دوفصلنامه محاسبات و سامانه های توزیع شده
سال هفتم، شماره ۱، شماره پیاپی ۱۳، صفحه ۳۳ تا ۵۰، سال ۱۴۰۳

بهبود انتقال وظایف با استفاده از منطق فازی نوع ۲ و الگوریتم یادگیری تقویتی با سیاست مجاور برای افزایش عملکرد برنامه های اینترنت اشیاء

مصطفی عابدی[۱]، سید ابراهیم دشتی[۲]

[۱]دانشجوی کارشناسی ارشد ، دانشکده برق و کامپیوتر، واحد شیراز، دانشگاه آزاد اسلامی

abediiii.mostafaaaa@gmail.com

[۲]دانشکده برق و کامپیوتر، واحد جهرم، دانشگاه آزاد اسلامی (نویسنده مسئول)

Sayed.Dashty@gmail.com

چکیده

این مقاله روش نوین مهاجرت وظایف ارائه می‌دهد که برای بهبود کارایی برنامه های اینترنت اشیا در یک معماری سه لایه در محیط محاسبات مه و ابر طراحی شده است. لایه اینترنت اشیا شامل دستگاه‌های هوشمندی است که تعداد زیادی وظایف تولید می‌کنند؛ وظایفی که هر یک دارای مشخصات متنوعی مانند اندازه، نیازهای محاسباتی، نیاز های ارتباطی و محدودیت های زمانی هستند. به دلیل محدودیت ذخیره‌سازی و ظرفیت محاسباتی دستگاه‌های اینترنت اشیا، لازم است این وظایف به لایه‌های مختلف منتقل شوند تا پردازش مؤثری انجام شود و اهداف کیفیت خدمات (QoS) برآورده شوند. برای حل این چالش، از یک برنامه‌ریز وظایف مبتنی بر منطق فازی نوع ۲ استفاده شده است تا تصمیم‌گیری‌های هوشمندانه‌ای برای مهاجرت وظایف انجام دهد. این برنامه‌ریز بر اساس ویژگی‌های وظایف، مناسب‌ترین لایه پردازشی را انتخاب می‌کند. علاوه بر این، در این مقاله از یادگیری عمیق تقویتی بهینه‌سازی سیاست مجاور (PPO) استفاده شده است تا با مهاجرت مناسب وظایف سطح مه، تعادل بار میان گره‌های همکار حفظ گردد. نتایج تجربی نشان می‌دهند که طرح پیشنهادی در مقایسه با روش‌های پیشرفته موجود، از نظر کاهش تأخیر، مصرف انرژی، استفاده از شبکه، توان عملیاتی و نرخ مهاجرت وظایف عملکرد بهتری دارد.

کلمات کلیدی : انتقال وظایف، منطق فازی نوع دو، برنامه های اینترنت اشیا ، رایانش مه ، یادگیری تقویتی عمیق، بهینه سازی سیاست مجاور (PPO).

تاریخچه‌ی مقاله:
تاریخ ارسال: ۱۴۰۳/۰۲/۰۲
تاریخ اصلاحات: ۱۴۰۳/۰۵/۲۷
تاریخ پذیرش: ۱۴۰۳/۰۶/۲۱
تاریخ انتشار: ۱۴۰۳/۰۶/۳۰
ایمیل نویسنده مسئول: sayed.dashty@gmail.com

۱-مقدمه

گسترش فناوری اینترنت اشیا (IoT) به‌عنوان یکی از مؤلفه‌های اصلی تحول دیجیتال، به ظهور برنامه‌های کاربردی نوینی منجر شده است که نیازمند پردازش سریع، دقیق و کم‌هزینه داده‌ها هستند. برنامه‌هایی مانند وسایل نقلیه خودران [۱]، واقعیت افزوده، و سامانه‌های پایش سلامت هوشمند [۲], [۳]، از جمله نمونه‌هایی هستند که نه‌تنها حساس به زمان‌اند، بلکه حجم عظیمی از داده‌ها را برای پردازش تولید می‌کنند. این نیازها، دستگاه‌های IoT را که معمولاً از منابع محاسباتی و انرژی محدودی برخوردارند، با چالش‌های جدی مواجه می‌سازد. برای پاسخگویی به این چالش‌ها، فناوری رایانش مه به‌عنوان مکملی برای رایانش ابری معرفی شده است. رایانش مه با انتقال خدمات محاسباتی به لبه شبکه، امکان پردازش داده‌ها را نزدیک به منبع تولید آن‌ها فراهم می‌کند [۴]. این قابلیت نه‌تنها تأخیر ناشی از انتقال داده‌ها به ابر را کاهش می‌دهد، بلکه مصرف انرژی و بار شبکه را نیز بهبود می‌بخشد. با این حال، بهره‌گیری از این

فناوری مستلزم حل مسائل پیچیده‌ای مانند انتقال وظایف و توزیع بار کاری در میان گره‌های مه است[۵].

انتقال وظایف به‌عنوان یکی از مهم‌ترین چالش‌ها، مستلزم تصمیم‌گیری هوشمندانه درباره محل پردازش وظایف (محلی، گره‌های مه یا ابر) و همچنین توزیع بهینه بار کاری میان گره‌های ناهمگون شبکه مه است. با افزایش تعداد درخواست‌ها، خطر افزایش زمان انتظار و کاهش عملکرد سیستم وجود دارد. در همین راستا، استراتژی‌های مختلفی مانند انتقال وظایف عمودی به ابر یا انتقال وظایف افقی میان گره‌های مه پیشنهاد شده‌اند، اما این روش‌ها هنوز با محدودیت‌هایی مواجه‌اند. برای مقابله با این چالش‌ها، این مقاله استفاده از منطق فازی نوع ۲ و الگوریتم یادگیری تقویتی با سیاست مجاور (PPO) را به‌عنوان یک استراتژی ترکیبی معرفی می‌کند. منطق فازی نوع ۲ با توانایی مدیریت عدم قطعیت در تصمیم‌گیری، برای انتخاب بهینه لایه پردازشی (محلی، مه یا ابر) به‌کار گرفته می‌شود [۶]. در همین حال، الگوریتم PPO به‌عنوان یکی از تکنیک‌های پیشرفته یادگیری تقویتی، فرآیند تخصیص وظایف به گره‌های مه و توزیع متوازن بار کاری در میان گره‌های یک خوشه را بهینه می‌کند [۷]. هدف اصلی این پژوهش ارائه یک استراتژی نوآورانه برای بهبود عملکرد برنامه‌های IoT از طریق انتقال بهینه وظایف در معماری سه‌لایه مه-ابر است. مزایای کلیدی این استراتژی عبارت‌اند از:

۱. کاهش تأخیر: پردازش وظایف حساس به زمان در نزدیکی منبع داده.
۲. بهینه‌سازی مصرف انرژی: کاهش بار ارسالی به ابر و استفاده بهینه از منابع مه.
۳. توزیع بار کاری متوازن: جلوگیری از بارگذاری بیش‌ازحد گره‌های قدرتمند و استفاده از ظرفیت تمامی گره‌های مه.

برای ارزیابی کارایی این روش، شبیه‌سازی‌هایی با استفاده از ابزار iFogSim انجام شده است که نشان می‌دهد استراتژی پیشنهادی در مقایسه با روش‌های موجود، تأخیر، مصرف انرژی، استفاده از شبکه، توان عملیاتی و نرخ موفقیت انتقال وظایف را بهبود می‌بخشد [۸].

ساختار مقاله به شرح زیر است: بخش دوم به بررسی مطالعات مرتبط و روش‌های موجود در انتقال وظایف می‌پردازد. در بخش سوم، استراتژی پیشنهادی به تفصیل تشریح می‌شود. الگوریتم‌های مورد استفاده در بخش چهارم توضیح داده می‌شوند. بخش پنجم به ارزیابی عملکرد روش پیشنهادی و مقایسه آن با روش‌های مبنا اختصاص دارد. در نهایت، بخش ششم به نتیجه‌گیری و معرفی مسیرهای تحقیقاتی آینده می‌پردازد.

۲ - کارهای مرتبط

انتقال وظایف یک مفهوم تثبیت شده است که به طور گسترده در رایانش ابری استفاده می‌شود. تخصیص وظایف شامل انتقال وظایف از دستگاه‌های IoT که دارای منابع محدود هستند به دستگاه‌های غنی از منابع، با هدف بهبود عملکرد برنامه‌های حساس به زمان است. دستگاه‌های کاربر در لبه شبکه قرار می‌گیرند تا وظایف محاسباتی سنگین به گره‌های مه و ابر از طریق شبکه‌های 4G/5G یا شبکه‌های محلی بی‌سیم (WLAN) منتقل شوند. در مواردی که یک گره مه قادر به مدیریت حجم کاری فزاینده نباشد، گره‌های اضافی مه یا ابر برای کمک در نظر گرفته می‌شوند. این راه‌حل عملی، از طریق ارسال وظایف محاسباتی به سرورهای قدرتمند در سیستم مه-ابر، به پشتیبانی برنامه‌های IoT کمک می‌کند. محققان [۱۰] به‌طور سیستماتیک چالش‌های مربوط به انتقال وظایف را با استفاده از الگوریتم‌های یادگیری تقویتی (RL) و یادگیری عمیق تقویتی (DRL) تحلیل کرده‌اند. این مطالعه به بررسی فرآیند انتقال وظایف از جنبه‌های مختلف مانند تصمیم‌گیری انتقال وظایف، معیارها، جهت‌ها و حالت‌های انتقال وظایف می‌پردازد.

این بخش به بررسی مطالعات مرتبط و روش‌هایی می‌پردازد که با تحقیق ما ارتباط نزدیکی دارند. به عنوان مثال، مطالعه [۹] یک چارچوب بهینه‌سازی انتقال وظایف میان لایه‌های محاسباتی مه، ابری برای مدیریت داده‌های حجیم تولیدشده در اینترنت اشیا (IoT) ارائه می‌دهد. هدف اصلی این چارچوب کاهش مصرف انرژی و تأخیر سرویس در محیط‌های پویا و تأخیر حساس است. این تحقیق با استفاده از مدل‌سازی ریاضی و تکنیک‌های برنامه‌ریزی خطی عدد صحیح، روشی را پیشنهاد

می‌کند که توزیع بهینه وظایف را بین دستگاه‌های مه و ابر فراهم می‌کند. نتایج نشان می‌دهد که این رویکرد در مقایسه با روش‌های موجود، تأخیر و مصرف انرژی کمتری دارد و عملکرد کلی سیستم را بهبود می‌بخشد.

