر نیز ثانیه باشد.

در واقع سیستمی کاراست که سه متغیر ذکر شده را با کمترین هزینه ممکن، کاهش دهد. با یک ترکیب ساده و منطقی از موارد بالا به صورت جمع کلیه پارامترها می‌توان به (۱) رسید.

1. API Gateway

$$P_n \propto \frac{۱}{R_M}, \frac{۱}{R_P}, \frac{۱}{R_T} \quad (۱)$$

که در آن P_n، کارایی سیستم مدیریت منابع در شبکه Fog-DSDN، R_M، میزان منابع حافظه‌ای مصرفی، R_P، میزان منابع پردازشی مصرفی و R_T، میزان زمان مصرفی می‌باشد.

ازآنجاکه متغیرهای ذکر شده فاقد یک واحد یکسان هستند، جمع سه متغیر ذکر شده در یک رابطه امکان‌پذیر نخواهد بود؛ بنابراین قبل از انجام هر کاری باید مقادیر ذکر شده برای هر متغیر را در یک دستگاه واحد نگاشت نمود. این عمل از طریق نرمال‌سازی موجود در (۲) انجام خواهد گرفت. این رابطه با شناسایی حداکثر/ حداقل مقادیر هر یک از متغیرهای ذکر شده، به یک سیستم عددی ۰ تا ۱ می‌رسد.

$$d_{n(ij)} = \frac{d_{ij} - \min_j}{\max_j - \min_j} \quad (۲)$$

که در آن $d_{n(ij)}$، داده نرمال شده از سطر iام و ویژگی jام، d_{ij}، میزان واقعی داده از سطر iام و ویژگی jام و $\max_j$ و $\min_j$ به ترتیب بیشترین و کمترین مقدار ستون jام در داده‌ها می‌باشد.

برای تبدیل رابطه تناسب به یک رابطه تساوی با علمِ به اهمیت هر یک از متغیرهای ذکر شده در (۳) می‌توان ضرایب $W_۱, W_۲, W_۳$ را تعریف نمود به شکلی که

$$P_n = W_۱ \frac{۱}{R_M} + W_۲ \frac{۱}{R_P} + W_۳ \frac{۱}{R_T} \quad (۳)$$

در این پژوهش هدف نهایی، افزایش میزان P_n به بیشترین مقداری که می‌تواند داشته باشد.

۳-۲ تشریح روش پیشنهادی

برای افزایش میزان کارایی در این بخش به بررسی روش پیشنهادی پرداخته می‌شود. روش پیشنهادی برای مدیریت منابع مصرفی دارای سه بخش اصلی است:

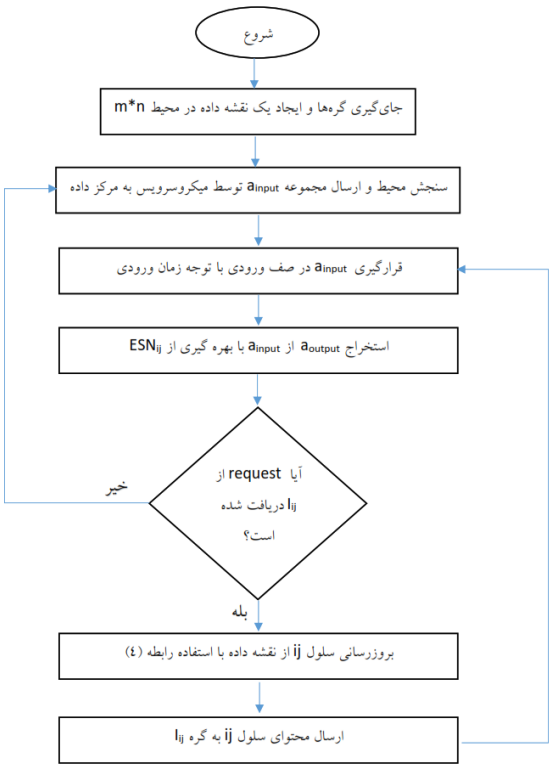
- ایجاد نقشه اطلاعات توسط میکروسرویس‌ها
- بررسی اطلاعات در هر سلول اطلاعاتی
- جایگزینی و بروزرسانی اطلاعات

در نمودار شکل ۱ نحوه فعالیت روش پیشنهادی به نمایش درآمده است.

۳-۲-۱ ایجاد نقشه اطلاعات توسط میکروسرویس‌ها

پیش از پرداختن به موضوع باید مسئله را در یک ساختار ریاضی پیاده‌سازی کنیم. برای این کار نیاز است فرض‌های زیر، در نظر گرفته شود:

- محیط عملیاتی روش پیشنهادی (A) در یک محیط $m \times n$ تقسیم شده است.
- در هر قسمت از محیط عملیاتی، در منطقه طول i و عرض j، یک گره Fog با عنوان I_{ij}
- در کل منطقه عملیاتی تعداد p حالت از یک مسئله وجود دارد که هدف شبکه، ایجاد ارائه راهکار مناسب برای آن p حالت است. باید توجه داشت که p تعداد «سناریوهای کاری» یا «حالات شبکه» است که الگوریتم باید برای آن‌ها تصمیم تخصیص منابع بگیرد.



شکل ۱: شمای روش پیشنهادی.

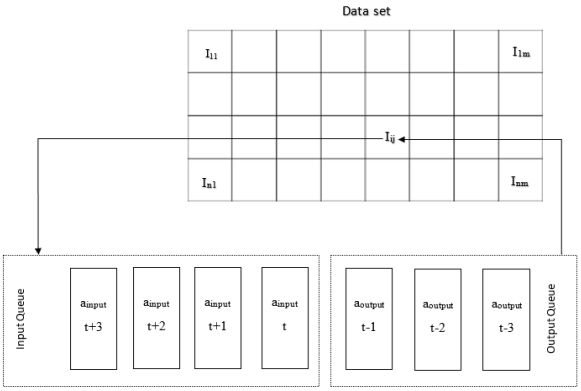
- هر داده‌ای که یک I_{ij} برای مرکز اطلاعات ارسال می‌کند از طریق یک میکروسرویس که آن را به‌صورت مجموعه از $a_{ij} = (a_۱, a_۲, ..., a_k)$ درآورده است، از طریق فرمت حالت Get به مرکز اطلاعات ارسال می‌گردد.
- تعداد میکروسرویس‌های فعال در هر گره I_{ij} بستگی به سخت‌افزار گره دارد و بین p تا $۲p$ است.

مجموعه‌داده مرکزی از به $m \times n$ قسمت تقسیم شده است و هر قسمت مطابق شکل ۲ دارای ۳ قسمت است.

- صف بافر ورودی از میکروسرویس‌های گره‌های ز
- صف بافر آماده به ارسال به گره I_{ij}

هر میکروسرویس در زمان t وظیفه ارسال اطلاعات به صف ورودی را دارد، هر a_{in} در زمان t به انتهای صف ورودی وارد خواهد شد و منتظر پردازش خواهد شد.

به عبارت دیگر در این مرحله برای مدلسازی ریاضی روش پیشنهادی، فرض می‌شود محیط عملیاتی به‌صورت شبکه‌ای دوبعدی با ابعاد $m \times n$ تقسیم‌بندی شده است، به‌گونه‌ای که در هر سلول (i, j) یک گره مه با شناسه I_{ij} قرار دارد. هر گره مجموعه‌ای از میکروسرویس‌های سبک را اجرا می‌کند که وظیفه جمع‌آوری، پیش‌پردازش و ارسال داده‌ها به مرکز تصمیم‌گیری را دارند. داده‌های تولیدی گره I_{ij} به‌صورت برداری از ویژگی‌های a_{ij} تعریف می‌شود که در آن هر مؤلفه بیانگر مقدار یک شاخص محیطی یا عملکردی است. داده‌های جمع‌آوری‌شده در هر گره از طریق یک فرمت استاندارد GET به مرکز اطلاعات ارسال می‌شوند و در قالب صف ورودی Q_{in} ذخیره می‌گردند. پس از پردازش اولیه و پیش‌بینی بار توسط مدل ESN، خروجی‌های تولیدی در صف خروجی Q_{out} آماده تخصیص به گره‌های هدف می‌شوند. در این ساختار، تعداد میکروسرویس‌های فعال در هر گره وابسته به ظرفیت سخت‌افزاری آن بوده و در بازه‌ای بین p تا $۲p$ تغییر می‌کند، تا انعطاف‌پذیری در مقیاس‌پذیری سیستم حفظ شود. در نهایت، داده‌های گردآوری‌شده از تمام



شکل ۲: نوع چینش اطلاعات و ساختار یک واحد از مرکز داده جامع.

