مقاله پژوهشی

بهبود بارگذاری داده‌ها با در نظر گرفتن مقدار مصرف انرژی و تازگی اطلاعات در شبکه اینترنت اشیای صنعتی با کمک الگوریتم ژنتیک تقویتی

سید ابراهیم دشتی، فاطمه مؤیدی و عادل سالمی

چکیده: با افزایش روزافزون کاربرد اینترنت اشیا در زندگی روزمره و مخصوصاً صنعت، بهبود کارایی و زمان تأخیر با کمک بارگذاری داده‌ها یکی از اهداف این مسائل شده است. کنترل این عوامل باعث بهبود مصرف انرژی و استفاده طولانی‌تر از باتری اشیا خواهد شد. در این مقاله روشی برای بهبود پردازش داده‌های حسگرها و محاسبات لبه و ابر در سیستم‌های اینترنت اشیای صنعتی معرفی گردیده و معماری مطابق با دنیای واقعی در نظر گرفته شده است. در این معماری از سرورهای لبه با قابلیت‌های محاسباتی در لبه شبکه به ویژه در ایستگاه‌های پایه استفاده می‌شود. درخواست‌های حساس به تأخیر می‌توانند از طریق کانال‌های بی‌سیم به سرورهای لبه نزدیک منتقل شوند؛ در نتیجه ترافیک در شبکه مرکزی و تأخیر انتقال داده کاربر را به ویژه برای برنامه‌های صنعتی با حجم داده زیاد کاهش دهد. هدف در اینترنت اشیای صنعتی، مدیریت منابع شبکه، انتقال محاسبات و کمینه‌سازی مصرف انرژی در دستگاه‌های اینترنت اشیا با تضمین تازگی داده‌های حسگر است. محیط شبکه و کارهای ورودی متغیر با زمان هستند. در این مقاله محیط مسئله و محدودیت‌های آن با فرمول بیان گردیده و این مسئله با استفاده از الگوریتم ژنتیک و یادگیری تقویتی پیشنهادی حل شده است. راه حل پیشنهادی سبب بهبود محیط پویای مسئله برای بارگذاری داده‌ها و کارها با در نظر گرفتن انرژی و انتقال محاسبات و داده‌ها با درنظرگیری تازگی آنها شده است. نتایج نشان‌دهنده بهبود متوسط ۴۰ درصدی نسبت به روش‌های قبلی می‌باشد.

***کلیدواژه:* بارگذاری، اینترنت اشیای صنعتی، الگوریتم ژنتیک، یادگیری تقویتی.**

۱- مقدمه

اینترنت اشیای صنعتی (IIoT)[1] یک ترکیب پیوسته از حسگرها و کنترل‌کننده‌های مختلف با قابلیت‌های دریافت و کنترل و نیز ارتباطات سیار، تجزیه و تحلیل هوشمند و سایر فناوری‌ها است که به طور مداوم در تمام لینک‌های فرایند تولید صنعتی به کار می‌رود و از طریق افزایش بهره‌وری تولید، بهبود کیفیت محصول و کاهش هزینه و مصرف منابع مواد، اثرگذاری قابل توجهی دارد. علاوه بر این به منظور ارائه خدمات بهتر، تعداد دستگاه‌های اینترنت اشیای صنعتی پشتیبانی‌کننده با سرعتی شگفت‌انگیز در حال افزایش است که فرصت‌های توسعه را برای ارائه‌دهندگان خدمات اینترنت به ارمغان می‌آورد. اینترنت اشیای صنعتی به عنوان یک الگوی امیدوارکننده و فناوری نوظهور برای تحقق صنعت ۴ و تولید هوشمند در نظر گرفته شده است [۱].

در اینترنت اشیای صنعتی، مجموعه‌ای گسترده از دستگاه‌های هوشمند به طور پیوسته، حجم بزرگی از داده‌های وضعیت یا رویداد را تولید می‌کنند. سیستم باید داده‌ها را پردازش و وضعیت را تجزیه و تحلیل کرده و به‌طور مناسب واکنش نشان دهد. برنامه‌های اینترنت اشیای صنعتی نیاز به تأخیر کمی دارند تا تحلیل آماری داده‌های زمان واقعی به‌موقع انجام شود. سال‌های اخیر شاهد پیشرفت‌های قابل توجهی در فناوری هوش مصنوعی مانند یادگیری ماشینی و یادگیری عمیق بوده است که عملکرد انسان‌ها را در زمینه‌هایی مانند تشخیص تصویر، پردازش زبان طبیعی و تشخیص ناهنجاری بهبود می‌بخشد [۲].

دستگاه‌های اینترنت اشیای صنعتی، منابع محدودی برای محاسبات و پردازش دارند و هزینه خرید تجهیزات با قدرت پردازش بالا بسیار گران است؛ در نتیجه این دستگاه‌ها تأخیر در اجرای محاسبات دارند و زمانی که کارهای ورودی که نیاز به پردازش دارند به صورت قابل توجهی افزایش می‌یابد، مقدار مصرف انرژی آنها به صورت نمایی افزایش می‌یابد [۱].