مطالعه‌ای دیگر به منظور اولویت‌بندی وظایف و سیاست‌های انتقال وظایف با استفاده از منطق فازی [۱۱] انجام شد. هدف اصلی این روش، کاهش تأخیر و زمان انتظار متوسط بود. همچنین، اولویت وظایف در نظر گرفته شد و وظایف به صف‌های مناسب برای زمان‌بندی تخصیص داده شدند. این مطالعه از یک الگوریتم جایا چندجمعیتی با نخبگی باینری برای زمان‌بندی وظایف استفاده کرد. اگرچه نتایج نشان‌دهنده بهبود عملکرد این روش بود، اما این طراحی نتوانست گره بهینه مه را انتخاب کند و بار کاری را به‌طور یکنواخت توزیع کند.

مطالعه ی [۱۶]، به بررسی مسئله انتقال وظایف اینترنت اشیا (IoT) به لایه مه می‌پردازد تا زمان پاسخ‌دهی برنامه‌های IoTکاهش یابد. نویسندگان دو الگوریتم بهینه‌سازی الهام‌گرفته از طبیعت، شامل بهینه‌سازی کلونی مورچه‌ها (ACO)و بهینه‌سازی ازدحام ذرات(PSO) ، را برای توزیع وظایف بر روی گره‌های مه معرفی می‌کنند. این الگوریتم‌ها هزینه ارتباطات، زمان محاسبه، و بار موجود روی گره‌های مه را در نظر می‌گیرند. نتایج تجربی نشان می‌دهد که الگوریتم ACOدر کاهش زمان پاسخ‌دهی و توازن بار روی گره‌های مه عملکرد بهتری نسبت به PSO و الگوریتم Round Robin دارد. این تحقیق با هدف بهبود کیفیت سرویس در محیط‌های تأخیر حساس مانند شهرهای هوشمند ارائه شده است.

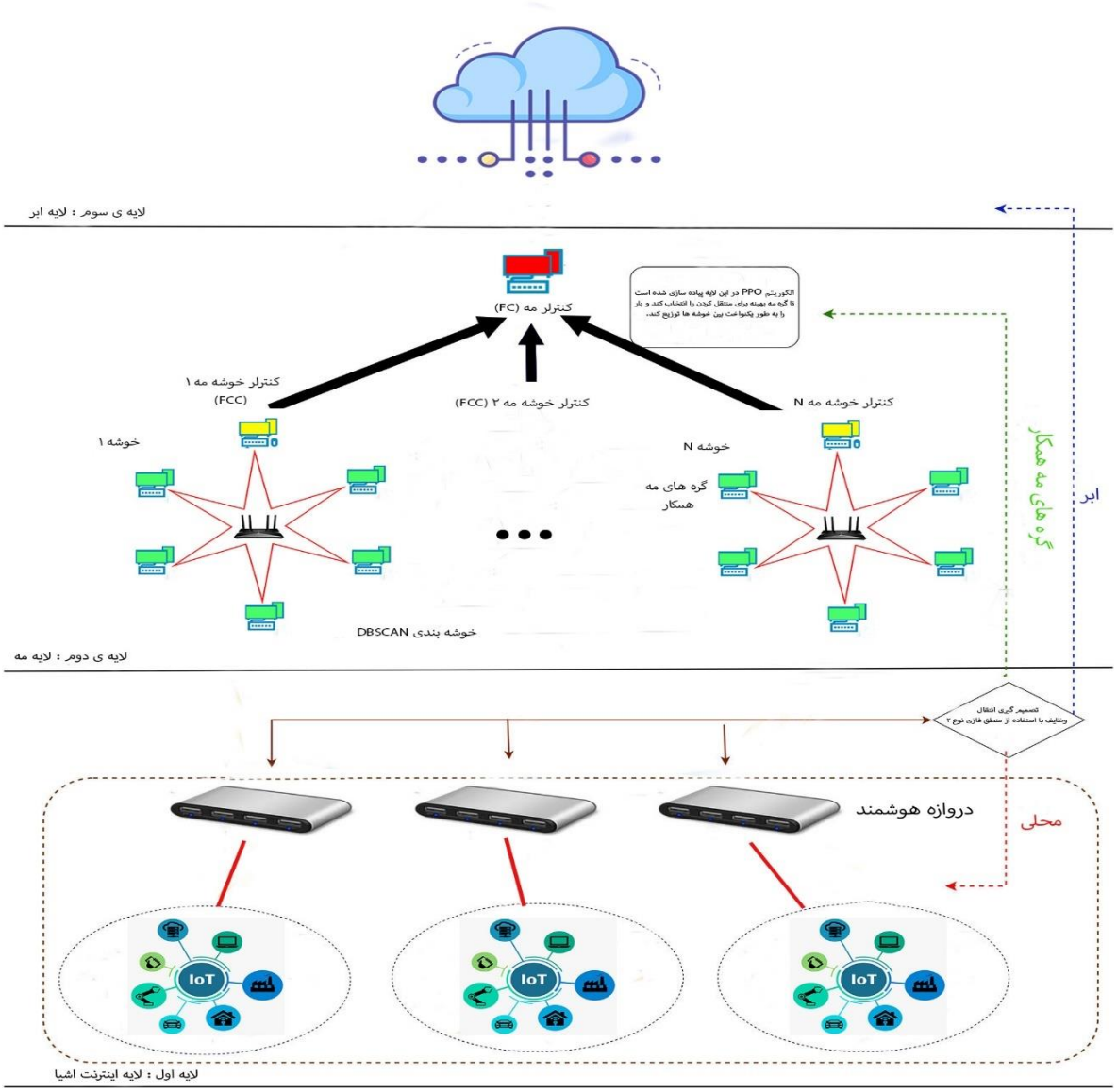
علاوه بر این، نویسندگان [۱۲] یک استراتژی نوآورانه برای انتقال وظایف ارائه کردند که هدف آن تصمیم‌گیری بهینه درباره زمان و مکان انتقال وظایف (گره‌های مه یا سرور ابری) بود. این مسئله به‌عنوان یک فرآیند تصمیم‌گیری مارکوف (MDP) فرموله شد.MDP معرفی‌شده شامل دو تصمیم‌گیرنده بود : کاربران IoT که مسئول انتخاب گره مه مناسب برای انتقال وظایف خود هستند و گره‌های مه که می‌توانند تصمیم بگیرند وظایف خاص را به سایر گره‌های مه یا به ابر منتقل کنند تا بار کاری به‌طور عادلانه توزیع شود. برای غلبه بر چالش‌های فضاهای بزرگ حالت و اقدام، یک روش مبتنی بر Q-Learning توسعه یافت تا یک سیاست ایده‌آل مشتق شود. نتایج شبیه‌سازی نشان داد که این روش زمان تأخیر را کاهش داده و توزیع بار کاری را در مقایسه با روش‌های جایگزین بهبود می‌بخشد. با این حال، طرح انتقال وظایف آن‌ها فاقد مکانیزم زمان‌بندی کارآمد و از پیش تعیین شده برای اختصاص لایه مناسب به هر وظیفه بود.

در برخی از مطالعات مرتبط [۱۵] ، از منطق فازی و الگوریتم‌های یادگیری عمیق تقویتی (DQN) برای حل چالش‌های انتقال وظایف در محیط رایانش مه-ابر استفاده شده است. در این روش‌ها، منطق فازی برای تصمیم‌گیری اولیه در خصوص انتخاب لایه مناسب (مانند لایه محلی، مه یا ابر) بر اساس ویژگی‌های وظایف از جمله اندازه، نیاز محاسباتی و حساسیت به تأخیر به کار گرفته شده است. سپس الگوریتم DQN با استفاده از فرآیند یادگیری عمیق تقویتی، گره بهینه مه را برای پردازش وظایف انتخاب کرده و توزیع بار متوازن را میان گره‌های همکار مه تضمین می‌کند. این روش‌ها بهبودهایی در کاهش تأخیر، مصرف انرژی و توزیع بار کاری نسبت به سایر روش‌های موجود نشان داده‌اند، اما همچنان در نظر گرفتن ناهمگنی گره‌ها و همکاری پیشرفته میان گره‌های مه نیازمند توسعه بیشتر است.

برای غلبه بر محدودیت‌های موجود در مطالعات پیشین، ما در این مطالعه از منطق فازی نوع ۲ برای تصمیم‌گیری انتقال وظایف و از الگوریتم PPO برای انتخاب گره مه بهینه و توزیع بار متوازن استفاده می‌کنیم. این رویکرد ضمن توجه به ناهمگنی وظایف و گره‌های محاسباتی، همکاری میان گره‌های مه و الزامات حساس به تأخیر، عملکرد بهتری در کاهش تأخیر، مصرف انرژی و افزایش نرخ انتقال وظایف ارائه می‌دهد.