سلول‌های $m \times n$ در قالب نقشه اطلاعاتی ترکیب می‌شوند که مبنای تصمیم‌گیری تطبیقی برای تخصیص منابع در سطح کنترل‌کنندهمه است. این مدل ریاضی امکان تحلیل کمی رفتار شبکه، بهینه‌سازی صف‌های ورودی و خروجی، و ارزیابی عملکرد الگوریتم پیشنهادی در سناریوهای پویا را فراهم می‌سازد. در انتها در مرکز اطلاعات نقشه اطلاعاتی مانند آنچه که در شکل ۲ قابل نمایش است ایجاد خواهد شد.

۳-۲-۲ بررسی اطلاعات در هر سلول اطلاعاتی

یک سیستم مدیریت منابع باید قابلیت برخط داشته باشد. برای مشخص‌کردن هر یک از خروجی‌های a_{out} یک ورودی از صف a_{in} وارد شبکه ESN_{ij} مختص سلول (i,j) (سلولی از اطلاعات که مختص گره I_{ij} است) شده و خروجی a_{out} را تولید می‌نماید. به بیان دیگر هر سلول دارای ESN محلی است که به‌صورت مستقل وضعیت گره خود را پیش‌بینی می‌کند. خروجی ESN تابعی از ورودی و وزن‌های داخلی مخزن است. در هر لحظه‌ی زمانی t، داده‌ی ورودی مربوط به سلول (i,j) که از صف ورودی استخراج می‌شود، به مدل محلی شبکه حالت پژواک (ESN_{ij}) ارسال می‌گردد. این مدل با استفاده از وضعیت فعلی مخزن $(x^{ij}(t))$ و وزن‌های یادگرفته‌شده‌ی خود، خروجی $a_{out}^{ij}(t)$ را بر اساس تابع (۴) تولید می‌کند

$$a_{out}^{ij}(t) = f_{\mathrm{ESN}}(a_{in}^{ij}, W_{res}^{ij}) \qquad (۴)$$

که در آن $a_{out}^{ij}(t)$ خروجی سلول (i,j) در زمان t، a_{in}^{ij} ورودی میکروسرویس در ناحیه (i,j)، W_{res}^{ij} وزن‌های یادگیری در ESN_{ij} و $f_{\mathrm{ESN}}(...)$ ماژول محاسبه ESN می‌باشد. خروجی تولیدشده در صف خروجی Q_{out}^{ij} ذخیره می‌شود تا میکروسرویس‌های محلی بر اساس آن عملیات تخصیص منابع یا تصمیمات کنترلی را اجرا کنند. ویژگی کلیدی این ساختار در توزیع مدل‌های ESN به‌صورت محلی در هر گره Fog نهفته است، که باعث می‌شود هر گره به‌صورت مستقل و تطبیقی رفتار خود را با تغییرات لحظه‌ای شبکه تنظیم نماید. به‌منظور مقابله با نوسانات ناگهانی ترافیک، هر مدل ESN قادر است وضعیت گره خود را به‌صورت بلادرنگ به‌روزرسانی کرده و الگوهای جدیدِ بار را شناسایی کند. این سازوکار یادگیری تطبیقی موجب می‌شود تخصیص منابع در سطح شبکه به‌صورت پویا بازتنظیم شده و از ازدحام، تأخیر پردازش و افت کارایی سیستم جلوگیری گردد. به‌منظور مقابله با تغییرات ناگهانی ترافیک شبکه، مدل ESN مورد استفاده قادر است با بروزرسانی لحظه‌ای وضعیت گره‌ها، الگوهای جدید بار را شناسایی کند. این ویژگی موجب می‌شود سیستم به‌صورت تطبیقی منابع را بازتخصیص داده و از ازدحام یا افت کارایی جلوگیری نماید.

الگوریتم ۱

Resource management (Fog-DSDN nodes I, Micro service ms, Data dataset, Operating environment A)
Input: set of node, set of mc, dataset, area feature, resource feature.
Output: map data, resource management policy.

For all I_{ij}
  get information from A_{ij}
  for all ms_k $\mathrm{ms_k}$ in I_{ij}
    send a_{in} to data map
  end
end
do process for each $\mathrm{cell_{ij}}$
  Put $\mathrm{a_{in}}$ in input queue
  $\mathrm{ESN_{ij}}$ process a_{in} and make a_{out}
  Put a_{out} in output set
while (receive "get request" from I_{ij})
get request from I_{ij}
calculate response from (Eq.5)
send response to I_{ij}
update $cell_{ij}$
update E^{ij}, w
clear both queues
end

شکل ۳: فرایند تخمین میزان منابع مناسب برای هر سوئیچ.

۳-۲-۳ بررسی اطلاعات در هر سلول اطلاعاتی

در ساختار پیشنهادی، داده‌های ذخیره‌شده در صف‌های ورودی و خروجی هر سلول (i,j) تا زمان دریافت درخواست جدید از سوی گره Fog مربوطه در حالت انتظار باقی می‌مانند. به‌منظور اطمینان از آنکه تصمیم تخصیص منابع نه‌تنها بر اساس آخرین داده‌ها بلکه با در نظر گرفتن روند تغییرات مقادیر قبلی اتخاذ شود، خروجی نهایی برای هر گره (۵) محاسبه می‌گردد. این رابطه، تغییرات میانگین خروجی و آخرین مقدار ثبت‌شده را با ضریب اهمیت داده‌ها و خطای پیشین گره ترکیب می‌کند تا نرخ به‌روزرسانی به‌صورت تطبیقی و برخط انجام گیرد.

$$a_{out}^{ij}(t+1) = \frac{1}{2}\left(a_{mean}^{ij}(t) - a_{last}^{ij}(t-1)\right)^2 w_k^{ij}(t) + E^{ij}(t-1) \qquad (۵)$$

که در آن به‌ترتیب $a_{out}^{ij}(t+1)$ مقدار خروجی نهایی برای سلول (i,j) در زمان $t+1$، $a_{mean}^{ij}(t)$ میانگین مقادیر موجود در صف خروجی در زمان t، $a_{last}^{ij}(t-1)$ آخرین مقدار ذخیره‌شده در سلول اطلاعاتی در زمان $t-1$، $w_k^{ij}(t)$ ماتریس ضرایب اهمیت داده‌ها در زمان t (بر اساس شرایط لحظه‌ای شبکه به‌صورت تطبیقی تنظیم می‌شود) و $E^{ij}(t-1)$ خطای پیش‌بینی قبلی خروجی در زمان $t-1$ هستند. پس از محاسبه‌ی $a_{out}^{ij}(t+1)$ و ارسال پاسخ به گره مربوطه، حافظه سلول به‌روزرسانی شده و محتوای صف‌های ورودی و خروجی پاک‌سازی می‌شود تا فضای لازم برای داده‌های جدید فراهم گردد. این فرآیند باعث می‌شود سیستم همواره در وضعیت بروز، پویا و مقاوم نسبت به تغییرات لحظه‌ای بار شبکه باقی بماند. الگوریتم کلی روند بروزرسانی و مدیریت منابع در شکل ۳ تشریح گردیده است.

به‌صورت خلاصه، در این مرحله در هر لحظه‌ی زمانی t، بردار ورودی $a_{in}^{ij}(t)$ شامل مقادیر فعلی مصرف منابع و پارامترهای محیطی به مدل ESN_{ij} داده می‌شود. الگوهای زمانی مصرف از طریق تغییرات متوالی در

حالت مخزن یاد گرفته می‌شوند و مدل بدون نیاز به بازآموزی سنگین قادر است رفتار آینده گره را پیش‌بینی کند. خروجی $t+1$ شامل منابع مصرفی آینده در لحظه $t+1$، به‌صورت مستقیم برای تصمیم‌گیری تخصیص تطبیقی منابع به کار می‌رود تا از ازدحام، کمبود منابع و تأخیر در پاسخ جلوگیری شود. در واقع مدل ESN با تحلیل الگوهای زمانی مصرف در گذشته، تغییرات آینده منابع را بر اساس همبستگی‌های پویا میان داده‌ها پیش‌بینی کرده و تصمیمات تخصیص را به‌صورت برخط و تطبیقی تنظیم می‌کند.