دستگاه‌های اینترنت اشیا (IoT)[2] و سرور لبه[3] به صورت دوره‌ای و برحسب اولویت‌های مشخص، درخواست‌های خود را ارسال می‌کنند. در این بین، قدرت محاسباتی (به صورت محلی در دستگاه‌های IoT و سراسری در سرور لبه)، میزان اتلاف زمان در بافر سرور لبه (صف دریافت درخواست‌ها) و همچنین میزان انرژی مصرف‌شده توسط دستگاه‌های IoT به صورت دقیق توسط سیستم نظارت مورد بررسی قرار گرفته است. یک مدل جدید با در نظر داشتن انرژی دستگاه‌های IoT و میزان نرخ محاسباتی در ماشین‌های مجازی (و همچنین دستگاه‌های IoT به صورت محلی) ارائه شده است. هدف، کاهش میانگین انرژی مصرفی توسط دستگاه‌هاست که به صورت مستقیم با دیگر مؤلفه‌ها همانند زمان انتقال و حداکثر قدرت پردازشی سرور لبه در ارتباط می‌باشد [۲].

در IIoT، هدف از مدیریت منابع شبکه و انتقال محاسبات، کمینه‌سازی مصرف انرژی در دستگاه‌های IoT با تضمین تازگی داده‌های دریافت‌شده

این مقاله در تاریخ ۱۱ شهریور ماه ۱۴۰۲ دریافت و در تاریخ ۲۹ بهمن ماه ۱۴۰۲ بازنگری شد.

سید ابراهیم دشتی (نویسنده مسئول)، دانشکده برق و کامپیوتر، واحد جهرم، دانشگاه آزاد اسلامی، جهرم، ایران، (email: sayed.dashty@gmail.com).

فاطمه مؤیدی، گروه مهندسی کامپیوتر، مجتمع آموزش عالی لارستان، لار، ایران، (email: moayedi@lar.ac.ir).

عادل سالمی، دانشکده برق و کامپیوتر، واحد شیراز، دانشگاه آزاد اسلامی، شیراز، ایران، (email: adelsalemi.sadaf@gmail.com).

1. Industrial Internet of Things
2. Internet of Things
3. Edge

از حسگر است. محیط شبکه و وظیفه ورودی، متغیر با زمان هستند.

در ادامه برخی از الگوریتم‌های پیشین که در این زمینه مورد استفاده قرار گرفته، به اختصار بررسی شده است. گرادیان سیاست قطعی عمیق (DDPG)[1] یک الگوریتم است که در سیستم‌های کنترل برای بهبود عملکرد کنترل سیستم مورد استفاده قرار می‌گیرد. مبتنی بر یادگیری تقویت عمیق است و سیاست کنترل بهینه را از فرایند تصمیم مارکوف و الگوریتم بازیگر- منتقد دریافت می‌کند. در این کاربردها، DDPG پایداری، همگرایی و توانایی ضد تداخل بهتری را در مقایسه با روش‌های کنترل سنتی نشان داده است. علاوه بر این، استفاده از مدل‌های کوانتومی در DDPG برای کاهش پیچیدگی محاسباتی و از دست دادن آموزش مورد بررسی قرار گرفته که منجر به پاداش‌های بالاتر و تلفات آموزش کمتر می‌شود. به طور کلی، DDPG یک رویکرد برای دستیابی به کنترل با عملکرد بالا در محیط‌های پیچیده و پویاست [۳].

سافت بازیگر- منتقد (SAC)[2] یک الگوریتم یادگیری تقویتی است که در حوزه‌های مختلف کاربرد داشته است. SAC شامل یک شبکه منتقد است که سریع‌تر از شبکه بازیگر یاد می‌گیرد و سازگاری نهایی بین این دو را تضمین می‌کند. در زمینه یادگیری تقویت چندعاملی (MARL)، SAC با الگوریتم‌های دیگر ترکیب شده تا چالش‌هایی مانند بی‌ثباتی حالت را رفع کند. مثلاً الگوریتم SAC S4AC[3] را با الگوریتم SSIG[4] ترکیب می‌کند تا حالت کامل عوامل همکار در یک سیستم چندعاملی را استنباط کند. به طور کلی SAC و انواع آن توانایی یادگیری و سازگاری قوی را در سناریوهای مختلف از خود نشان می‌دهند [۴].