۳ – معماری سیستم پیشنهادی

همان‌طور که پیش‌تر اشاره شد، چالش اصلی در رایانش مه این است که مشخص شود آیا وظایف باید به‌صورت محلی اجرا شوند یا به گره‌های مه و ابر منتقل شوند. چالش دیگر، توزیع مؤثر وظایف میان منابع موجود برای کاهش تأخیر و مصرف انرژی است. با وجود راه‌حل‌های ارائه‌شده برای این چالش‌ها در بخش



شکل ۱ : مدل سیستم پیشنهادی

قبل، همچنان فضا برای بهبود تأخیر و مصرف انرژی در سیستم‌های مه-ابر وجود دارد. این بخش یک استراتژی نوآورانه انتقال وظایف را معرفی می‌کند که با بهره‌گیری از زمان‌بندی وظایف و توزیع بار کاری، عملکرد برنامه‌های IoT را در معماری سه‌لایه مه-ابر بهبود می‌بخشد .معماری سلسله مراتبی پیشنهادی **شکل ۱** شامل سه لایه است:

- لایه IoT (لایه اول)
- لایه مه (لایه دوم)
- لایه ابر (لایه سوم)

این معماری شامل دستگاه‌های IoT در لایه اول و گره‌های میانی برای مسیریابی وظایف به گره‌های مه یا ابر می‌شود. در ادامه، ویژگی‌های عملیاتی هر لایه توضیح داده شده است:

لایهIoT

این لایه شامل دستگاه‌های هوشمند ناهمگن است که حجم قابل‌توجهی از داده‌ها را از طریق عملگرها مانند (موتورها و شیرها) ، حسگرها مانند (حسگرهای دما، رطوبت و حرکت) و دستگاه‌های IoT هوشمند مانند (دوربین‌های هوشمند، دستگاه‌های پوشیدنی و ابزارهای نظارت محیطی) تولید می‌کنند. علاوه بر این، این لایه گره‌های میانی مانند سوئیچ‌ها و

روترها را در بر می‌گیرد که به‌عنوان گیت‌وی‌های هوشمند عمل می‌کنند. به دلیل محدودیت در ظرفیت پردازشی و ذخیره‌سازی دستگاه هایIoT ، این گیت‌وی‌ها وظایف تولیدشده را به گره‌های مه یا ابر منتقل می‌کنند. در این لایه، از منطق فازی نوع ۲ برای ارزیابی و دسته‌بندی وظایف بر اساس ویژگی‌های آن‌ها استفاده می‌شود تا لایه مناسب برای انتقال وظایف (محلی، گره مه همکار یا ابر) مشخص شود.

لایه مه

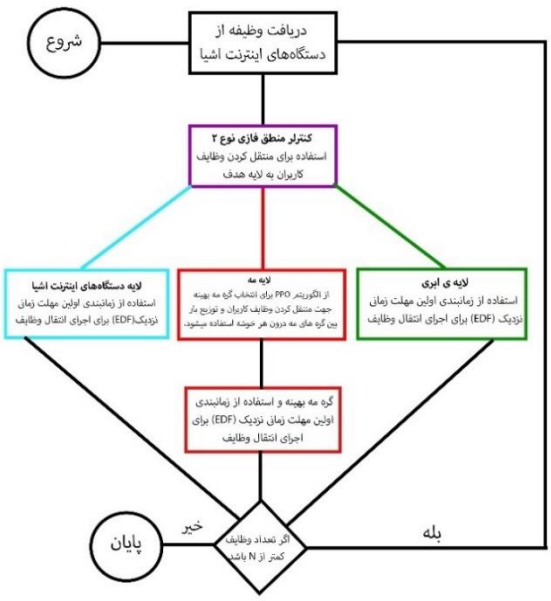
این لایه شامل گره‌های محاسباتی توزیع‌شده جغرافیایی است که به‌عنوان گره‌های مه شناخته می‌شوند. در این مرحله، از الگوریتم خوشه‌بندی DBSCAN برای خوشه‌بندی گره‌های مه بر اساس تراکم نقاط در فضای داده کار می‌کند. هر خوشه ترکیبی از منابع ناهمگن و همگن را برای برآورده کردن اهداف کیفیت خدمات در بر دارد. در هر خوشه، گره‌های مه با یک کنترل‌کننده خوشه مه (FCC) در ارتباط هستند و یک کنترل‌کننده مرکزی مه (FC) تمامی FCC ها را مدیریت می‌کند. در این لایه، از الگوریتم PPO برای انتخاب گره مه بهینه جهت منتقل کردن وظایف و توزیع متوازن بار کاری بین گره‌های همکار استفاده می‌شود.

لایه ابر

این لایه وظیفه پردازش وظایف سنگین را که محدودیت تأخیر کمتری دارند بر عهده دارد و هدف آن به حداقل رساندن زمان خدمات و برآورده کردن مهلت های کلی وظایف است. این لایه شامل ماشین‌های مجازی قدرتمند و متمرکزی است که به‌عنوان گره‌های محاسباتی عمل می‌کنند. بخش بعدی به تشریح جزئیات ویژگی‌های عملیاتی هر لایه در معماری پیشنهادی و الگوریتم‌های مورد استفاده برای زمان‌بندی وظایف و توزیع بار کاری جهت حل مسائل انتقال وظایف می‌پردازد. **شکل ۲** جریان کلی مراحل استراتژی انتقال وظایف را نشان می‌دهد. همچنین، **جدول ۱** انگیزه‌های اصلی تحقیق و الگوریتم‌های استفاده‌شده در این مطالعه را مشخص کرده.

جدول ۱ : انگیزه های اصلی تحقیق

لایه ها	ویژگی های عملیاتی هر لایه	الگوریتم استفاده شده
لایه ی ۱ (دستگاه های IOT)	۱ . جمع آوری داده ها ۲ . تصمیم گیری بهینه در مورد : الف) به صورت محلی ب) گره مه مشارکتی پ) ابر	الگوریتم منطق فازی نوع ۲
لایه ی ۲ (لایه مه)	۱ . خوشه بندی گره های مه ۲ . انتخاب بهینه گره های مه و توزیع بار	خوشه بندی DBSCAN الگوریتم PPO
لایه ی ۳ (لایه ابر)	مسئول پردازش وظایف فشرده ای که محدودیت های سخت تاخیر ندارند.	----



شکل ۲ : جریان کلی مراحل استراتژی انتقال وظایف

۴ – الگوریتم های استفاده شده در معماری پیشنهادی

این بخش به تشریح جزئیات ویژگی‌های عملیاتی هر لایه در معماری پیشنهادی می‌پردازد. همچنین، الگوریتم‌های مورد استفاده برای زمان‌بندی وظایف و توزیع بار کاری جهت حل چالش‌های انتقال وظایف معرفی می‌شوند.

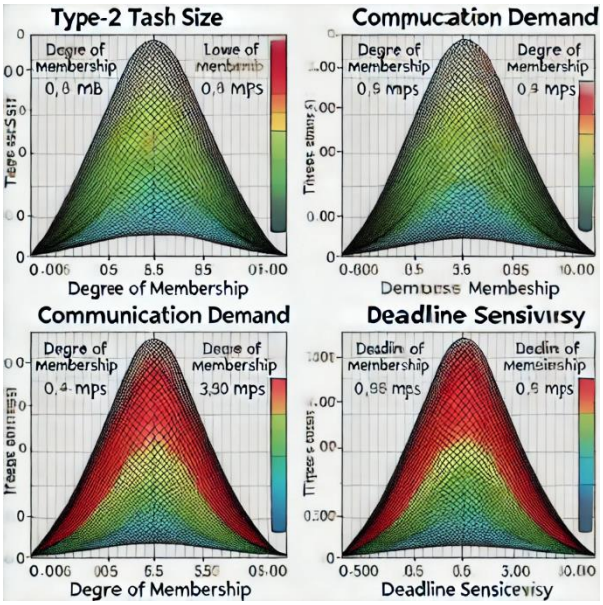
الف) لایه :IoT جمع‌آوری داده‌ها

لایه IoT حجم عظیمی از داده‌ها را از دستگاه‌های هوشمند ناهمگن IoT تولید می‌کند. این داده‌ها دارای ویژگی‌های متفاوتی مانند اندازه، نیاز محاسباتی، نیاز ارتباطی و محدودیت تأخیر هستند. مراحل آماده‌سازی برنامه ها به صورت شبه کد به شرح زیر است: برای اجرای روش پیشنهادی، عناصر زیر ایجاد می‌شوند : دستگاه‌های IoT ، گیت‌وی‌های هوشمند، گره‌های مه ناهمگن، مراکز داده ابری، برنامه‌های IoT و توپولوژی شبکه. در ادامه، شبیه‌ساز اجرا می‌شود. در هر بازه زمانی، هر وظیفه، یک برنامه IoT با پارامترهای متفاوتی (مانند اندازه، نیاز محاسباتی، نیاز ارتباطی و محدودیت تأخیر) ایجاد می‌کند. در مرحله نهایی این الگوریتم، بهینه‌ترین منابع محاسباتی برای انتقال وظایف کاربر انتخاب می‌شوند.

الگوریتم ۱: مقداردهی اولیه برنامه اینترنت اشیا و مهاجرت وظایف

```
1: Initialize IoT system with devices,
gateways, and computing layers
2: Deploy fog nodes and cloud data center
3: Define application structure and
communication framework
4: while application is running do
5:     Generate task and identify attributes
6:     Evaluate uncertainties using Type-2
Fuzzy Logic
7:     Decide execution (local, fog, or cloud)
using PPO
8: end while
```

در این مرحله، رویکرد پیشنهادی اطلاعات مرتبط با وظایف منتقل شده را جمع‌آوری می‌کند تا مکان مناسب برای انتقال وظایف (محلی در لایه IoT ، گره‌های مه همکار یا ابر) را تعیین کند. این فرآیند در **شکل ۳** نشان داده شده است.



شکل ۳ : تصمیم گیری انتقال وظایف با استفاده از منطق فازی نوع ۲

فرآیند زمان‌بندی و تخصیص وظایف

در این فاز، از کنترل‌کننده منطق فازی نوع ۲ برای ارزیابی و دسته‌بندی وظایف بر اساس ویژگی‌های آن‌ها استفاده می‌شود. هدف از این فرآیند، انتخاب لایه پردازشی مناسب است. وظایف در سه کلاس به شرح زیر دسته‌بندی می‌شوند:

کلاس محلی (Q1) : شامل وظایف دارای اولویت بالا که دارای تأخیر بسیار کم هستند و نیاز به منابع محاسباتی و ارتباطی زیادی ندارند. این وظایف در لایه IoT پردازش می‌شوند.

کلاس گره‌های مه همکار (Q2) : شامل وظایفی با تأخیر متوسط که نیاز به سطوح متوسطی از منابع محاسباتی و ارتباطی دارند. این وظایف به گره‌های مه همکار با منابع ناهمگن و همگن انتقال داده می‌شوند.

کلاس ابر (Q3) : شامل وظایفی که نیاز به منابع محاسباتی و ارتباطی بالا دارند و محدودیت تأخیر کمتری دارند. این وظایف در ابر پردازش می‌شوند. در این فرآیند، منطق فازی نوع ۲ برای دسته‌بندی درخواست‌های کاربر به صف‌های مناسب (Q1 ، Q2 ،Q3) استفاده می‌شود.