همان‌طور که از این الگوریتم پیداست مراحل در ۳ واحد:

- میکروسرویس
- گره‌های I_{ij}
- مرکز کنترل شبکه (کنترل‌کننده‌های SDN)

انجام می‌شود. این مراحل شامل جمع آوری، نظم‌دهی، نتیجه‌گیری و اعلام نتایج (بروزرسانی اطلاعات نقشه داده) به هر گره است.

باید توجه داشت که در هر سلول اطلاعاتی، دو صف مستقل برای داده‌های ورودی و خروجی تعریف شده است. مدیریت این صف‌ها بر عهده میکروسرویس محلی است که با استفاده از سیاست زمان‌بندی تطبیقی و بر مبنای پیش‌بینی بار توسط شبکه ESN، اولویت اجرای وظایف را تعیین می‌کند. اطلاعات مربوط به طول صف‌ها و وضعیت اجرای وظایف به‌صورت بلادرنگ در نقشه اطلاعاتی ثبت می‌شود. در صورت افزایش غیرمنتظره بار، وظایف به سلول‌های مجاور با ظرفیت آزاد مهاجرت می‌کنند. این مکانیسم موجب کاهش تراکم صف‌ها و در نتیجه کاهش زمان پاسخگویی سامانه می‌شود.

٤- نتایج و شبیه‌سازی

سیستم‌های کنترل منابع و ارتباط در یک شبکه Fog-DSDN سیستم‌های هوشمندی هستند که شامل روش‌های هدفمند برای ایجاد یک ارتباط مناسب بین گره‌ها با کمترین ضریب خطا و پایین‌ترین میزان مصرف منابع هستند. این روش‌ها می‌توانند بین انواع فاکتورهای مختلف از یک شبکه Fog-DSDN، یک رابطه متناسب به وجود بیاورند. برای مثال با درنظرگرفتن زمان پردازش، سعی می‌کنند کمترین میزان خطا در ارسال و دریافت داده‌ها را تضمین نمایند. استفاده از معماری‌های موجود در شبکه Fog-DSDN یکی از راه‌حل‌های ممکن برای ایجاد این رابطه تناسب است.

با این مقدمه بیان می‌شود که چرا روش پیشنهادی می‌تواند یک تناسب خوب بین منابع مصرفی، منابع حافظه‌ای و زمان مصرفی در یک اتصال بین گره‌های شبکه Fog و مرکز کنترل شبکه، ایجاد نماید، بنابراین این قسمت به دو بخش تقسیم شده است:

- محیط و شرایط شبیه‌سازی
- نتایج و مقایسه‌ها

همان‌طور که از نام قسمت‌ها می‌توان تشخیص داد، هدف از بخش نخست معرفی محیط شبیه‌سازی شامل: نرم‌افزار و سخت‌افزار شبیه‌سازی، متغیرهای مورداستفاده و مجموعه‌داده است. در بخش دوم نتایج حاصل از روش پیشنهادی با ۲ بستر پایه و یک روش مدیریت منابع TFS[1] [۲۱] در بستر Fog-DSDN، مقایسه شده و میزان کارایی هر روش، نسبت به روش پیشنهادی، محاسبه و مقایسه می‌گردد. پیش از شروع این بخش به بررسی دو بستر مورد مطالعه پرداخته می‌شود.

1. Traffic Forecasting Native SDN

معماری میکروسرویس پایه: در معماری میکروسرویس پایه[2] ساختار ماژولار میکروسرویس را حفظ می‌کند؛ اما ویژگی‌های پیش‌بینی گر و زمان‌بندی تطبیقی روش پیشنهادی غیرفعال‌اند. مشخصه‌های پیاده‌سازی عبارت‌اند از:

- تقسیم وظایف به میکروسرویس‌های مستقل برای جمع‌آوری و ارسال داده
- وجود یک کنترل‌کننده مرکزی سبک که فقط عملیات هماهنگی ساده را انجام می‌دهد (بدون الگوریتم تخصیص پیش‌بینی‌شونده)
- صف‌بندی FIFO[3] محلی برای پردازش درخواست‌ها
- عدم وجود مهاجرت وظایفِ خودکار یا بازتخصیص مبتنی بر مدل یادگیرنده.

این روند برای سنجش اثر ماژولار بودن معماری در نبود مکانیزم‌های هوشمند طراحی شده است و بر اساس پژوهش ادانازا و همکاران طراحی شده است [۲۱].

معماری پایه Fog-DSDN: در این حالت، هیچ‌گونه مکانیزم مدیریتی متمرکز یا توزیع‌شده برای تخصیص منابع وجود ندارد. این حالت بر اساس معماری پایه Fog-DSDN ارائه شده توسط متین‌فر و همکاران ارائه شده است. مشخصه‌های این معماری شامل موارد زیر هستند:

- هر گره به‌صورت مستقل و فقط بر اساس توان محلی خودپردازش می‌کند
- هیچ ساختار میکروسرویسی وجود ندارد، پردازش‌ها به‌صورت یکپارچه[4] (تک بلوکی) محلی اجرا می‌شوند
- صف‌بندی حداقلی (بدون اولویت‌بندی و بدون صف خروجی تجمیعی)
- هیچ‌گونه پیش‌بینی بار، مهاجرت وظیفه یا هماهنگی کنترلری اعمال نمی‌شود.

این حالت معیار پایه برای ارزیابی سود حاصل از طراحی معماری و مکانیزم‌های پیش‌بینی و تخصیص است [۸].

٤-١ بررسی محیط و شرایط شبیه‌سازی

یک شبیه‌سازی دارای استانداردها، و شرایط و محدودیت‌های ویژه‌ای است. در یک شبیه‌سازی، سناریوهای انتخاب شده برای آزمایش‌ها، باید علاوه بر سنجش متغیرهای موردنظر، نکات زیر را نیز در نظر داشته باشد:

- ساختار سناریوهای موردبررسی باهدف سنجش فرضیه‌ها و متناسب با ویژگی‌ها و رفتارهای کلیدی پدیده موردنظر باشد.
- تقریب‌ها، مفروضات و محدودیت‌ها در شبیه‌سازی باید به شکل درست استفاده گردند. به این معنی که استفاده از یک نوع تقریب نباید بر سنجش دقت متغیرها اثر سو داشته باشد، یا یک محدودیت نباید رفتار یک متغیر در حال عادی را تحت‌تأثیر قرار دهد.
- نتایج آزمایش‌ها باید نشان‌دهنده تغییرات متغیرها در شرایطی معتبر و قابل‌استناد باشند.

در راستای دستیابی به اهداف ذکر شده در بخش نخست و مقایسه نتایج بررسی فرضیه‌های پژوهش، طراحی یک محیط با شرایط برابر، برای انجام شبیه‌سازی هر ایده، امری ضروری است. ازاین‌رو این بخش به بیان محیط، متغیرها و شرایط شبیه‌سازی خواهد پرداخت.

2. Microservice Baseline
3. First In, First Out
4. Monolithic

٤-١-١ محیط شبیه‌سازی

OMNeT++[1] یک کتابخانه شبیه‌سازی مؤلفه محور می‌باشد که با زبان C++ نوشته می‌شود و برای شبیه‌سازی شبکه‌های ارتباطی مانند متحرک، شبکه‌های تلفن همراه، شبکه‌های مبتنی اینترنت اشیا و... به کار می‌رود. از بستر شبیه‌سازی OMNeT++ می‌توان برای مدل‌سازی ترافیک شبکه اینترنت اشیا، بررسی ازدحام کانال‌ها ارتباطی این شبکه، مدل‌سازی پروتکل‌های مهم انواع شبکه‌ها، مدل‌سازی ریزپردازنده و سایر مؤلفه‌های سخت‌افزاری موجود در شبکه‌های موردمطالعه، بررسی میزان مصرف انرژی توسط گره‌ها، وضعیت کانال‌های ارتباطی، بالانس بار در شبکه‌های مختلف و ... استفاده نمود.