الگوریتم گرادیان سیاست قطعی عمیق با تأخیر دوقلو (TD3)[5]، یک الگوریتم یادگیری تقویتی است که به مسئله برآورد بیش از حد در الگوریتم‌های یادگیری تقویتی عمیق مانند DQN و DDPG می‌پردازد. TD3 حداقل مقدار بین یک جفت منتقد را برای محدودکردن بیش از حد ارزیابی و بهبود عملکرد در نظر می‌گیرد. در حوزه‌های مختلف از جمله مدیریت انرژی سیستم‌های ذخیره انرژی ترکیبی برای وسایل نقلیه الکتریکی، کنترل محدودیت سرعت متغیر سطح خط در سیستم‌های بزرگراه اتوماتیک وسایل نقلیه متصل و برنامه‌ریزی مسیر سه‌بعدی برای وسایل نقلیه هوایی بدون سرنشین استفاده شده است. الگوریتم TD3 عملکرد بهتری از نظر اقتصاد، کاهش زمان آموزش، ایمنی ترافیک و راندمان ترافیک نسبت به سایر الگوریتم‌ها از خود نشان داده است [۵].

کنترل انرژی و تخلیه محاسبات موضوعات مهمی در زمینه محاسبات لبه موبایل (MEC) هستند. MEC پارادایمی است که منابع محاسباتی را در لبه شبکه در نزدیکی دستگاه‌های ترمینال استقرار می‌دهد و امکان عملکرد محاسبات بهتر و طول عمر باتری را در عین اطمینان از کیفیت خدمات (QoS)[6] فراهم می‌کند. چندین مقاله استراتژی‌ها و الگوریتم‌هایی برای بهینه‌سازی تخلیه وظایف، تخصیص کانال، کنترل توان و زمان‌بندی منابع در سیستم‌های MEC پیشنهاد می‌کنند. هدف از این رویکردها حداقل‌کردن تأخیر سیستم و مصرف انرژی و بهبود کیفیت خدمات برای تلفن همراه است. روش‌های پیشنهادی شامل تکنیک‌هایی مثل یادگیری تقویتی چندعاملی، شبکه‌های عصبی و الگوریتم‌های توزیع‌شده است. نتایج شبیه‌سازی نشان می‌دهند که این رویکردها از لحاظ تأخیر، مصرف انرژی و نرخ دورانداخته بهتر عمل می‌کنند. به طور کلی، این مطالعات بینش و راه حل‌های ارزشمندی را برای تخلیه محاسبات کارآمد انرژی و مدیریت منابع در شبکه‌های مبتنی بر MEC ارائه می‌دهند [۶].

در این مقاله با استفاده از الگوریتم ژنتیک، مسئله بهینه‌سازی پویا در کنترل انرژی و انتقال محاسبات فرموله شده و یک الگوریتم هوشمند کنترل انرژی و انتقال محاسبات طراحی گردیده است. در بخش ۲ مروری بر آثار مرتبط با این مقاله صورت گرفته و در بخش ۳ مدل سیستم برای یک سیستم IIoT ارائه شده و تحلیل‌های کمی انجام می‌شود. در بخش ۴ مسئله بهینه‌سازی پویا فرموله شده و الگوریتم‌ها توضیح داده می‌شوند. در بخش ۵ عملکرد رویکرد پیشنهادی ارزیابی گردیده و نهایتاً در بخش ۶ قسمت پایانی مقاله بیان شده است.

۲- مرور ادبیات و سوابق مربوطه

مسئله بهینه‌سازی منابع و شبکه و انتقال محاسبات با آگاهی از طول عمر اطلاعات در یک سناریوی اینترنت اشیای صنعتی در محیط پویا با چندین دستگاه اینترنت اشیا و چندین سرور لبه بررسی شده است. با توجه به دستگاه‌های اینترنت اشیا که منابع شبکه، محاسبات و انرژی محدود دارند، مسئله در محیط پویا برای کمینه‌سازی مصرف میانگین انرژی محاسباتی به منظور طولانی‌شدن طول عمر شبکه با محدودیت میانگین طول عمر اطلاعات مدل می‌شود. استفاده از یادگیری تقویتی عمیق با استفاده از فرایند تصمیم‌گیری مارکوف، ایجاد الگوریتم کنترل انرژی- انتقال هوشمند، استفاده از طرح تقسیم چندگانه فرکانس برای ارتباط بین دستگاه‌ها و سرور لبه و استفاده از فرایند پواسون استفاده شده است. این معماری از سرورهای لبه با قابلیت‌های محاسباتی در لبه شبکه، به ویژه در ایستگاه‌های پایه استفاده می‌کند. درخواست‌های حساس به تأخیر می‌توانند از طریق کانال‌های بی‌سیم به سرورهای لبه نزدیک منتقل شوند و در نتیجه منجر به عدم تداخل در دستگاه‌های شبکه گردد و ترافیک در شبکه مرکزی و تأخیر انتقال داده کاربر را به ویژه برای برنامه‌های صنعتی با حجم داده زیاد، کاهش دهد. در محاسبات لبه، شبکه‌سازی و محاسبات ترکیب می‌شوند و تصمیم بر این که آیا وظایف به یک سرور لبه منتقل شوند یا نه، باید با دقت بر اساس نیازهای سرویس‌های مبتنی بر کیفیت، وضعیت شبکه و تأمین انرژی اتخاذ شود. این روش مناسب برای محیط شبکه اینترنت اشیای صنعتی پویا است؛ اما عدم پیش‌بینی محدودیت منابع سرورهای لبه از معایب این روش می‌باشد [۲].