معماری منطق فازی نوع ۲

منطق فازی نوع ۲ از چهار بخش اصلی تشکیل شده است:

۱ - ورودی‌های فازی(Fuzzy Inputs)

۲ - فازی سازی(Fuzzification)

۳ - موتور استنتاج فازی(Inference Engine)

۴ - دی فازی سازی(Defuzzification)

ورودی های فازی :

در این مرحله، داده‌های ورودی که ممکن است دارای ابهام و عدم قطعیت باشند، به سیستم وارد می‌شوند. ورودی‌ها شامل اندازه وظیفه، نیاز محاسباتی، نیاز ارتباطی و محدودیت تأخیر باشند.

فازی‌سازی :

در منطق فازی نوع ۲، فازی سازی به جای استفاده از توابع عضویت قطعی مانند نوع ۱، از توابع عضویت فازی استفاده می‌کند. این توابع عضویت فازی مقادیر درجه عضویت را به‌صورت بازه‌ای و فازی نمایش می‌دهند. درجه عضویت برای هر مقدار ورودی، یک بازه از اعداد است که به آن تابع عضویت نوع ۲ یا Membership Function Type-2 می‌گویند. این بازه به‌صورت مجموعه فازی سطح بالاتر نمایش داده می‌شود. یک تابع عضویت نوع ۲ به صورت زیر تعریف می‌شود:

$$\{[1,0] \subseteq J_x \in X , u \in \mu_{\tilde{A}}(x,u) \mid x , (x,u)\} = \tilde{A} \quad (۱)$$

x	متغیر ورودی
u	درجه عضویت فازی نوع ۱ (مقادیر بین ۰ و ۱)
J_x	مجموعه‌ای از مقادیر ممکن برای$\mu(x)$
$\mu_{\tilde{A}}$ (x,u)	تابع عضویت نوع ۲ که مقدار توزیع عضویت را بر u مشخص می‌کند.

موتور استنتاج فازی :

در موتور استنتاج، قوانین فازی مانند نوع ۱ بر اساس منطق "اگر-آنگاه" اعمال می‌شوند. اما در منطق فازی نوع ۲، به دلیل توابع عضویت فازی، خروجی هر قانون یک مجموعه فازی نوع ۲ است، نه یک مقدار دقیق. قوانین فازی معمولاً به صورت زیر تعریف می‌شوند:

اگر زیاد = Task Size و کم = Delay Tolerance آنگاه محلی = Offload

در نوع ۲، محاسبات قوانین فازی باید برای تمامی مقادیر احتمالی در بازه عضویت انجام شود. این باعث می‌شود خروجی هر قانون، یک تابع فازی دو بعدی باشد.

دی فازی سازی :

در این مرحله، خروجی‌های فازی نوع ۲ که از موتور استنتاج تولید شده‌اند، به یک مقدار قطعی (Crisp) تبدیل می‌شوند. در منطق فازی نوع ۲، دی‌فازی سازی معمولاً به کمک کاهش نوع (Type Reduction) انجام می‌شود.

روند کاهش نوع :

کاهش نوع فرایندی است که یک مجموعه فازی نوع ۲ را به یک مجموعه فازی نوع ۱ تبدیل می‌کند .روش‌های معروف کاهش نوع عبارت‌اند از:

- روش کرنر
- روش میانگین مرکز

پس از کاهش نوع، خروجی به یک مقدار قطعی دی‌فازی‌سازی می‌شود. این مقدار خروجی نهایی برای تصمیم‌گیری در سیستم خواهد بود.

الگوریتم ۲: تصمیم‌گیری مهاجرت وظایف با استفاده از الگوریتم منطق فازی نوع ۲

```
Input: Application's Task Ti with parameters
(Task Size, Computational Demand, Delay
Constraint),
number of computing nodes Cni
Output: Optimal Offloading decision (Locally,
Collaborative fog, Cloud)
1: For all Tasks in Ti do
2:    F = Type2FuzzyLogicSystem(TaskSize,
ComputationalDemand , DelayConstraint)
3:    Compute uncertainty level U in F
4:    If F – U ≤ Flow then
5:       Allocate Ti on IoT layer
6:    else
7:       If F – U ≤ Fmedium then
8:          Allocate Ti on collaborative fog
9:          Start Algorithm 4
10:      else
11:         Allocate Ti on cloud
12:      End if
13:   End if
14: End for
```

ب) لایه مه: انتخاب گره مه بهینه با الگوریتم PPO

این لایه وظایفی را پردازش می‌کند که دارای نیازهای محاسباتی و ارتباطی متوسط و تأخیر متعادل هستند. در این مرحله، رویکرد پیشنهادی از الگوریتم خوشه‌بندی DBSCAN برای خوشه‌بندی گره‌های مه به خوشه‌های مختلف استفاده می‌شود. بر اساس [۱۳] فرآیند خوشه‌بندی DBSCAN شامل مراحل زیر است:

۱ - تعیین پارامترها:

- **Epsilon (ε)** : شعاعی که برای جستجوی همسایگان هر نقطه استفاده می‌شود.
- **MinPts** : حداقل تعداد نقاط مورد نیاز در یک شعاع ε برای تشکیل یک خوشه.

۲- طبقه‌بندی نقاط : نقاط داده به سه دسته تقسیم می‌شوند:

- **Core Points (نقاط اصلی)** : نقاطی که حداقل MinPtsنقاط دیگر در شعاع ε خود دارند.
- **Border Points (نقاط مرزی)** : نقاطی که خودشان Core نیستند اما در شعاع ε یک نقطه Core قرار دارند.
- **Noise Points (نویز)** : نقاطی که نه Core هستند و نه در شعاع ε یک Core Point قرار دارند.

۳- انتخاب نقطه شروع : الگوریتم از یک نقطه شروع می‌کند و همسایگان آن را در شعاع ε جستجو می‌کند.

۴ - تشکیل خوشه‌ها:

- اگر نقطه شروع یک Core Point باشد:
  ۱. خوشه‌ای جدید تشکیل می‌شود.
  ۲. تمامی نقاط در شعاع ε که به این نقطه متصل هستند، به خوشه اضافه می‌شوند.
  ۳. این فرآیند به صورت بازگشتی ادامه می‌یابد تا زمانی که دیگر نقطه‌ای برای اضافه شدن وجود نداشته باشد.
- اگر نقطه شروع Core نباشد: نقطه بررسی شده به عنوان نویز علامت‌گذاری می‌شود یا به خوشه‌ای موجود اضافه می‌شود (اگر نقطه مرزی باشد).

۵ - تکرار برای نقاط باقی‌مانده : این فرآیند برای تمام نقاط داده تکرار می‌شود تا زمانی که تمامی نقاط بررسی شوند. هر خوشه دارای یک کنترل‌کننده خوشه مه (نمایش داده شده با رنگ زرد) است که مسئول مدیریت منابع در سراسر خوشه و توزیع مؤثر بار کاری است. این کنترل‌کننده همچنین تعیین می‌کند که کدام گره مه می‌تواند درخواست‌های ورودی را پردازش کند. هر گره مه در یک خوشه با کنترل‌کننده خوشه مه (FCC) ارتباط برقرار می‌کند. کنترل‌کننده مه (FC) مسئول اتصال تمامی FCC های لایه مه است. در فرآیند انتقال

وظایف، اگر یک خوشه در لایه مه نتواند به دلیل نیازهای پردازشی یک وظیفه را مدیریت کند، کنترل‌کننده مه بررسی می‌کند که آیا وظیفه می‌تواند به خوشه دیگری ارسال شود یا خیر. اگر پاسخ مثبت باشد، وظیفه به خوشه جایگزین ارسال می‌شود؛ در غیر این صورت به لایه ابر منتقل خواهد شد.

استفاده از الگوریتم PPO برای انتقال هوشمند وظایف

از الگوریتم PPO (Proximal Policy Optimization) برای تصمیم‌گیری هوشمندانه در انتقال وظایف و متعادل‌سازی بار کاری استفاده شده است. الگوریتم PPO با بهینه‌سازی یک سیاست θ^{π} اطمینان می‌دهد که به‌روزرسانی‌های سیاست در یک محدوده قابل اعتماد باقی می‌مانند و از تغییرات شدید جلوگیری می‌کند. مراحل اجرای این الگوریتم به شرح زیر است:

۱ - تعریف محیط:

- **حالات (s_t) :** شامل اطلاعات مربوط به وظایف فعلی و وضعیت گره‌های مه (مانند منابع موجود، بار کاری و تأخیر).
- **اقدامات (a_t) :** انتخاب بهترین گره مه یا خوشه برای انتقال یک وظیفه.
- **پاداش (r_t) :** معیارهایی مانند کاهش زمان پردازش، کاهش تأخیر و متعادل‌سازی بار کاری.

۲- بهینه‌سازی سیاست:

الگوریتم PPO تابع هدف زیر را بیشینه می‌کند:

$$E_t [\min (r_t(\theta)A_t, clip(r_t(\theta), 1- \epsilon, 1 + \epsilon)A_t)] =L^{CLIP}(\theta) \quad (۲)$$

در این فرمول :

$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{old}\theta(a_t|s_t)}$ نسبت احتمال سیاست جدید به سیاست قدیمی است.

A_t تابع مزیت است که نشان می‌دهد یک اقدام نسبت به میانگین عملکرد چقدر بهتر است. ϵ یک مقدار کوچک است که محدوده تغییرات را کنترل می‌کند.

۳ - جمع‌آوری داده‌ها و تجربه‌ها:

عامل با محیط تعامل دارد و تجربیات را به شکل $(a_t ، r_t ، s_t ، s_{t+1})$ جمع آوری و در حافظه ذخیره می‌کند.

۴ - به‌روزرسانی سیاست:

با استفاده از داده‌های جمع‌آوری‌شده، پارامترهای سیاست θ از طریق بیشینه‌سازی $L^{CLIP}(\theta)$ به‌روزرسانی می‌شوند. روش ها ی مبتنی بر گرادیان برای بهینه سازی استفاده می‌شوند.