OMNeT++ در مقایسه با نرم‌افزارهای مشابه از جمله NS2، OPNET و ... حافظه موقت (RAM) پایین‌تری نیاز دارد، منابع پردازشی کمتری استفاده می‌کند و در اکثر سیستم‌عامل‌ها مانند ویندوز، لینوکس و مکینتاش قابل‌نصب و بهره‌برداری است.

علاوه بر این، از دیگر مزیت‌های OMNeT++، می‌توان به امکان دریافت خروجی زبان‌های C، C++ و در سال‌های اخیراً پایتون (با انتشار نسخه ۶) و همچنین نمایش خروجی‌های گرافیکی در قالب نمودار و منحنی از کدهای C++ نام برد.

کدهای OMNeT++ با کمک APIها به‌سهولت با زبان‌های دیگر همچون Java و C ترکیب شده و می‌توان از آن‌ها در بسترهای مختلف شبیه‌سازی شبکه اینترنت اشیا بهره برد. همچنین با افزودن فریم‌ورک‌هایی مانند INET و MiXiM، می‌توان از OMNeT++ برای شبیه‌سازی شبکه‌های بزرگ حسگر بی‌سیم، شبکه‌هایی با قابلیت حرکت گره‌ها، شبکه‌های مبتنی بر اینترنت اشیا و ... نیز بهره برد.

نسخه OMNeT++ به‌کاررفته در این پژوهش، نسخه OMNeT++۶.۰.۱ می‌باشد. این نسخه در کنار ماژول کامپایلر C++، از زبان پایتون نیز برای محاسبه و تحلیل اطلاعات استفاده نماید. از جمله ویژگی‌های برجسته‌ی این نسخه از نرم‌افزار OMNeT++، می‌توان به موارد زیر اشاره نمود:

- سهولت شبیه‌سازی، در مقایسه با نسخه‌های گذشته OMNeT++
- توسعه‌ی استقلالِ بستر نرم‌افزاری شبیه‌ساز از نرم‌افزار پایه آن (Eclips)
- توسعه توابع از پیش تعریف‌شده در کتابخانه‌های نسخه‌ی رسمی کتابخانه OMNeT++
- توسعه ابزارهای رسم نمودارها در فایل‌های برداری، هیستوگرامی و...
- توسعه واسط گرافیکی مورداستفاده کاربر در مقایسه به نسخه‌های رسمی قبلی مانند نسخه ۵ (OMNeT++۵.۵.۴) [۱].

دیگر محیط همسان برای دستیابی به نتایج معتبر بسیار ضروری است. بستر سخت‌افزاری و نرم‌افزاری اجرای آزمایش‌ها را می‌توان به شرح زیر بیان نمود:

- پردازنده سیستم: corei۵ با فرکانس تا ۲/۸ مگاهرتز از نسل ۵
- حافظه: RAM ۸ گیگابایت
- نسخه نرم‌افزار شبیه‌ساز: OMNeT++ نسخه ۶.۰.۱
- سیستم‌عامل: Windows ۱۰

جدول ۳: ساختار داده‌های اولیه گره‌های شبکه.

نام متغیر	بازه متغیر	توضیحات
شناسه گره	۱ تا تعداد Max_{node}	نوع عدد integer برای هر گره یک مقدار منحصربه‌فرد
طول جغرافیایی	۱ تا ۱۰۰۰	نوع عدد long با واحد متر (انتخاب به‌صورت تصادفی)
عرض جغرافیایی	۱ تا ۱۰۰۰	نوع عدد long با واحد متر (انتخاب به‌صورت تصادفی)
برد گره	۲۰ تا ۳۰	نوع عدد long با واحد متر (انتخاب به‌صورت تصادفی)

٤-١-٢ متغیرهای شبیه‌سازی

در این پژوهش به‌رسم پژوهش‌های مطرح در حوزه اینترنت اشیا، متغیر به‌عنوان مفهومی تعریف می‌گردد که بیش از ۲ یا چند ارزش یا عدد را می‌توان به آن اختصاص داد. بنابراین تعریف می‌توان گفت تمامی متغیرهای مورد استفاده در این پژوهش متغیرهای عدد هستند. در این مقاله، ۳ متغیر وابسته برای بیان میزان کارایی روش‌های مورد بررسی تعریف شده است. این سه متغیر در بستر ۲ متغیر مستقل، سنجیده می‌شوند. رفتار سه متغیر وابسته در اثر تغییر متغیرهای مستقل نشان می‌دهد که روش پیشنهادی در مقایسه با روش‌های مورد بررسی تا چه اندازه کارا است.

در این پژوهش متغیرهای مستقل به ترتیب شامل:

- تعداد گره‌های شبکه
- تعداد داده ارسال شده از هر گره.

هستند. همچنین طبق بخش ۳، متغیرهای وابسته عبارت‌اند از:

- منابع حافظه‌ای: میزان حافظه مورداستفاده برای اجرای روش پیشنهادی در مجموع گره‌ها، بر حسب kB
- منابع پردازشی: میزان مشغول بودن پردازنده گره ها در اثر استفاده از روش پیشنهادی و سایر روش های موردبررسی. این میزان از زمان درخواست یک گره تا زمانی که گره همه پردازش‌های لازم را انجام داده و نتایج را نمایش می‌دهد، محاسبه می‌گردد. برحسب تعداد پردازش موردنیاز در ثانیه.
- منابع زمانی: میزان زمان لازم برای انجام تمامی کارهای محوله به مجموع گره‌ها تا زمانی که نتایج به هر کاربر یا بهره‌بردار شبکه، نمایش داده شود. واحد این متغیر ثانیه خواهد بود.

لازم به ذکر است، کارایی روش‌های موردبررسی، باتوجه‌به ضریب وزنی ۰/۳۳ برای سه متغیر وابسته ذکر شده در بالا و از (۳) محاسبه می‌گردد.

٤-١-٣ داده‌های مورد استفاده در شبیه‌سازی

قابل‌توجه است که در آزمایش روش‌های مورد بررسی دو نوع داده مورداستفاده قرار گرفته‌اند، این داده‌ها عبارت‌اند از:

- اطلاعات گره‌ها: گره‌ها خود به دو صورت تعریف شده‌اند
  - گره مرکز داده
  - گره سنجش یا گره محاسبه‌گر پارامترها

طبق آنچه که ذکر شد، یک گره دارای اطلاعاتی به فرم جدول ۳ است. این ساختار برای هر دو نوع گره تعریف شده یکسان است.

1. Objective Modular Network Testbed

جدول۴: جدول داده ارسال‌شده و شرایط شبیه‌سازی.

مهاجرت وظایف[1]	حالت عادی: غیر فعال میکروسرویس پایه: غیرفعال	انتقال وظایف بین گره‌ها در این دو حالت انجام نمی‌شود.
مدیری تتطبیقی منابع	حالت عادی ندارد میکروسرویس پایه: ندارد	در این دو بیس لاین تخصیص منابع بر اساس پیش‌بینی انجام نمی‌شود
نوع کنترل معماری	حالت عادی: محلی میکروسرویس پایه: متمرکزساده	نحوه‌ی کنترل و تصمیم‌گیری در تخصیص وظایف
	حالت عادی: تک‌بلوکی	
نوع پردازش داده	میکروسرویس پایه: چندبلوک مستقل	ساختار پردازش داده‌ها در هر گره

داده‌های دریافت شده از گره‌ها شامل ۱۰ ویژگی مربوط به اطلاعات مواد شیمیایی موجود در طبیعت، دریافت شده از ۱۰۰ نقطه در ۹ کشور اروپایی است. در یک ویژگی سطح آلودگی، و در آخرین ویژگی گره‌ای که باید اطلاعات را به مرکز داده‌ها ارسال کند، ثبت شده است. در واقع سناریوی طراحی شده با این هدف است که گره‌های شبکه همچون یک حسگر، اطلاعات مربوط به هر نقطه را به مرکز داده‌ها ارسال کنند و اقدام لازم را با توجه به داده‌های ارسال شده، از مرکز داده دریافت نمایند. بر این اساس ساختار داده‌های مورد استفاده طبق داده‌های آژانس محیط زیست اروپا[2] OMNeT++ در جدول ۴ نمایش داده شده است.