برای بررسی انتقال محاسبات وظایف پیچیده و تخصیص منابع پویا در اینترنت اشیای صنعتی با کمک محاسبات لبه با توجه به منابع محدود و متنوع سرورهای لبه، انتقال محاسبات و تخصیص منابع ترکیب شده‌اند. هر دو مصرف انرژی، مصرف زمان دستگاه‌های صنعتی و نیز استفاده از منابع و توازن بار سرورهای لبه به عنوان اهداف بهینه‌سازی در نظر گرفته می‌شوند. به طور خاص، وظایف در اینترنت اشیای صنعتی به‌عنوان وظایف پیچیده مدل‌سازی شده‌اند که شامل برنامه‌های گردش کار و برنامه‌های هم‌زمان می‌شوند. محدودیت زمانی برنامه‌های گردش کار در نظر گرفته شده است. به همین ترتیب، یک روش هوشمندانه و همکاری محاسباتی بین انتها، لبه و ابر بر اساس الگوریتم چندمعیار ژنتیک برای حل این حالت ارائه شده است. به دلیل اندازه بزرگ دستگاه‌های اینترنت اشیای صنعتی و منابع محدود و متفاوت سرورهای لبه، استفاده بهینه از منابع یک چالش را ایجاد می‌کند. برای حل این چالش، رویکرد جدیدی مبتنی بر شبکه‌های هوشمند و شبکه‌های مجازی خصوصی ارائه شده است. این رویکرد

1. Deep Deterministic Policy Gradient
2. Soft Actor-Critic
3. State Super Sampling Soft Actor-Critic
4. Super Sampling Info-GaN
5. Twin Delayed Deep Deterministic Policy Gradient
6. Quality of Services

می‌تواند محاسبات لبه را برای دستگاه‌های اینترنت اشیای صنعتی متناسب کند و قابلیت تغییرپذیری و استفاده بهینه از منابع را فراهم نماید. انتقال وظایف محاسباتی آنها به سرورهای لبه می‌تواند توانایی آنها را در اجرای برنامه‌هایی که به منابع غنی نیاز دارند به طور قابل توجهی افزایش دهد تا زمان اجرا و مصرف کلی انرژی را به طور قابل توجهی کاهش دهد. در نتیجه، استفاده از محاسبات لبه و شبکه‌های مبتنی بر نرم‌افزار در اینترنت اشیای صنعتی می‌تواند راه حلی کارآمد برای بهبود عملکرد و کارایی این دستگاه‌ها باشد و به ارائه‌دهندگان خدمات اینترنت این امکان را می‌دهد که از فرصت‌های توسعه بزرگی که با توجه به افزایش ترافیک داده‌ها ایجاد می‌شود، بهره ببرند. این روش مناسب برای تسک‌های موازی نیست؛ ولی مناسب برای شبکه‌هایی با تعداد زیاد دستگاه می‌باشد [۱].

برای بررسی مسئله تسهیل امکان یادگیری لبه به صورت کارا و انعطاف‌پذیر در شبکه‌های اینترنت اشیای صنعتی، یک چارچوب یادگیری نوآورانه پیشنهاد داده شده تا مشکل همگنی سیستم را حل کند. الگوریتم پیشنهادشده، چندین خروجی در مدل محلی استفاده می‌کند تا بتواند با رعایت محدودیت تأخیر، آموزش مدل کامل را انجام دهد. در این حالت، دستگاه‌ها قادرند بهترین خروجی را انتخاب کرده و بخش خاصی از مدل را بر اساس نیازهای خود آموزش دهند. با این حال، زیرمجموعه‌های زودهنگام ممکن است باعث کاهش عملکرد شود؛ بنابراین از تکنولوژی فیلتر دانش برای حل این مشکل استفاده می‌کنیم. علاوه بر این، یک الگوریتم مشترک برای انتخاب خروجی و تخصیص پهنای باند بر اساس رویکرد حریصانه پیشنهاد داده شده تا تعداد مورد انتظار خروجی‌ها در هر دوره ارتباطی را بیشینه کنیم. ورود داده‌های زیاد در لبه شبکه، چالش قابل توجه دیگری را برای سرورهای ابری مرکزی ایجاد می‌کند. الگوی سنتی محاسبات ابری برای برنامه‌هایی با حساسیت به تأخیر مناسب نیست. برای مقابله با این مسائل، محاسبات لبه همراه پیشنهاد شد که منابع محاسباتی را در نزدیکی منبع داده‌ها استقرار می‌دهد. در این مفهوم، دستگاه‌های اینترنت اشیا می‌توانند داده‌ها و وظایف محاسباتی را به سرور لبه انتقال دهند که منجر به کاهش تأخیر، مصرف انرژی پایین‌تر و کیفیت خدمات بالاتر می‌شود.