مزایای استفاده از PPO در رایانش مه

الگوریتم PPO با جلوگیری از به‌روزرسانی‌های شدید، پایداری و قابلیت اطمینان در فرآیند یادگیری را فراهم می‌کند. این ویژگی در محیط‌های پویا مانند رایانش مه که در آن شرایط شبکه و منابع به‌طور مداوم تغییر می‌کنند، بسیار مفید است. استفاده از PPO باعث می‌شود وظایف به صورت کارآمد بین گره‌های مه توزیع شوند، تأخیر و مصرف انرژی کاهش یابد و بار کاری به طور متعادل تقسیم شود. این روش شامل دو مرحله آموزش و آزمایش است که هدف آن منتقل کردن مؤثر وظایف در محیط محاسبات مه-ابر می‌باشد. در ابتدا، وظایف و گره‌های مه بر اساس حداقل تأخیر مورد نیاز و ظرفیت محاسباتی مرتب‌سازی می‌شوند. سپس شبکه‌های سیاست و ارزش (Value Network) الگوریتم PPO مقداردهی اولیه می‌شوند. همچنین، یک حافظه تجربی برای ذخیره تجربیات شامل حالت، اقدام، پاداش و مزیت (Advantage) ایجاد می‌شود.

مرحله آموزش

در مرحله آموزش، فرآیند زیر طی می‌شود:

۱. عامل PPO بر اساس شبکه سیاست اقدام به انتخاب گره مه برای انتقال وظایف می‌کند.
۲. پس از اجرای هر اقدام، پاداش به‌دست می‌آید و وضعیت جدید ثبت می‌شود.
۳. مزیت هر اقدام با استفاده از برآورد مزیت تعمیم یافته محاسبه می‌شود تا میزان عملکرد بهتر از سیاست مرجع را مشخص کند:

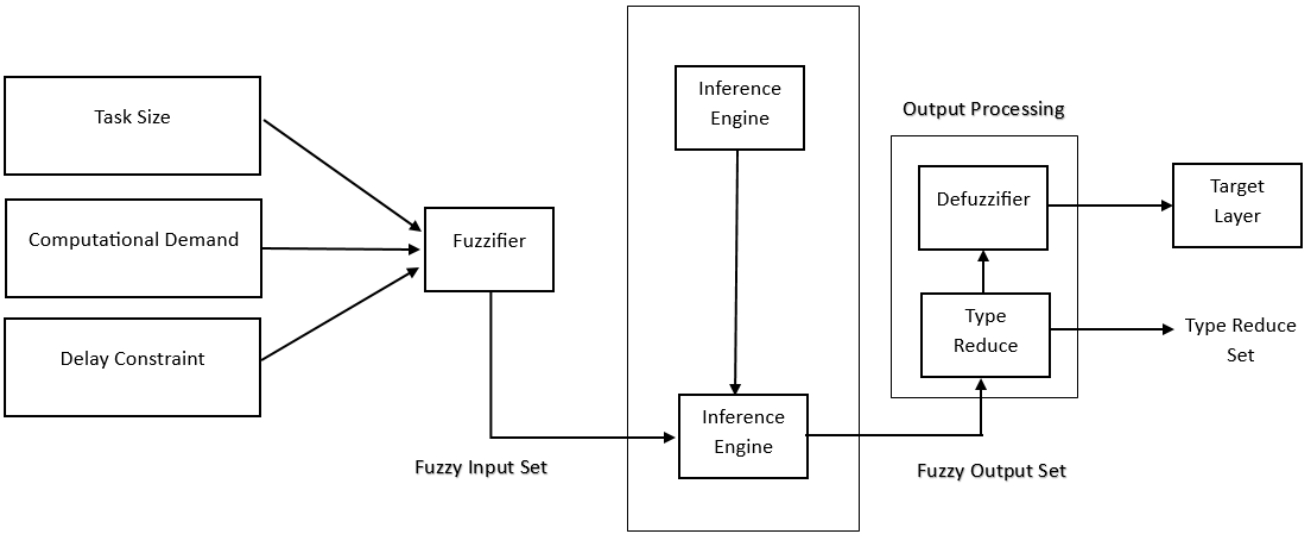
$$A_t^{\wedge} = \sum (\lambda_{\Upsilon}{}^t) (V(S_t) - V(S_{t+1}) + R_t) \quad (۳)$$

٤. داده های تجربی در حافظه بازپخش ذخیره می‌شوند.
٥. در فواصل زمانی مشخص، شبکه سیاست π با استفاده از هدف PPO-Clip بهینه سازی می‌شود
٦. شبکه ارزش $V(S)$ نیز با مینیمم‌سازی خطای میانگین مربعات (MSE) به‌روزرسانی می‌شود :

$$L(V) = (V_{\Upsilon} (s') + R) - V(S))^2 \quad (۴)$$

مرحله آزمایش

در مرحله آزمایش، عامل آموزش‌دیده برای تخصیص وظایف به گره‌های مه از شبکه سیاست استفاده می‌کند:



شکل ۵ : شماتیک منطق فازی نوع ۲

۱. برای هر وظیفه، وضعیت فعلی به شبکه سیاست داده شده و احتمال اقدام‌ها محاسبه می‌شود.
۲. اقدام (گره مه) با بیشترین احتمال به‌عنوان مقصد بهینه وظیفه انتخاب می‌شود.
۳. وضعیت جدید پس از تخصیص ثبت شده و این فرآیند برای تمام وظایف تکرار می‌شود.

الف. مطالعه موردی

سناریوی مورد بررسی شامل چندین دستگاه IoT از جمله سیستم‌های دوربین مداربسته هوشمند، خودروهای خودران و تلفن‌های هوشمند است. این دستگاه‌ها در حوزه‌هایی مانند حمل‌ونقل، تولید، بهداشت و امنیت کاربرد دارند. این برنامه‌ها

انتقال وظایف به لایه ابر

در صورتی که وظیفه‌ای به دلیل محدودیت‌های منابع در گره‌های مه نتواند اجرا شود، به لایه سوم یعنی ابر منتقل می‌شود. در این لایه، وظایف با استفاده از الگوریتم زمان‌بندی اولین مهلت زمانی نزدیک (EDF) پردازش می‌شوند تا کارایی کلی سیستم با حداقل تأخیر تضمین شود.

۵ – ارزیابی عملکرد

این بخش عملکرد طرح پیشنهادی انتقال وظایف را با مقایسه با روش‌های مرجع بررسی می‌کند. معیارهای ارزیابی شامل میانگین کاهش زمان تأخیر، مصرف انرژی متوسط، استفاده از شبکه، بازدهی، و نرخ انتقال وظایف می باشد.

الگوریتم ۳: انتخاب بهینه گره مه Fog با استفاده از روش PPO

```
Input: Tasks (Ti), Fog Nodes (Fi)
Output: Task assignments to optimal fog nodes
1: Sort tasks by minimum required delay
2: Sort fog nodes by computational capacity
3: Initialize PPO parameters and networks (Policy π, Value V)
4: Train PPO:
5:    For each episode and task, select action A using π
6:    Execute A, observe reward R, update state S'
7:    Compute advantage Ât using GAE
8:    Optimize π with PPO-Clip and update V network
9:    Periodically clear experience buffer
10: Assign tasks:
11:   For each task, use π to select the best fog node
12:   Store (Task, Optimal Fog Node) in TaskAssignments
13: Return TaskAssignments
```

نیازمند پردازش داده‌های حجیم و تأخیر پایین هستند. هر برنامه شامل وظایف مختلفی است؛ برای مثال، برنامه‌های CCTV هوشمند شامل وظایفی مانند تشخیص حرکت و شناسایی چهره می‌باشند. این وظایف از نظر مشخصاتی مانند اندازه، نیازهای محاسباتی، نیازهای ارتباطی و تأخیر تفاوت دارند. با توجه به این ویژگی‌ها و منابع موجود، وظایف می‌توانند در منابع محاسباتی محلی، گره‌های مه مشارکتی، یا ابر مستقر و اجرا شوند. جزئیات بیشتر درباره مدل‌های معماری پیشنهادی در بخش ۳ توضیح داده شده است.

ب. تنظیمات شبیه‌سازی

کارایی استراتژی پیشنهادی با استفاده از شبیه‌ساز iFogSim ارزیابی شد. آزمایش‌ها روی سیستمی با مشخصات زیر انجام شدند:

پردازنده	Intel i7 2.4 GHz
رم	16 GB
سیستم عامل	Windows

توپولوژی شبکه iFogSim در **شکل ۶** نشان داده شده است.