اگرچه آزمایش‌ها در محیط شبیه‌سازی OMNeT++ انجام شده‌اند، اما طراحی سبک و ماژولار روش پیشنهادی، امکان استقرار آن را در محیط‌های واقعی اینترنت اشیا فراهم می‌سازد. در پژوهش‌های آینده، پیاده‌سازی عملی روش در محیط‌های پویا شامل نویز داده، تحرک گره‌ها و تعاملات ناهمگن به‌عنوان گام بعدی در راستای اعتبارسنجی تجربی مدنظر قرار خواهد گرفت.

٤-۲ مقایسه نتایج حاصل

برای اثبات میزان کارایی روش پیشنهادی با دو حالت معماری میکروسرویس پایه و حالت عادی یا بدون استفاده از هیچ روشی و تنها بر اساس معماری پایه اینترنت اشیا، مقایسه شده است. این مقایسه‌ها همان‌طور که بیان شد در قالب متغیرهای ذکر شده در بخش قبل صورت پذیرفته اند. شرایط شبیه‌سازی برای هر ۳ روش، در شرایط یکسان بوده و میزان تغییرات، بر اساس تغییر تعداد گره‌ها و همچنین داده‌های ثبت شده از محیط (توسط گره های شبکه) محاسبه و ثبت شده‌اند.

٤-۲-۱ نتایج حاصل بر اساس مصرف حافظه

در جدول ۵ کارایی ۳ روش مورد بررسی، و حالت عادی، ازنظر میزان حافظه مصرفی، در تعداد ۲۵، ۵۰، ۷۵ و ۱۰۰ گره داده ثبت شده است. پس از افزایش یافتن تعداد گره‌ها، می‌توان گفت که روش پیشنهادی با افزایش میزان گره‌ها با شیب ثابتی به اندازه تعداد گره‌ها، میزان حافظه مصرفی‌اش رشد داشته است، در حالی که رشد میزان حافظه مصرفی در معماری میکروسرویس به صورت سهمی و در حالت عادی تقریباً غیرقابل تخمین است. همچنین روش پیشنهادی نسبت به الگوریتم TFS عملکرد بهتری از خود نشان داده است. لازم به ذکر است که این نتایج در تعداد ۵۰۰۰ داده ثبت شده‌اند.

1. Task Migration
2. https://www.eea.europa.eu/themes/air/air-emissions-data

جدول ۵: میزان افزایش مصرف حافظه با افزایش تعداد گره‌ها.

	۲۵	۵۰	۷۵	۱۰۰
روش پیشنهادی	۲۶	۵۲	۷۷	۱۰۴
معماری میکروسرویس	۱۲	۳۴	۱۰۹	۳۰۸
حالت عادی	۳۸	۸۰	۱۸۹	۴۰۹
TFS	۲۳	۵۵	۸۶	۱۲۲

جدول ۶: میزان حافظه مصرفی با افزایش تعداد داده ارسال‌شده.

	۱۰۰۰	۲۰۰۰	۳۰۰۰	۴۰۰۰	۵۰۰۰
روش پیشنهادی	۱۰۱	۱۰۲	۱۰۲	۱۰۳	۱۰۴
معماری	۴۴	۷۰	۱۳۶	۲۲۸	۳۰۸
حالت عادی	۸۸	۱۶۴	۲۶۰	۳۴۲	۴۰۹
TFS	۱۱۴	۱۱۶	۱۱۸	۱۲۰	۱۲۲

در جدول ۶ متغیر میزان مصرف حافظه تحت تأثیر افزایش تعداد داده ارسال شده از ۱۰۰۰ تا ۵۰۰۰ عدد داده و با تعداد ۱۰۰ گره محاسبه شده است. قابل ملاحظه است که افزایش تعداد داده در میزان حافظه مصرفی در روش پیشنهادی تأثیر چندانی نداشته است.

نمودارهای موجود در شکل‌های ۴ و ۵ نحوه تأثیر متغیرهای تعداد گره‌ها و تعداد داده‌های ارسال شده را بر روی میزان مصرف حافظه کل شبکه نشان می‌دهند.

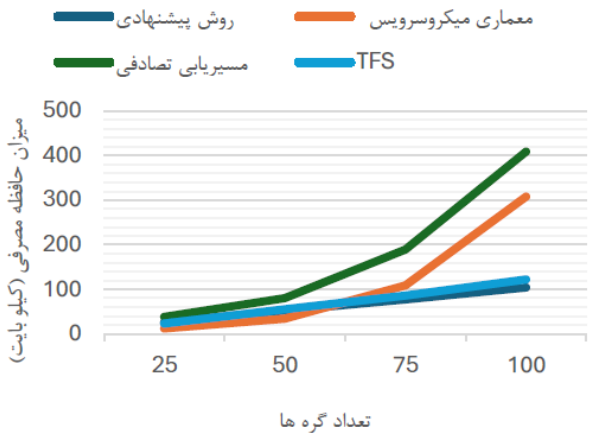
٤-۲-۲ نتایج حاصل بر اساس مصرف منابع پردازشی

در جدول ۷، عملکرد ۳ روش مورد بررسی در مقابل حالت عادی، ازنظر میزان مصرف منابع پردازشی مورد بررسی قرار گرفته‌اند، به مانند جدول ۵ این تغییرات در همان بازه تعداد گره محاسبه و ثبت شده است. در جدول ۷ مشخص است که پس از اضافه کردن گره‌ها می‌توان گفت که روش پیشنهادی با افزایش تعداد گره‌ها میزان سربار پردازشی کمتری نسبت به روش معماری میکروسرویس و روش TFS داشته است. پر واضح است که کمترین میزان سربار پردازشی زمانی اتفاق خواهد افتاد که هیچ کاری انجام نشود، مانند آنچه در حالت عادی رخ داده است. اعداد ثبت شده در جدول ۷ برای تعداد ۵۰۰۰ داده ارسالی ثبت شده است. لازم به ذکر است برای کاهش طول اعداد، این اعداد بر ۱۰۶ تقسیم شده و در جدول‌های ۵ و ۶ ثبت شده‌اند.

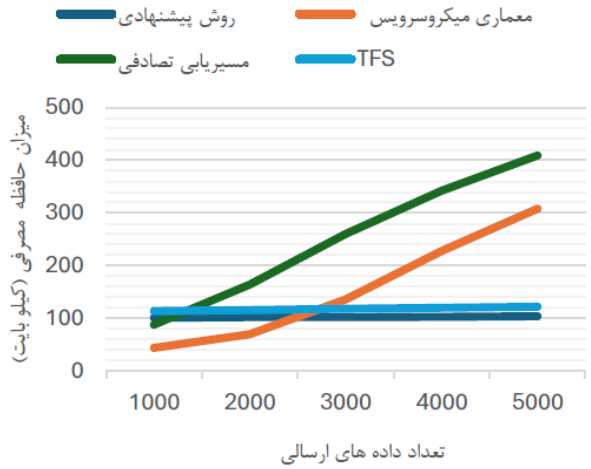
در جدول ۷ تأثیر افزایش تعداد داده‌های ارسال شده بر میزان متوسط منابع پردازشی مصرف شده با توجه به افزایش تعداد داده‌های ارسال شده از ۱۰۰۰ تا ۵۰۰۰ داده را در ۱۰۰ گره، نمایش می‌دهد.

نمودارهای موجود در شکل‌های ۶ و ۷ نحوه تأثیر متغیرهای تعداد گره ها و تعداد داده‌های ارسال شده بر روی میزان منابع پردازشی مصرف شده توسط کل شبکه را نشان می‌دهند.

نتایج جدول ۸ نشان می‌دهد که با تغییر نرخ ارسال داده از ۱۰۰۰ به ۵۰۰۰ داده (افزایش ناگهانی ترافیک)، روش پیشنهادی توانسته است، منابع پردازشی را بدون افزایش چشمگیر تأخیر بازتوزیع کند که نشان‌دهنده‌ی خاصیت تطبیق‌پذیری آن با نوسانات بار شبکه است.



شکل ۴: نحوه تغییرات مصرف حافظه‌ای در ۴ روش مورد بررسی با تغییر تعداد گره‌ها.



شکل ۵: میزان تغییرات مصرف حافظه‌ای در ۴ روش مورد بررسی با تغییر تعداد داده ارسال‌شده.