از چالش‌های اصلی دیگر در شبکه‌های اینترنت اشیای صنعتی، تأخیر پردازش قابل توجهی است که به دلیل منابع محدود مثل قدرت محاسباتی و پهنای باند به آن است. برای حل این مسئله، یک آستانه تأخیر به کار گرفته می‌شود تا آموزش را هماهنگ کند و انتظارات زمان انتظار کم را از بین ببرد. چالش دیگر تنوع سیستم است که در آن دستگاه‌های مختلف اینترنت اشیای صنعتی دارای قدرت محاسباتی و وضعیت کانال متفاوت هستند. در چنین مواردی، دستگاه‌هایی با منابع کافی می‌توانند آموزش محلی و بارگذاری[1] را در زمان تأخیر مورد نیاز به پایان برسانند؛ در حالی که دستگاه‌هایی با منابع محدود این کار را انجام نمی‌دهند. هرچند انتخاب یک مدل سبک برای تمامی دستگاه‌ها ممکن است مشخص به نظر برسد، اما می‌تواند منجر به تخریب شدید عملکرد و نبود انعطاف‌پذیری با بهبود وضعیت کانال شود. این چالش‌ها نیاز به یک چارچوب نوآورانه و انعطاف‌پذیر به نام یادگیری لبه فدرال را مشخص می‌کند. این روش به دستگاه‌ها امکان آموزش وفقی مدل‌های مختلف با مقیاس‌های مختلف بر اساس منابع موجود را می‌دهد؛ در حالی که سرور می‌تواند وزن‌ها را از دستگاه‌های مختلف جمع‌آوری کند. با الهام از مکانیزم چندخروجی در یادگیری عمیق، این روش مدل‌های آموزش‌داده را به عنوان یک انسجام دستگاه لبه با ورودی‌های متفاوت و خروجی‌های مناسب کنترل می‌کند. از طریق نمونه‌برداری مناسب و انتخاب مدل مناسب به دستگاه‌ها، روش ما می‌تواند تأخیر پردازش و مصرف انرژی را بهبود دهد و نیز حفظ حریم خصوصی را تضمین کند. این روش نامناسب برای محیط‌هایی با متغیر زمانی است و در کاهش مصرف انرژی ضعیف عمل می‌کند [۷].

در خصوص انتقال وظایف، فشرده‌سازی داده و حفاظت امنیت در اینترنت صنعتی به طور دقیق، یک معماری دولایه‌ای شامل لایه دستگاه ترمینال و لایه سرور لبه ایجاد می‌شود. فشرده‌سازی داده را با محاسبات لبه ترکیب کرده و با ترکیب تصمیمات انتقال وظایف و تخصیص منابع، نرخ فشرده‌سازی بهینه را تعیین می‌کنیم. هدف از این ترکیب، غلبه بر محدودیت ظرفیت پیوند بی‌سیم بین دستگاه پایانی و سرور لبه و کاهش تأخیر انتقال وظیفه است. علاوه بر این برای حفظ امنیت وظایف برای هر وظیفه سطوح امنیتی متفاوت تعیین می‌کنیم که با کاهش هزینه ریسک امنیتی هر وظیفه، حفاظت امنیت داده را بهبود می‌بخشد و محیط کارخانه به طور کلی ایمن‌تر می‌شود. الگوریتم استفاده‌شده برای حل مسئله انتقال وظایف طراحی شده است. برای جبران کمبود محاسبات ابری و محاسبات لبه، یک الگوی محاسباتی جدید، وظایف محاسباتی را از دستگاه‌های ترمینال به سرورهای لبه منتقل می‌کند. وظیفه موجود در ترمینال، خدمات محاسباتی را از نودهای لبه در نزدیکی خود دریافت می‌کند تا سرعت پردازش وظایف حساس به تأخیر را افزایش دهد. با مزیت‌های پهنای باند بالا و پاسخ تأخیر پایین، سرورهای لبه برای وظایف با زمان واقعی بالا و منابع محاسباتی محلی با تأخیر کم و قابلیت اعتماد بالاتری را برای تجهیزات میدانی کارخانه فراهم می‌کنند. نسبت به محاسبات ابری، محاسبات لبه منابع محدودی دارد و بنابراین تخصیص منابع محاسباتی بهینه و انتقال مؤثر مسئله‌های کلیدی برای سیستم‌های محاسباتی لبه برای کاهش مصرف انرژی و تأخیر است. با توسعه داده‌های صنعتی و افزایش تأخیر در انتقال داده بین کاربران و سرورهای لبه به دلیل ارتباطات بی‌سیم محدود، کیفیت خدمات برای کاربران تحت تأثیر قرار می‌گیرد. محققانی بسیار برای رفع این مشکل از تکنیک‌های فشرده‌سازی برای کاهش تأخیر استفاده کرده‌اند. با این حال، فشرده‌سازی داده نیاز به منابع محاسباتی اضافی برای فشرده‌سازی و بازسازی داده دارد. با اعمال فشرده‌سازی داده به فرایند انتقال محاسبات لبه، طراحی مشترک واگذاری فشرده‌سازی داده، تصمیمات انتقال و تخصیص منابع نه‌تنها در فضای ذخیره‌سازی داده صرفه‌جویی می‌کند، بلکه به طور قابل توجهی تأخیر را کاهش می‌دهد و به عنوان راهی برای کاهش فشار انتقال در شبکه‌های اصلی صنعتی عمل می‌کند. این روش مناسب برای شبکه‌هایی با تعداد زیاد دستگاه است؛ اما مصرف انرژی بالا و نامناسب‌بودن برای تسک‌های حساس به زمان از ضعف‌های اساسی در این روش می‌باشد [۸].