پ. روش‌های مرجع برای مقایسه

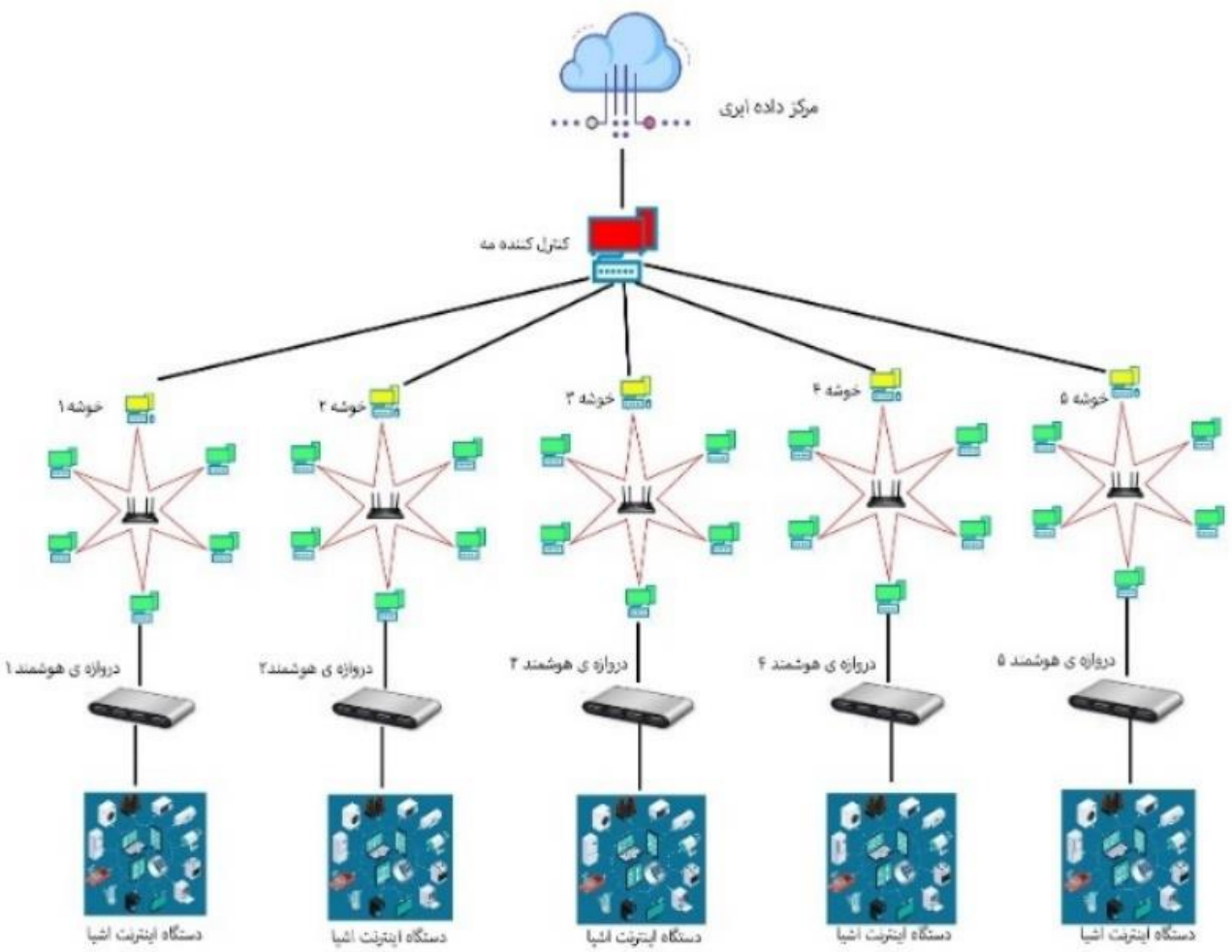
برای نشان دادن اثربخشی استراتژی پیشنهادی، عملکرد آن با روش‌های زیر مقایسه شده است:

۱ - NON-OFFLOAD : در این روش، وظایف به‌صورت محلی اجرا می‌شوند و هیچ‌گونه انتقال وظایف به گره‌های مه یا ابر انجام نمی‌شود.

۲ – FIRST FIT – FF : در این روش، مقصد وظیفه بر اساس ترتیب گره‌های مه یا ابر در شبکه تعیین می‌شود [۱۴]. فرآیند انتخاب به این صورت است که گره مه اول برای انتقال وظایف بررسی می‌شود در صورت مناسب بودن، انتخاب می‌شود. در غیر این صورت، گره دوم ارزیابی می‌شود و این روند به ترتیب ادامه می‌یابد تا در نهایت به ابر ختم شود.

۳ – GASDEO (Greedy Auto-Scaling Deep Reinforcement Learning-Based Offloading) : این روش از الگوریتم یادگیری تقویتی عمیق (DRL) برای انتخاب مقصد مناسب وظیفه استفاده می‌کند. در این استراتژی، تصمیم انتقال وظایف با هدف کاهش میانگین تأخیر انجام می‌شود. روش GASDEO شامل دو مرحله است ، ابتدا ، گره‌های مه به صورت محلی و با تکنیک حریصانه (Greedy) ارزیابی می‌شوند سپس، بهترین مقصد برای اجرای وظایف با استفاده از DRL مشخص می‌شود.

۴ – NAFITO-FLA (A Novel Approach for IoT Tasks Offloading Based on Fuzzy Logic Algorithm): این روش از منطق فازی برای شناسایی مکان مناسب انتقال وظایف استفاده می‌کند. جزئیات بیشتر در این مورد در [۱۵] توضیح داده شده است.



شکل ۶ : توپولوژی شبکه ی iFogSim

ت. معیارهای عملکرد

در این بخش، پارامترهای عملکردی به همراه فرمول‌های ریاضی مربوطه تعریف می‌شوند.

۱ - تأخیر(Delay)

تأخیر یا زمان تأخیر هنگامی رخ می‌دهد که یک وظیفه برای پردازش به یک لایه مشخص ارسال می‌شود. نرخ تأخیر برای هر

نوع وظیفه با توجه به ویژگی‌های خاص آن متفاوت است. به عنوان مثال، زمان مورد نیاز برای پردازش و انتقال یک وظیفه در محلی با زمانی که برای پردازش و اجرای وظایف در گره‌های مه یا ابر نیاز است، متفاوت خواهد بود.تأخیر که معمولاً بر حسب میلی‌ثانیه تخمین زده می‌شود، یکی از معیارهای مهم برای ارزیابی عملکرد سیستم‌های محاسبات مه است، به‌ویژه برای برنامه‌هایی که به حساسیت زمانی بالا نیاز دارند. تأخیر مربوط به یک وظیفه که توسط یک دستگاه IoT منتقل می‌شود، تحت تأثیر چهار عامل است:

تاخیر صف بندی (Q_d) ، تأخیر انتقال (T_d) ، تأخیر انتشار (Pro_d) ، تأخیر پردازش (P_d)

مجموع تأخیر از رابطه زیر محاسبه می‌شود:

$$Delay = Q_d + T_d + Pro_d + P_d \qquad (۱-۵)$$

۲ - مصرف انرژی(Energy Consumption)

مصرف انرژی برای کل توپولوژی شبکه با استفاده از معادله زیر محاسبه می‌شود:

$$E = E_c + (T_n - T_{lu}) * P_h \qquad (۲-۵)$$

EC	انرژی مصرفی فعلی
T_n	زمان فعلی
T_{lu}	زمان آخرین به‌روزرسانی استفاده
P_h	توان میزبان در آخرین به‌روزرسانی

مدل های مختلفی برای کاهش مصرف انرژی در شبکه‌های مه و IoT توسعه یافته‌اند. این مدل‌ها تلاش می‌کنند چالش‌های مربوط به مصرف انرژی و تأخیر را در محیط‌های محاسبات مه برطرف کنند.

۳ – استفاده از شبکه (Network Usage)

میزان استفاده از شبکه به اندازه وظیفه منتقل شده در یک زمان مشخص و تأخیر وابسته است. استفاده از شبکه با رابطه زیر محاسبه می‌شود :

$$\text{Network usage} = \text{Latency} * \text{tupleSize} \qquad (۵-۳)$$

۴-بازدهی(Throughput)

بازدهی به مقدار داده یا اطلاعاتی اشاره دارد که می‌تواند در یک بازه زمانی پردازش شود. این مقدار بر حسب بیت در ثانیه (bps)اندازه‌گیری می‌شود و با رابطه زیر بیان می‌گردد:

$$TH = \frac{\text{Nspt}}{Ti} \qquad (۴-۵)$$

N_{spt}	تعداد کل وظایفی که با موفقیت منتقل شده‌اند.
T_i	زمان انتقال وظایف

۵ – نرخ انتقال وظایف (Offloading Rate)

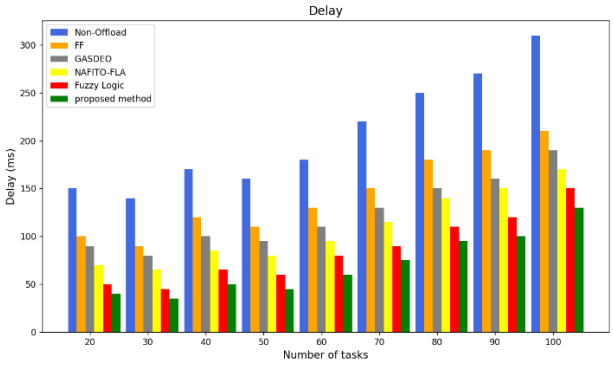
نرخ انتقال وظایف به میزان انتقال وظایف یا بارهای کاری از یک دستگاه محلی به گره مه، ابر یا گره‌های مه مشارکتی دیگر گفته می‌شود. هدف اصلی نرخ انتقال وظایف ، بهینه‌سازی استفاده از منابع و کاهش بار محاسباتی دستگاه‌های محلی از طریق انتقال وظایف به منابع قوی‌تر یا در دسترس در شبکه است. این نرخ به صورت ریاضی با رابطه زیر بیان می‌شود:

$$OFR = \frac{Tf}{N} * 100 \qquad (۵-۵)$$

T_f	تعداد کل وظایفی که با موفقیت منتقل شده‌اند
N	تعداد کل وظایف، شامل وظایف محلی و منتقل شده

ث. نتایج و بحث

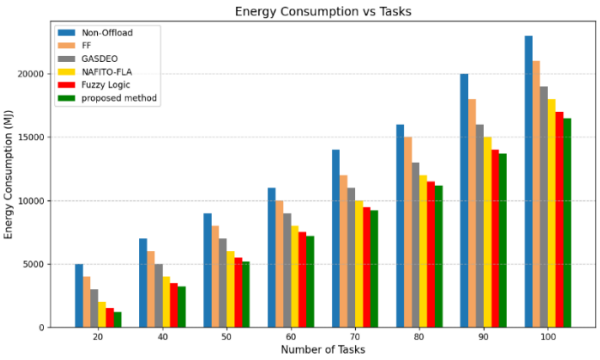
در این بخش، نتایج به‌دست‌آمده از ارزیابی طرح پیشنهادی و بحث پیرامون عملکرد آن بررسی می‌شود.