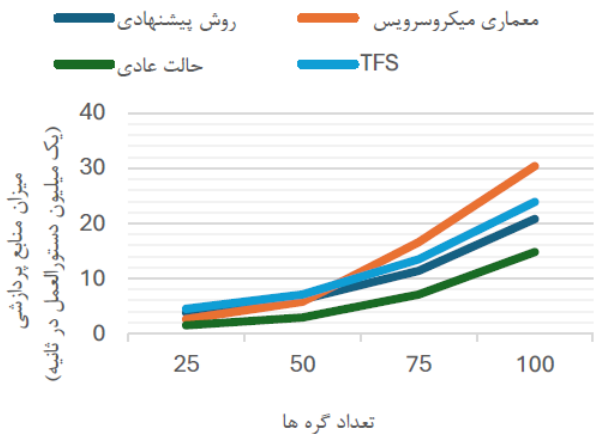
جدول ۷: میزان افزایش مصرف حافظه با افزایش تعداد گره‌ها.

	۲۵	۵۰	۷۵	۱۰۰
روش پیشنهادی	۳٫۸	۶٫۲	۱۱٫۴	۲۰٫۸
معماری میکروسرویس	۲٫۶	۵٫۸	۱۶٫۶	۳۰٫۴
حالت عادی	۱٫۵	۲٫۹	۷٫۱	۱۴٫۸
TFS	۴٫۵	۷٫۱	۱۳٫۵	۲۳٫۹

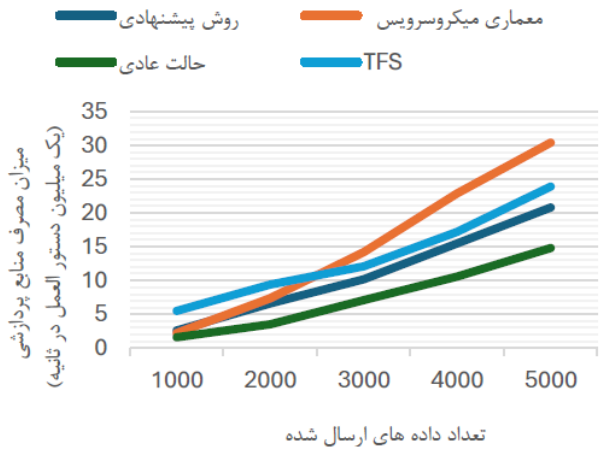
۴-۲-۳ نتایج حاصل بر اساس منابع زمانی

در جدول ۹، عملکرد سه الگوریتم مورد بررسی، و حالت عادی ازنظر منابع زمانی مورد بررسی قرار گرفته‌اند، به مانند جدول ۵ و جدول ۶ این تغییرات در بازه ۲۵ تا ۱۰۰ گره محاسبه و ثبت شده‌اند. از جدول ۹ می‌توان دریافت کرد که پس از اضافه کردن گره‌ها که در روش پیشنهادی با افزایش تعداد گره‌ها زمان کمتری نسبت به حالت عادی، روش معماری میکروسرویس و روش TFS مصرف می‌شود. لازم به ذکر است تعداد ۵۰۰۰ داده ارسالی برای اعداد به دست آمده در هر سلول جدول ۹، از کل گره‌ها به سمت مرکز داده ارسال شده است.

نمودارهای موجود در شکل‌های ۷ و ۸ نحوه تأثیر متغیرهای تعداد گره‌ها و تعداد داده ارسالرشده بر روی میزان منابع زمانی مصرف‌شده



شکل ۶: میزان مصرف منابع پردازشی با افزایش تعداد گره‌ها.



شکل ۷: میزان مصرف منابع پردازشی با افزایش تعداد داده ارسال‌شده.

جدول ۸: میزان مصرف منابع پردازشی با افزایش تعداد داده ارسال‌شده.

	۱۰۰۰	۲۰۰۰	۳۰۰۰	۴۰۰۰	۵۰۰۰
روش پیشنهادی	۲٫۶	۶٫۶	۱۰٫۲	۱۵٫۵	۲۰٫۸
معماری میکروسرویس	۲٫۲	۷٫۴	۱۴٫۲	۲۲٫۹	۳۰٫۴
حالت عادی	۱٫۶	۳٫۵	۷٫۱	۱۰٫۶	۱۴٫۸
TFS	۵٫۵	۹٫۴	۱۲٫۱	۱۷٫۲	۲۳٫۹

توسط کل شبکه را نشان می‌دهند. همان‌گونه که در نتایج مشاهده می‌شود، طراحی دولایه صف‌ها و استفاده از سیاست زمان‌بندی تطبیقی مبتنی بر ESN باعث کاهش میانگین زمان پاسخ در سناریوهای پرترافیک شده است.

در نهایت با توجه به هرکدام از متغیرهای ذکر شده می‌توان گفت که میزان افزایش یا کاهش منابع مصرفی ذکر شده، در روش پیشنهادی نسبت به معماری میکروسرویس و حالت عادی در نمودار شکل ۱۰ به نمایش درآورده شده است.

روش پیشنهادی مبتنی بر ترکیب معماری میکروسرویس و شبکه‌های ESN، با ایجاد ساختاری ماژولار و سبک، به‌صورت چشمگیری بهره‌وری منابع زمانی را در شبکه‌های Fog-DSDN افزایش می‌دهد. استفاده از معماری میکروسرویس موجب می‌شود که وظایف به اجزای کوچک‌تر و مستقل تقسیم شوند و هر سرویس به طور جداگانه مدیریت، استقرار و

جدول ۹: منابع زمانی با افزایش تعداد گره‌ها.

	۲۵	۵۰	۷۵	۱۰۰
روش پیشنهادی	۱۷	۱۸	۲۰	۲۱
معماری میکروسرویس	۱۴	۱۸	۲۳	۲۷
حالت عادی	۱۹	۲۴	۳۰	۳۵
TFS	۱۷	۱۸	۲۰	۲۱

جدول ۱۰: منابع زمانی با افزایش تعداد داده ارسال‌شده.

	۱۰۰۰	۲۰۰۰	۳۰۰۰	۴۰۰۰	۵۰۰۰
روش پیشنهادی	۴	۷	۱۲	۱۷	۲۱
معماری میکروسرویس	۴	۹	۱۵	۲۱	۲۷
حالت عادی	۷	۱۳	۲۱	۲۷	۳۵
TFS	۵	۹	۱۵	۲۰	۲۴

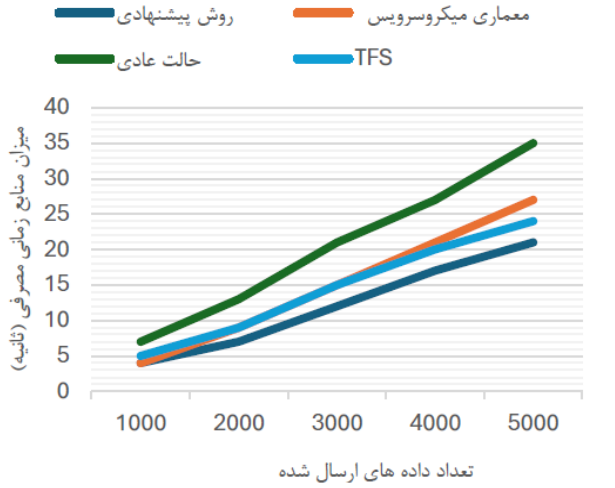
مقیاس‌پذیر شود. این ساختار به کاهش زمان پاسخگویی سیستم و تسریع در تصمیم‌گیری‌های مرتبط با تخصیص منابع کمک می‌کند. همچنین، قابلیت اجرای موازی میکروسرویس‌ها منجر به پردازش همزمان درخواست‌ها و کاهش چشمگیر زمان اجرای وظایف نسبت به روش‌های یکپارچه و متمرکز می‌شود.

از منظر مصرف منابع حافظه‌ای ترکیب ESN با معماری میکروسرویس باعث کاهش چشمگیر در حافظه مصرفی می‌شود، زیرا شبکه‌های ESN برخلاف شبکه‌های عصبی بازگشتی سنتی، تنها بخش مخزن را تصادفی مقداردهی می‌کنند و وزن‌های داخلی آن در طول آموزش تغییر نمی‌کنند. این ویژگی موجب ساده‌سازی فرایند آموزش و حذف نیاز به ذخیره‌سازی پی‌درپی وزن‌ها و گرادیان‌ها می‌شود. همچنین، پیاده‌سازی ماژولار میکروسرویس‌ها به سیستم اجازه می‌دهد فقط اجزای موردنیاز در حافظه بارگذاری شوند که این امر مصرف حافظه را بهینه کرده و از اشغال غیرضروری منابع توسط سرویس‌های غیرفعال جلوگیری می‌کند.