با هدف کاهش تأخیر در پردازش وظایف در شبکه‌های اینترنت صنعتی اشیای چندگامه، الگوریتمی توزیع‌شده بر پایه ADMM[2] برای بارگذاری وظایف و تخصیص بهینه منابع ارتباطی و محاسباتی ارائه شده است. مدل سیستم شامل دستگاه‌های صنعتی، سوئیچ‌های SDN[3] و نقاط دسترسی است. دستگاه‌ها می‌توانند وظایف محاسباتی شدید خود را از طریق ارتباطات بی‌سیم چندگامه به سرورهای لبه متصل به سوئیچ‌های SDN بارگذاری کنند. یک مدل ارتباطی برای در نظر گرفتن تأثیر لینک‌های نامطمئن بر تأخیر انتقال ساخته شده است. یک مسئله بهینه‌سازی

1. Offloading

2. Administrative Data Maturity Model

3. Software-Defined Networking

غیرخطی عدد صحیح مختلط (MINLP)[1] برای تصمیم‌گیری مشترک در مورد بارگذاری وظایف، تخصیص پهنای باند و منابع محاسباتی با هدف کمینه‌کردن تأخیر پردازش وظایف فرمول‌بندی شده است. با معرفی متغیرهای کمکی و استفاده از تکنیک‌های خطی‌سازی، مسئله به یک مسئله بهینه‌سازی محدب تبدیل گردیده و سپس الگوریتمی مبتنی بر ADMM طراحی شده تا به صورت توزیع‌شده مسئله را حل کند. در این الگوریتم، کپی‌هایی محلی از متغیرها نگهداری و به صورت تکراری به‌روزرسانی می‌شوند. نتایج شبیه‌سازی نشان می‌دهند الگوریتم پیشنهادی می‌تواند تأخیر متوسط پردازش وظایف را کاهش دهد و به عملکرد نزدیک به بهینه الگوریتم‌های متمرکز با پیچیدگی کمتر دست یابد. در مجموع، الگوریتم پیشنهادی راه‌حلی مؤثر برای بارگذاری وظایف چندگامه در IIoT با تأخیر کمینه است [۹].

مرجع [۱۰] الگوریتمی برای بارگذاری وظایف در سیستم محاسبات لبه- ابری همکار بر پایه بهینه‌سازی لیاپانوف ارائه می‌دهد تا مصرف انرژی کاهش یابد. معماری لبه- ابر همکار پیشنهاد می‌شود که شامل سه لایه دستگاه، لبه- ابر و مدیریت است. مدل‌های صف‌بندی پویا برای سرورهای لبه و ابر تعریف می‌شود تا وضعیت سیستم نشان داده شود. مسئله به صورت بهینه‌سازی مصرف انرژی با رعایت محدودیت‌های پایداری برای صف سرورها تبدیل می‌شود. مسئله به زیرمسائل تخصیص منابع محاسباتی و بهینه‌سازی انرژی تفکیک می‌شود و راه‌حل‌های بهینه ارائه می‌گردد. در مجموع الگوریتم پیشنهادی با کاهش مصرف انرژی و ازدحام شبکه، پایداری سیستم لبه- ابر را حفظ می‌کند. نتایج شبیه‌سازی برتری الگوریتم نسبت به روش‌های تخصیص تصادفی و یکنواخت را نشان می‌دهد. مجموعاً نوآوری اصلی مقاله ارائه الگوریتمی برای بارگذاری وظایف در محاسبات لبه- ابر است که بهره‌وری منابع و مصرف انرژی پایین را به دست می‌آورد. یک الگوریتم تخصیص بهینه منابع محاسباتی و توان انتقال برای سیستم همکاری ابر- لبه ارائه می‌دهد که مصرف کل انرژی سیستم را کمینه و از ازدحام شبکه جلوگیری می‌کند [۱۰].