شکل ۷ : نمودار تاخیر در برابر وظایف

۱ – تأخیر(Delay)

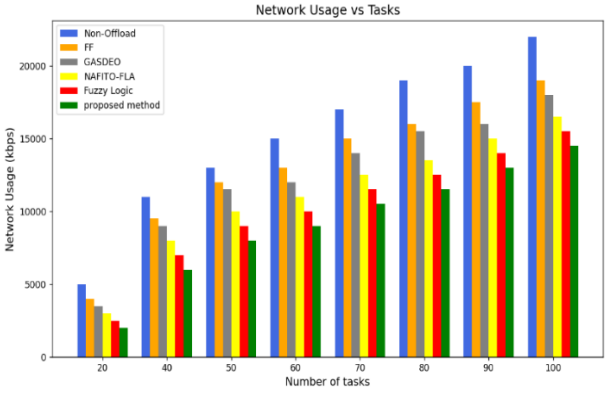
در ابتدا میانگین تأخیر هر وظیفه به دلیل فرآیند انتقال وظایف مورد بررسی قرار می‌گیرد .**شکل ۷** عملکرد روش پیشنهادی را در مقایسه با روش‌های مرجع نشان می‌دهد. مطابق این شکل، با افزایش تعداد وظایف، میانگین تأخیر نیز افزایش می‌یابد. این امر به دلیل ایجاد بار بیشتر روی گره‌های مه، افزایش تأخیر صف‌بندی و نیاز بیشتر به پهنای باند شبکه رخ می‌دهد. روش پیشنهادی در مقایسه با روش‌های مرجع در تمامی موارد عملکرد بهتری دارد. در روش‌های پایه، گره مه بهینه برای پردازش وظایف در نظر گرفته نمی‌شود که منجر به افزایش تأخیر خدمات می‌شود. روش Non-Offload بیشترین تأخیر را تجربه می‌کند، زیرا تمام وظایف به‌صورت محلی پردازش می‌شوند.



شکل ۸ : مصرف انرژی در برابر وظایف

۲ – مصرف انرژی(Energy Consumption)

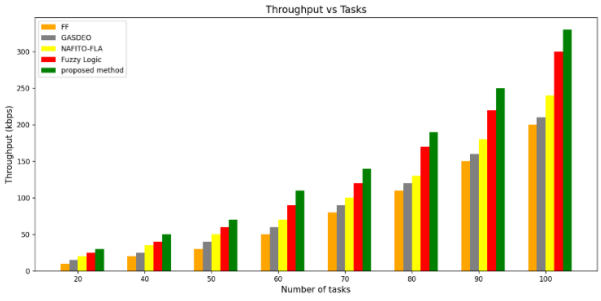
شکل ۸ کارایی روش پیشنهادی را از نظر مصرف انرژی در مقایسه با روش‌های موجود نشان می‌دهد. مطابق این شکل، مصرف انرژی تقریباً به‌صورت خطی با افزایش تعداد وظایف افزایش می‌یابد. این کاهش مصرف انرژی به دلیل استفاده از معماری پیشنهادی با روش بهینه برای زمان‌بندی وظایف و اجرای آن‌ها است. وظایف به گره‌های مه قدرتمندتر منتقل می‌شوند، که مصرف انرژی در گره‌های ضعیف‌تر را کاهش داده و بازدهی کلی را بهبود می‌بخشد. برنامه‌ریزی مؤثر وظایف و استراتژی‌های انتقال وظایف نیز می‌توانند با تعیین مکان پردازش بهینه، مصرف انرژی را به حداقل برسانند. روش‌های Non-Offload و FF به دلیل پردازش محلی و انتخاب غیر بهینه گره‌ها، مصرف انرژی بالاتری دارند.



شکل ۹ : میزان استفاده از شبکه در برابر وظایف

۳ – استفاده از شبکه(Network Usage)

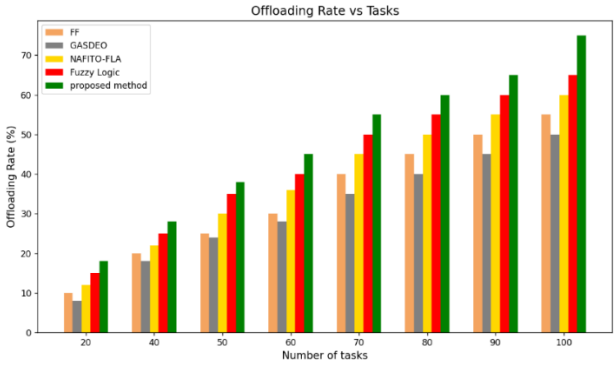
زمانی که ترافیک روی سرورهای ابری افزایش یابد، تنها از منابع ابری استفاده می‌شود. این موضوع باعث بهبود استفاده از شبکه در سرورهای ابری می‌شود. در سیستم‌های توزیع‌شده جغرافیایی، هر گره مه به منطقه‌ای خاص اختصاص دارد که درخواست‌ها از آن منطقه را مدیریت می‌کند. این موضوع باعث کاهش استفاده از شبکه در این نوع سناریوها می‌شود. **شکل ۹** نتایج مقایسه استفاده از شبکه را نشان می‌دهد. روش Non-Offload از منابع شبکه استفاده نمی‌کند. با این حال، روش‌های FF، GASDEO و NAFITO-FLA نتایجی تقریباً مشابه دارند. روش پیشنهادی توانسته در این معیار بهتر عمل کند و حتی با افزایش اندازه شبکه نیز کارایی خود را حفظ کند. استراتژی‌های هوشمند آفلودینگ در روش پیشنهادی می‌توانند با تعیین مکان پردازش مناسب، استفاده از شبکه را بهینه کنند.



شکل ۱۰ : عملکرد بازدهی در برابر وظایف

۴ – بازدهی(Throughput)

شکل ۱۰ عملکرد بازدهی میانگین را در برابر تعداد وظایف نشان می‌دهد. روش پیشنهادی در مقایسه با سایر روش‌ها بهترین عملکرد را در این معیار داشته است. این به دلیل توزیع بهتر وظایف بین چندین گره و سرور است. روش پیشنهادی اجازه می‌دهد وظایف متعددی به‌صورت همزمان بر روی منابع مختلف اجرا شوند، که منجر به پردازش سریع‌تر وظایف و بهبود بازدهی می‌شود. همچنین، تکنیک‌های تعادل بار مؤثر می‌توانند وظایف و پردازش داده‌ها را به‌طور یکنواخت بین گره‌های مه توزیع کرده و از تراکم شبکه جلوگیری کنند.



شکل ۱۱ : نرخ انتقال وظایف در برابر وظایف

۵ - نرخ انتقال وظایف (Offloading Rate)

نرخ انتقال وظایف به میزان وظایفی اشاره دارد که از دستگاه محلی به گره مه، ابر یا گره‌های مه دیگر منتقل می‌شوند .**شکل ۱۱** عملکرد نرخ انتقال وظایف میانگین را در برابر تعداد وظایف نشان می‌دهد. این معیار نشان می‌دهد که روش پیشنهادی کارایی بهتری نسبت به سایر روش‌ها دارد. روش پیشنهادی منابع را به‌طور مؤثر استفاده می‌کند و وظایف را به منابع قدرتمندتر انتقال می‌دهد. این فرآیند باعث بهبود عملکرد، کاهش مصرف انرژی و استفاده بهینه از منابع می‌شود. همچنین، زمان‌بندی مؤثر و اولویت‌بندی سریع وظایف به افزایش نرخ انتقال وظایف کمک می‌کند.

ج. محدودیت‌ها

این بخش محدودیت‌های روش پیشنهادی را در زمینه‌های توپولوژی ایستا، امنیت و حفظ حریم خصوصی روشن می‌کند:

۱ - توپولوژی ایستا(Static Topology)

یکی از محدودیت‌های این روش وابستگی به توپولوژی ایستا است، به‌طوری که موقعیت دستگاه‌های IoT و گره‌های مه ثابت فرض شده‌اند. این در حالی است که در سناریوهای واقعی IoT ، معمولاً نودهای متحرک حضور دارند که پویایی بیشتری را به سیستم اضافه می‌کنند و در این مطالعه به‌صراحت به آن پرداخته نشده است [۱۷]. پژوهش‌های اخیر نشان داده‌اند که استفاده از توپولوژی‌های پویا و طراحی مکانیزم‌های انتقال وظایف مبتنی بر حرکت، می‌تواند عملکرد سیستم را در چنین سناریوهایی بهبود بخشد[۱۸].

۲ - امنیت و حریم خصوصی(Security and Privacy)

سیستم‌های مه-ابر شامل انتقال و پردازش داده‌های حساس هستند که نگرانی‌هایی در مورد امنیت و حفظ حریم خصوصی اطلاعات ایجاد می‌کنند [۱۹]. برای کاهش این خطرات:

- پروتکل‌های رمزنگاری قوی باید برای انتقال داده استفاده شوند، مانند پروتکل‌های SSL/TLS و رمزنگاری انتها به انتها [۲۰] (E2EE).
- مکانیزم‌های کنترل دسترسی مناسب باید اطمینان حاصل کنند که تنها موجودیت‌های مجاز به سیستم دسترسی دارند. این مکانیزم‌ها می‌توانند شامل احراز هویت مبتنی بر توکن یا بلاک‌چین باشند [۲۱].
- استفاده از تکنیک‌های حفظ حریم خصوصی مانند داده‌های دیفرانسیلی (Differential Privacy) می‌تواند خطرات افشای اطلاعات حساس را کاهش دهد[۲۲].

ادغام اقدامات امنیتی در کل معماری، انجام بازرسی‌های امنیتی دوره‌ای، و همکاری میان ذینفعان از مؤلفه‌های حیاتی این استراتژی هستند [۲۳].

۶- نتیجه گیری

در این مطالعه، ما یک استراتژی جدید برای برون‌سپاری وظایف ارائه دادیم که عملکرد برنامه‌های اینترنت اشیاء (IoT) را در محیط‌های مه محاسباتی بهبود می‌بخشد. ایده اصلی طرح پیشنهادی شامل پردازش داده‌های حس‌شده تولید شده توسط دستگاه‌های اینترنت اشیاء در لایه‌های مختلف معماری مه‌محاسباتی است .الگوریتم‌های منطق فازی نوع ۲ برای تصمیم‌گیری بهینه در مورد برون‌سپاری وظایف به لایه پردازشی هدف، چه در گره‌های مه محلی، چه گره‌های مه تعاملی یا ابر، با در نظر گرفتن چهار نیازمندی مختلف وظایف استفاده شدند. سپس، از روش یادگیری تقویتی عمیق مبتنی بر الگوریتم PPO برای انتخاب گره مه بهینه در میان گره‌های مه تعاملی جهت برون‌سپاری وظایف استفاده شد. در ادامه، یک روش زمان‌بندی وظایف با اولویت اولین مهلت زمانی نزدیک (Earliest Deadline First) برای زمان‌بندی وظایف بر روی تمام منابع محاسباتی پیشنهاد شد. در نهایت، نتایج

شبیه‌سازی در چندین معیار عملکردی مقایسه شده و نشان داده شد که استراتژی برون‌سپاری پیشنهادی در معماری مه، از سایر روش‌های موجود بهتر عمل می‌کند.