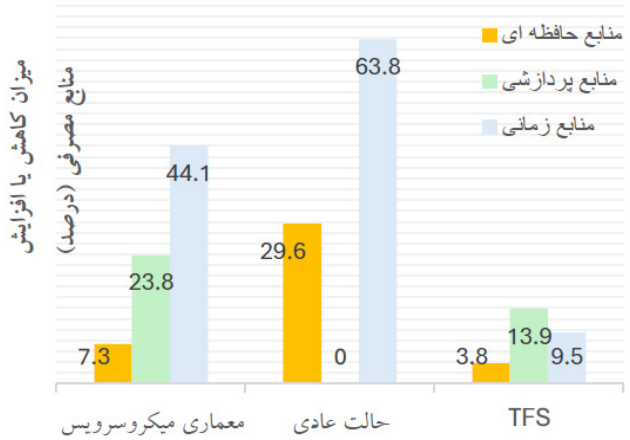
در زمینه منابع پردازشی، الگوریتم پیشنهادی به دلیل استفاده از ESN که دارای پیچیدگی محاسباتی بسیار پایین‌تری نسبت به سایر شبکه‌های یادگیری عمیق مانند LSTM و GRU است، بار پردازشی کمتری به سیستم تحمیل می‌کند. این ویژگی، به‌ویژه در گره‌های مه با توان پردازشی محدود، اهمیت بالایی دارد. از سوی دیگر، قابلیت سبک بودن میکروسرویس‌ها موجب می‌شود عملیات پردازشی بین گره‌ها توزیع شود و از تمرکز بار روی یک یا چند نقطه جلوگیری گردد. این توزیع بار به بهبود توازن منابع پردازشی، افزایش پایداری و کاهش احتمال گلوگاه در زیرساخت‌های Fog-DSDN منجر می‌شود.

نتایج به‌دست‌آمده بیانگر آن است که نوآوری‌های به‌کاررفته در روش پیشنهادی توانسته‌اند به‌صورت مؤثر موجب افزایش بهره‌وری منابع در محیط‌های واقعی Fog-DSDN شوند. بهبود ۱۲٫۵۷٪ در بهره‌وری منابع پردازشی نشان می‌دهد که سیستم با پیش‌بینی دقیق بار گره‌ها، از تخصیص اضافی و بلااستفاده منابع جلوگیری کرده است. افزایش ۱۸٫۸۵٪ در استفاده بهینه از حافظه نشان‌دهنده‌ی توان مدل در تنظیم پویا و هم‌زمانی حجم داده‌های پردازشی میان‌گره‌ها است که منجر به

شکل ۸: منابع زمانی با افزایش تعداد گره‌ها.



شکل ۹: منابع زمانی با افزایش تعداد داده ارسال‌شده.



شکل ۱۰: مقایسه کاهش یا افزایش منابع مصرفیِ روش پیشنهادی با روش معماری میکروسرویس و حالت عادی.

کاهش تراکم صف‌های ورودی و خروجی شده است. همچنین، کاهش ۳۹٫۱۳٪ در تأخیر سرویس، تأیید می‌کند که تصمیم‌گیری توزیع‌شده و برخط در سطح میکروسرویس‌ها توانسته است زمان پاسخ سامانه را در برابر تغییرات ناگهانی بار بهبود دهد. به‌صورت عملی، این نتایج نشان می‌دهد روش پیشنهادی می‌تواند در پیاده‌سازی‌های صنعتی یا خدماتی

واقعی، موجب پایداری بیشتر شبکه، تخصیص هوشمندتر منابع، و ارائه سرویس باکیفیت بالاتر نسبت به ساختارهای رایج Fog–DSDN گردد.

۵- پیشنهادها

فناوری اینترنت اشیا در سالیان اخیر توانسته است رشد زیادی را تجربه نموده و بر روی حوزه‌های مختلف، از بهداشتی و درمانی گرفته تا کاربردهای نظامی و امنیتی، تأثیر فراوانی بگذارد. علی‌رغم تمامی این پیشرفت‌ها اینترنت اشیا همچنان با چالش‌های جدّی و مهمی مواجه است، در بخش‌های اخیر تلاش شد که این چالش‌ها بیان و برای چالش مصرف منابع مصرفی در اینترنت اشیا یک راه‌حل بهینه ارائه شود.

۵-۲ بررسی چالش‌های جاری

استفاده از روش پیشنهادی به‌عنوان یک روش مدیریت منابع، با چالش های فراوانی همراه است، همان‌طور که در بخش گذشته بیان شد، این چالش‌ها می‌توانند مزایای استفاده از میکروسرویس‌ها را به شیوه پیشنهادی، تحت‌تأثیر خود قرار دهد. اما چالش‌هایی وجود دارد که باوجود بهینه‌کردن متغیرهای موردبررسی، راه‌حلی برای آن‌ها یافت نشد. برخی از این چالش‌ها عبارتند از:

- عدم دسترسی به محیط واقعی برای پیاده‌سازی روش پیشنهادی
- محدودیت‌های سخت‌افزاری / نرم‌افزاری که دست نویسنده را در سنجش دامنهٔ وسیع‌تر متغیرها می‌بندد.
- اثر منفی تعداد داده‌های ارسالی بر متغیرهای موردبررسی در روش پیشنهادی
- عملکرد ضعیف روش پیشنهادی در تعداد داده ارسالی کم نسبت به روش معماری میکروسرویس
- عدم بررسی برخی دیگر از متغیرهای تأثیرگذار در کارایی یک روش مدیریت منابع، مثل مقیاس‌پذیری، مصرف انرژی و...
- درنظرنگرفتن همه چالش‌های موجود بر سر راه یک روش مدیریت منابع، مثل چالش امنیت، چالش حفظ حریم خصوصی و...
- تمرکز بر سه متغیر حافظه‌ای، پردازشی و زمانی است و متغیرهای دیگری مانند مصرف انرژی، مقیاس‌پذیری و امنیت موردبررسی قرار نگرفته‌اند. این عوامل می‌توانند در سناریوهای واقعی تأثیر قابل‌توجهی بر عملکرد سیستم داشته باشند.

۵-۲ پیشنهادهایی برای پژوهش

موارد بیان شده در بخش ۵-۱ را می‌توان به صورت پیشنهادهایی برای پژوهش در اختیار سایر پژوهشگران، دانشجویان و علاقه‌مندان شیوه‌های مدیریت منابع در شبکه اینترنت اشیا قرار داد؛ لذا در این بخش فهرستی به صورت زیر تدوین شده است:

- شبیه‌سازی ساختار موردبررسی در یک محیط واقعی و ثبت و بررسی چالش‌های آن
- محاسبه هر یک از متغیرهای موردبررسی با دامنه‌های عددی وسیع تر و ثبت و سنجش رفتار هر یک از متغیرهای موردبررسی
- ارائه راه‌حلی برای چالشِ اثر منفی تعداد داده‌های ارسالی بر متغیرهای موردبررسی
- ارائه راه‌حلی برای بهبود عملکرد روش پیشنهادی در تعداد داده ارسالیِ پایین
- ارائه راه‌حل‌هایی برای سایر چالش‌های بررسی نشده در سیستم مدیریت منابع شبکه اینترنت اشیا
- ایجاد مدل کارایی جامع‌تر با بررسی سایر متغیرهای تأثیرگذار در یک روش بهبود منابع مصرفی، در شبکه اینترنت اشیا
- ارزیابی تأثیر مصرف انرژی، مقیاس‌پذیری و امنیت شبکه بر کارایی روش پیشنهادی در محیط‌های واقعی
- توسعه نسخه مقاوم در برابر نویز داده و تحرک گره‌ها برای کاربرد در سامانه‌های پویا
- ارزیابی روش در سناریوهای با تحرک بالا (با استفاده از SUMO + OMNeT++/Veins) و بررسی معیارهایی مانند تاخیر سرویس، هزینه مهاجرت وظایف و پایداری نقشه اطلاعاتی
- تحلیل کمی تأثیر معماری میکروسرویس بر شاخص‌های امنیت و حریم خصوصی داده‌ها از طریق شبیه‌سازی حملات احتمالی و سنجش میزان نشت داده

مراجع

[1] L. G. F. da Silva, D. F. H. Sadok, and P. T. Endo, "Resource optimizing federated learning for use with IoT: A systematic review," *Journal of Parallel and Distributed Computing*, vol. 175, pp. 92-108, May 2023.