با توجه به کاربرد روزافزون محاسبات مه، لزوم ارائه راه‌حل‌هایی جهت افزایش کارایی آنها به شدت احساس می‌شود. بر اساس این مسئله که تعداد دستگاه‌های لبه‌ای زیاد است، باید سازوکاری برای انتخاب این وظایف و تخلیه آنها به ابر وجود داشته باشد. مسئله مورد نظر برای تصمیم‌گیری این است که از بین آن دستگاه لبه‌های موجود برای تخلیه کدام یک از آنها انتخاب و سپس تخلیه گردد که این مسئله در زمره مسائل غیرچندجمله‌ای سخت قرار گرفته و با استفاده از الگوریتم‌های قطعی به سادگی و در زمان چندجمله‌ای نمی‌توان راه‌حلی مناسب و کارآمد برای آن یافت نمود. برای حل این مسئله از الگوریتم بهینه‌سازی ازدحام ذرات استفاده شده است. روش پیشنهادی با تعریف تابع هدف مناسب برحسب تخلیه مناسب بار و توزیع عادلانه وظایف بر روی منابع محاسباتی کارایی را نسبت به روش‌های مشابه بهبود داده است. در مقایسه با روش‌های دیگر مانند روش بدون بارگیری، بارگیری کامل به روش ابر و بارگذاری کامل به روش ابر، آزمایش‌ها و شبیه‌سازی‌های گسترده نشان داده‌اند که روش پیشنهادی مؤثر است و می‌تواند استراتژی بارگذاری بهینه را برای کاربران سیار فراهم کند. برای ارزیابی این روش از داده‌های واقعی پلنت‌لب استفاده شده و نتایج بیانگر این است که روش پیشنهادی مصرف انرژی را بین ۳ تا ۱۰ درصد و زمان اجرای کل را نیز بین ۵ تا ۸ درصد در مقایسه با روش‌های دیگر کاهش داده است [۱۱].

رایانش ابری اخیراً به موضوعی بسیار محبوب تبدیل شده است. امروزه با افزایش تقاضا برای اتوماسیون فرایند در ابر، بررسی استراتژی‌های زمان‌بندی گردش کار ابری به یک موضوع مهم تبدیل شده است. اکثر الگوریتم‌های زمان‌بندی گردش کار موجود، فقط منابع محاسباتی را در نظر می‌گیرند که معمولاً نمی‌توان آنها را بر اساس اندازه تقاضای گردش کار تهیه کرد یا تا زمانی که اجرای گردش کار به پایان برسد، در محیط منتشر نمی‌شود. به همین دلیل است که عملکرد این الگوریتم‌ها کاهش و زمان و هزینه آنها افزایش می‌یابد. یک روش زمان‌بندی گردش کار جدید را بر اساس الگوریتم بهینه‌سازی کلونی مورچه‌ها ارائه می‌کنیم تا با در نظر گرفتن مشکلات فوق در محیط پویا، این سربار زمان‌بندی را کاهش دهیم. علاوه بر این، ما این مشکلات و انواع ماشین‌های مجازی را در حین اجرا به صورت پویا بر اساس Amazon EC۲ در نظر می‌گیریم. همچنین در مقایسه با روش‌های پیشرفته در روش زمان‌بندی مقیاس بزرگ، مجموعه داده‌های ما بر اساس برنامه‌های جریان کار واقعی با حداکثر ۱۰۰ گره است. نتایج نشان می‌دهند عملکرد الگوریتم ما به طور قابل توجهی بهتر از روش جستجوی تصادفی تطبیقی حریصانه و مقیاس‌پذیر برای افزایش گره‌های گردش کار است [۱۲].

برای یک استراتژی انتقال وظایف در جهت کاهش مصرف انرژی سیستم همکاری ابر مدل صف سرور MEC[2] و مدل صف سرور MCC[3] تشکیل گردیده و آستانه صف تعیین می‌شود. تخصیص منابع محاسباتی و نرخ انتقال به یک مسئله بهینه‌سازی محدود تبدیل می‌شود. با استفاده از نامساوی مارکوف، مسئله به سه مسئله زیر تقسیم می‌گردد و یکی‌یکی حل می‌شود. این الگوریتم‌ها می‌تواند تضمین کند که سیستم سرور MEC و سیستم سرور MCC در حالت ترافیک پایینی قرار گیرند و مصرف کلی انرژی سیستم را کاهش دهند [۱۳].

مشکل شبکه‌های کارآمد انرژی را در خدمات ابری با مراکز داده توزیع‌شده جغرافیایی برای شبکه‌های صنعتی اینترنت اشیا به‌ویژه برای شبکه‌های چندرسانه‌ای مطالعه شده است. این امر به طور قابل توجهی توسط تقاضاهای پویا از انتها به انتها و بازده انرژی لینک نامتعادل، استفاده از لینک نامتعادل و متغیر با زمان و محدودیت‌های پهنای باند و تأخیر برای الزامات خدمات به چالش کشیده شده است. برای حل این مسائل، یک مدل بهینه‌سازی چندمحدودیتی برای بهینه‌سازی کارایی انرژی در خدمات رایانش ابری پیشنهاد شده که در آن مراکز داده از نظر جغرافیایی توزیع شده‌اند و توسط شبکه‌های ابری به هم متصل می‌شوند. مدل به طور مشترک بهره‌وری انرژی را در مراکز داده و شبکه‌های ابری بهینه می‌کند. یک الگوریتم اکتشافی هوشمند برای حل این مدل برای تقاضاهای درخواست پویا بین مراکز داده مختلف و بین مراکز داده و کاربران ارائه شده است. این با ترکیب الگوریتم ژنتیک طاقچه و جستجوی تصادفی عمق اول اجرا می‌شود. نتایج شبیه‌سازی برای شبکه‌های کارآمد انرژی نشان می‌دهد با بهینه‌سازی مشترک می‌توان به دستاوردهای بهتری در بهره‌وری انرژی شبکه دست یافت. بهینه‌سازی مشترک بین مراکز داده صنعتی و شبکه‌های ابری صنعتی می‌تواند صرفه‌جویی در مصرف انرژی و استفاده از پیوند برای درخواست‌های با زمان متفاوت را بهبود بخشد [۱۴].