در کارهای آینده، ما بر روی برون‌سپاری وظایف بدون در نظر گرفتن پارامترهای امنیتی تمرکز خواهیم کرد. به عنوان مثال، چه اتفاقی می‌افتد اگر یک وظیفه به یک گره مه در معرض خطر برون‌سپاری شود؟ ما در حال حاضر در نظر داریم این ویژگی را برای پیاده‌سازی‌های آینده بررسی کنیم. علاوه بر این، این مطالعه از یک توپولوژی ثابت استفاده کرده است و فرض کرده که دستگاه‌های اینترنت اشیاء و گره‌های مه در موقعیت‌های ثابت قرار دارند. در حالی که در سناریوهای واقعی اینترنت اشیاء ممکن است گره‌های متحرک در حال کار باشند. در نتیجه، ما قصد داریم در مرحله بعدی این تحقیق، تأثیر توپولوژی دینامیک بر کارایی طرح برون‌سپاری را بررسی کنیم.

۷ – منابع

[1] A. Ahmed, S. Abdullah, S. Iftikhar, I. Ahmad, S. Ajmal, and Q. Hussain,''A novel blockchain based secured and QoS aware IoT vehicular network in edge cloud computing,'' IEEE Access, vol. 10, pp. 77707–77722, 2022,doi: 10.1109/ACCESS.2022.3192111.

[2] Y.-L. Jiang, Y.-S. Chen, S.-W. Yang, and C.-H. Wu, ''Energyefficient task offloading for time-sensitive applications in fog computing,'' IEEE Syst. J., vol. 13, no. 3, pp. 2930–2941, Sep. 2019, doi: 10.1109/JSYST.2018.2877850.

[3] M. Izhar, S. A. A. Naqvi, A. Ahmed, S. Abdullah, N. Alturki, and L. Jamel, ''Enhancing healthcare efficacy through IoT-edge fusion: A novel approach for smart health monitoring and diagnosis,'' IEEE Access, vol. 11, pp. 136456–136467, 2023, doi: 10.1109/ACCESS.2023.3337092.

[4] F. Bonomi, R. Milito, P. Natarajan, and J. Zhang, "Fog Computing and Its Role in the Internet of Things," Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, pp. 13-16, Aug. 2012

[5] M. Zorzi, L. G. Giordano, and A. F. M. D. M. P. D. T. M. W. G. B. F. S. M. M. M. R. S. C., "Task Offloading for IoT Applications: A Survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2641-2675, 2016.

[6] L. A. Zadeh, "Type-2 Fuzzy Logic: Theory and Applications," Proceedings of the IEEE, vol. 90, no. 5, pp. 1032-1052, May 2002.

[7] J. Schulman, F. Wolski, and R. H. D. S. McCool, "Proximal Policy Optimization Algorithms," arXiv:1707.06347, 2017.

[8] A. K. M. A. Hossain, M. A. Chowdhury, A. Gani, and R. Ranjan, "iFogSim: A toolkit for modeling and simulation of resource management in the Internet of Things," Proceedings of the 2017 IEEE/ACM International Conference on Utility and Cloud Computing (UCC), 2017.

[9] S. Bebortta, S. S. Tripathy, U. M. Modibbo, and I. Ali, ''An optimal fogcloud offloading framework for big data optimization in heterogeneous IoT networks,'' Decis. Anal. J., vol. 8, Sep. 2023, Art. No. 100295, doi: 10.1016/j.dajour.2023.100295.

[10] D. H. Abdulazeez and S. K. Askar, ''Offloading mechanisms based on reinforcement learning and deep learning algorithms in the fog computing nvironment,'' IEEE Access, vol. 11, pp. 12555–12586, 2023, doi: 10.1109/ACCESS.2023.3241881.

[11] C. Chakraborty, K. Mishra, S. K. Majhi, and H. K. Bhuyan, ''Intelligent latency-aware tasks prioritization and offloading strategy in distributed fog-cloud of things,'' IEEE Trans. Ind. Informat., vol. 19, no. 2, pp. 2099–2106, Feb. 2023, doi: 10.1109/TII.2022.3173899.

[12] S. Aljanabi and A. Chalechale, ''Improving IoT services using a hybrid fog-cloud offloading,'' IEEE Access, vol. 9, pp. 13775–13788, 2021, doi: 10.1109/ACCESS.2021.3052458.

*[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD)*, Portland, OR, USA, 1996, pp. 226–231.*

[14] Q.-H. Nguyen and T.-A. Truong Pham, ''Studying and developing a resource allocation algorithm in fog computing,'' in Proc. Int.Conf. Adv. Comput. Appl. (ACOMP), Nov. 2018, pp. 76–82, doi:10.1109/ACOMP.2018.00020.

[15] Dezeen H. Abdulazeez and Shavan K. Askar , "A Novel Offloading Mechanism Leveraging Fuzzy Logic and Deep Reinforcement Learning to Improve IoT Application Performance in a Three-Layer Architecture Within theFog-Cloud Environment" , date of current version 20 March 2024 , ACCESS.2024.3376670

[16] M. K. Hussein and M. H. Mousa, ''Efficient task offloading for IoT-based applications in fog computing using ant colony optimization,'' IEEE Access, vol. 8, pp. 37191–37201, 2020, doi: 10.1109/ACCESS.2020.2975741.

[17] A. Leon-Garcia et al., "Dynamic Topology in IoT: Challenges and Solutions," IEEE Internet of Things Journal, vol. 7, no. 3, pp. 2152-2165, 2020.

[18] X. Wang et al., "Mobility-Aware Offloading in IoT," Proceedings of IEEE ICC, 2019.

[19] F. Bonomi et al., "Fog Computing and Its Role in IoT Security," Proceedings of ACM MCC Workshop, 2012.

[20] M. Abomhara et al., "Cyber Security and the Internet of Things," Future Generation Computer Systems, vol. 89, pp. 421-432, 2018.

سامانه‌های توزیع شده، سال هفتم، شماره ۱، شماره پیاپی ۱۳، صفحه ۳۳ تا ۵۰، سال ۱۴۰۳

How to cite: M. Abedi, S. E. Dashti, **Improving Task Offloading Using Type-2 Fuzzy Logic and Proximal Policy Reinforcement Learning to Enhance IoT Application Performance**, Journal of Distributed Computing and Systems (JDCS), Vol 7, Issue 1, Pages 33 - 50, 2024.

Improving Task Offloading Using Type-2 Fuzzy Logic and Proximal Policy Reinforcement Learning to Enhance IoT Application Performance

Mostafa Abedi[1], Seyed Ebrahim Dashti[2]

[1]M.Sc. Student, Faculty of Electrical and Computer Engineering, Shiraz Branch, Islamic Azad University
abediiii.mostafaaaa@gmail.com

[2]Faculty of Electrical and Computer Engineering, Jahrom Branch, Islamic Azad University (Corresponding Author)
Sayed.Dashty@gmail.com

Abstract

This paper presents a novel task offloading approach designed to improve the efficiency of IoT applications within a three-layer architecture in fog and cloud computing environments. The IoT layer consists of smart devices that generate numerous tasks, each with diverse characteristics such as size, computational requirements, communication needs, and time constraints. Due to the storage and computational limitations of IoT devices, it is necessary to offload these tasks to different layers to ensure efficient processing and meet Quality of Service (QoS) objectives.To address this challenge, a Type-2 Fuzzy Logic-based task scheduler is employed to make intelligent task migration decisions. This scheduler selects the most suitable processing layer based on the task

[21] K. Zhao et al., "Blockchain-Based Access Control for IoT," IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3632-3641, 2019.

[22] C. Dwork, "Differential Privacy: A Survey of Results," Lecture Notes in Computer Science, 2008.

[23] S. Pearson and Y. Shen, "Privacy Management in IoT Architectures," IEEE Cloud Computing, vol. 3, no. 3, pp. 34-42, 2016.

مصطفی عابدی دانشجوی کارشناسی ارشد نرم افزار در دانشکده برق و کامپیوتر، دانشگاه آزاد اسلامی، واحد شیراز، ایران است. وی مدرک کارشناسی خود را در رشته مهندسی کامپیوتر از دانشگاه آزاد اسلامی واحد شیراز دریافت کرده است. علاقه مندی های پژوهشی وی شامل الگوریتم های یادگیری ماشین، اینترنت اشیاء، داده کاوی، هوش مصنوعی، مدل سازی و راستی آزمایی رسمی، و تحلیل قابلیت اطمینان می باشد. نشانه رایانامه ایشان عباتند از :

abediiii.mostafaaaa@gmail.com

سیدابراهیم دشتی، دانشیار دانشگاه آزاد اسلامی استان فارس، ایران است. وی مدرک دکتری خود را در رشته مهندسی کامپیوتر از دانشگاه علوم و تحقیقات تهران دریافت کرده است. و دارای چندین جلد کتاب و دهها مقاله می باشند. علاقه مندی های پژوهشی وی شامل رایانش ابری، اینترنت اشیاء، رایانش فراگیر و محاسبات نرم می باشد. نشانه رایانامه ایشان عباتند از :

Sayed.Dashty@gmail.com

روش ارجاع: م.عابدی، س.ا.دشتی، بهبود انتقال وظایف با استفاده از منطق فازی نوع ۲ و الگوریتم یادگیری تقویتی با سیاست مجاور برای افزایش عملکرد برنامه های اینترنت اشیاء، دوفصلنامه محاسبات و

characteristics. Additionally, this paper leverages Proximal Policy Optimization (PPO) deep reinforcement learning to maintain load balancing among cooperative nodes in the fog layer through optimal task migration. Experimental results demonstrate that the proposed scheme outperforms existing advanced methods in terms of reducing latency, energy consumption, network utilization, throughput, and task migration rate.