[2] C. Li, *et al.*, "Deep reinforcement learning based controller placement and optimal edge selection in SDN-based multi-access edge computing environments," *Journal of Parallel and Distributed Computing*, vol. 193, Article ID: 104948, Nov. 2024.

[3] X. Zhou, J. Yang, Y. Li, S. Li, and Z. Su, "Deep reinforcement learning-based resource scheduling for energy optimization and load balancing in SDN-driven edge computing," *Computer Communications*, vol. 226-227, Article ID: 107925, Oct. 2024.

[4] M. Mahmoudi, B. Barekatain, Z. Beheshti, A. A. Quintana, and M. R. Velayati, "A new method for load balancing in DSDN-Based data centers using adaptive clustering and normal cone-based estimation approaches," *Expert Systems with Applic*ations, Article ID: 127606, Jun. 2025.

[5] F. Briatore and M. Braggio, "Edge, Fog and Cloud Computing framework for flexible production," *Procedia Computer Science*, vol. 253, pp. 2206-2218, 2025.

[6] Y. Jin, "An effective method for prospective scheduling of tasks in cloud-fog computing with an energy consumption management approach based on Q-learning," *Engineering Applications of Artificial Intelligence*, vol. 151, Article ID: 110705, Jul. 2025.

[7] A. M. Rahmani, *et al.*, "Optimizing task offloading with metaheuristic algorithms across cloud, fog, and edge computing networks: A comprehensive survey and state-of-the-art schemes," *Sustainable Computing: Informatics and Systems*, vol. 45, Article ID: 101080, Jan. 2025.

[8] E. Matinfar, M. Mahmoudi, and B. Barekatain, "ELDM-EDSDN: A novel resource allocation method in edge-DSDN-based networks using improved manifold approach," *The Journal of Supercomputing*, vol. 81, no. 14, Article ID: 1341, 2025.

[9] M. Kabeer, I. Yusuf, and N. A. Sufi, "Distributed software defined network-based fog to fog collaboration scheme," *Parallel Computing*, vol. 117, Article ID: 103040, Sept. 2023.

[10] I. Ahammad, M. A. R. Khan, and Z. U. Salehin, "QoS performance enhancement policy through combining fog and SDN," *Simulation Modelling Practice and Theory*, vol. 109, Article ID: 102292, May 2021.

[11] S. O. Ogundoyin and I. A. Kamil, "Optimization techniques and applications in fog computing: An exhaustive survey," *Swarm and Evolutionary Computation*, vol. 66, Article ID: 100937, Oct. 2021.

[12] J. Al Muhtadi, R. A. Alamri, F. A. Khan, and K. Saleem, "Subjective logic-based trust model for fog computing," *Computer Communications*, vol. 178, pp. 221-233, Oct. 2021.

[13] A. J. Kadhim, S. A. H. Seno, J. I. Naser, and J. Hajipour, "DMPFS: Delay-efficient multicasting based on parked vehicles, fog computing and SDN in vehicular networks," Vehicular Communications, vol. 36, Article ID: 100488, Aug. 2022.

[14] K. A. Darabkh, B. Z. Alkhader, A. F. Khalifeh, F. Jubair, and M. Abdel-Majeed, "ICDRP-F-SDVN: An innovative cluster-based dual-phase routing protocol using fog computing and software-defined

vehicular network," *Vehicular Communications*, vol. 34, Article ID: 100453, Apr. 2022.
[15] E. Hosseini, M. Nickray, and S. Ghanbari, "Optimized task scheduling for cost-latency trade-off in mobile fog computing using fuzzy analytical hierarchy process," *Computer Networks*, vol. 206, Article ID: 108752, Apr. 2022.
[16] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," *Journal of Network and Computer Applications*, vol. 201, Article ID: 103333, May 2022.
[17] B. Sarma, R. Kumar, and T. Tuithung, "Machine learning enabled network and task management in SDN based Fog architecture," *Computers and Electrical Engineering*, vol. 108, Article ID: 108705, May 2023.
[18] M. M. Islam, F. Ramezani, H. Y. Lu, and M. Naderpour, "Optimal placement of applications in the fog environment: A systematic literature review," *Journal of Parallel and Distributed Computing*, vol. 174, pp. 46-69, Apr. 2023.
[19] Z. H. Ali, N. El-Rashidy, M. A. Elhosseini, and S. M. Ayyad, "SDN-based reliable emergency message routing schema using Digital Twins for adjusting beacon transmission in VANET," Journal of Network and Computer Applications, vol. 230, Article ID: 103944, Oct. 2024.
[20] E. Matinfar, *et al.*, "FABLSTM: Aan optimal resource allocation in SDVN networks with contextual variables integration," *The Journal of Supercomputing*, vol. 81, no. 15, Article ID: 1454, Oct. 2025.
[21] D. Adanza, *et al.*, "Enabling traffic forecasting with cloud-native SDN controller in transport networks," *Computer Networks*, vol. 250, Article ID: 110565, Aug. 2024.

عباس پیامنی در سال ۱۳۹۰ مدرک کارشناسی گرایش نرم‌افزار خود را از دانشگاه صفاهان اصفهان دریافت کرد و در سال ۱۳۹۴ مدرک کارشناسی‌ارشد گرایش هوش مصنوعی را از دانشگاه شهاب دانش قم دریافت نمود. از سال ۱۴۰۰ به بعد در دانشگاه ملی مهارت مشغول به تدریس می‌باشد. حوزه فعالیت ایشان پردازش تصویر، بینایی ماشین و یادگیری ماشین می‌باشد.

احسان متین‌فر در سال ۱۳۸۸ مدرک کارشناسی کامپیوتر خود را از دانشگاه بوعلی همدان و در سال ۱۳۹۵ مدرک کارشناسی‌ارشد مهندسی کامپیوتر خود را از دانشگاه آزاد واحد دورود دریافت نمود. نام‌برده هم‌اکنون به عنوان هنرآموز کامپیوتر در وزارت آموزش و پرورش مشغول به کار است. همچنین ایشان به عنوان پژوهشگر مستقل در پژوهش‌های آکادمیک و صنعتی همکاری می‌کند. زمینه‌های پژوهشی مورد علاقه ایشان عبارتند از: شبکه‌های نرم‌افزار محور، شبکه‌های خودرویی، بهینه‌سازی استفاده از منابع، شبکه‌های اینترنت اشیا و طراحی الگوریتم‌های مبتنی بر ریاضیات مدرن.

الهام مقامی در سال ۱۳۹۳ مدرک کارشناسی کامپیوتر خود را از دانشگاه پیام‌نور واحد دورود و در سال ۱۴۰۲ مدرک کارشناسی‌ارشد مهندسی کامپیوتر گرایش هوش مصنوعی را از دانشگاه صنعتی اصفهان دریافت نمود. نام‌برده هم‌اکنون به عنوان پژوهشگر مستقل در پروژه‌های صنعتی و آکادمیک همکاری دارد. زمینه‌های پژوهشی مورد علاقه ایشان عبارتند از: شبکه‌های نرم‌افزار محور، شبکه‌های خودرویی و بهینه‌سازی استفاده از منابع.

محمد جانبزرگی در سال ۱۳۹۶ مدرک کارشناسی‌ارشد مهندسی نرم‌افزار کامپیوتر را از دانشگاه آزاد اسلامی واحد دورود دریافت نمود و از سال ۱۳۸۹ الی ۱۴۰۱ به مدت ۱۲ سال مسئول فن آوری اطلاعات شبکه بهداشت مشغول فعالیت بوده و از سال ۱۳۹۹ الی ۱۴۰۵ به عنوان مدرس دانشکده سلامت؛ دانشکده فنی، دانشگاه علمی‌کاربردی و هنرستان‌های شهرستان دورود مشغول به تدریس می‌باشد. زمینه‌های علمی مورد علاقه نام‌برده متنوع بوده و شامل موضوعاتی در زمینه شبکه‌های کامپیوتری، اینترنت اشیا، هوش مصنوعی و داده‌کاوی می‌باشد.