در ادامه در جدول ۱ مروری بر کارهای قبلی به تفکیک عملکردهای آنها آورده شده است.

1. Mixed Integer Nonlinear Programming

2. Multi-Access Edge Computing

3. Mobile Cloud Computing

جدول ۱: مروری بر کارهای پیشین بر اساس معیارهای مختلف.

معیار ارزیابی	AVERAGE DELAY OF TASK PROCESSING	AVERAGE RESOURCE UTILIZATION	ACCURACY
منتشرشده در	Future Generation Computer Systems	IEEE Trans. on Network Science and Engineering	IEEE Trans. on Network Science and Engineering
تاریخ انتشار	13 Sep. 2022	8 Mar. 2022	8 Jun. 2022
موضوع	کنترل انرژی و اجرای محاسباتی با توجه به شاخص زمان انتشار اطلاعات برای اینترنت اشیای صنعتی	بارگذاری محاسبات هوشمند و تخصیص منابع در IIoT با رایانش ابری انتهای لبه با استفاده از NSGA-III	هوش محاسباتی و یادگیری عمیق برای اینترنت اشیای صنعتی (IIoT) با امکانات لبه
نوآوری	۱) استفاده از یادگیری تقویتی عمیق با استفاده از فرایند تصمیم‌گیری مارکوف، ۲) ایجاد الگوریتم کنترل انرژی و ۳) استفاده از طرح تقسیم چندگانه فرکانس (FDMA) برای ارتباط بین دستگاه‌ها و سرور لبه	۱) تخلیه هوشمند محاسباتی مشترک مبتنی بر لبه- ابر مطابق با الگوریتم ژنتیک مرتب‌سازی غیرغالب III (NSGA-III) و ۲) ترکیب تخلیه محاسبات و تخصیص منابع به طور هم‌زمان	۱) استفاده از تئوری بهینه‌سازی لیاپانوف، ۲) معماری همکاری محاسبات ابری و محاسبات لبه‌ای، ۳) الگوریتم offloading مبتنی بر اختلال و ۴) استفاده از نامساوی مارکوف
پارامتر مقاله	۱) احتمال ارسال، ۲) توان انتقال و ۳) سیاست زمان‌بندی	۱) زمان مصرفی، ۲) انرژی مصرفی، ۳) استفاده از منابع و ۴) بارهای مختلف سرورها	۱) پهنای باند و ۲) مصرف انرژی
مزیت	۱) کنترل انرژی و اجرای محاسباتی برای اینترنت اشیای صنعتی و ۲) استفاده از مفهوم «طول عمر اطلاعات» (Age of Information) در کنترل انتقال داده و بهینه‌سازی مصرف انرژی دستگاه‌های IoT	۱) بارگذاری محاسباتی برای وظایف پیچیده، ۲) تخصیص منابع پویا در IIoT با کمک MEC، ۳) ترکیب تخلیه محاسباتی و تخصیص منابع با هم و ۴) کاهش مصرف انرژی، زمان مصرف دستگاه‌های صنعتی و همچنین استفاده از منابع، تعادل بار سرورهای لبه به عنوان اهداف بهینه‌سازی دیده می‌شود.	۱) تبدیل تخصیص منابع محاسباتی و نرخ انتقال به یک مسئله بهینه‌سازی محدود و ۲) تضمین این که سیستم همکاری سمت ابر در حالت تراکم خفیف قرار دارد.
نقطه ضعف	۱) ضعف عملکرد در شرایطی که فضای عمل پیوسته نیست یا بسیار بزرگ است.	۱) در قسمت تعادل بار الگوریتم EECCT بهینه نیست و الگوریتم FCFS بهتر عمل می‌کند، ۲) نامناسب برای تسک‌های موازی و ۳) نامناسب برای داده‌های امنیتی	۱) عدم در نظر گرفتن تهدیدات امنیتی، ۲) نامناسب برای شبکه‌های پیچیده و داده‌های متنوع و ۳) نامناسب برای محیط‌های متغیر زمانی
مجموعه داده	مجموعه داده‌های واقعی از شانگهای تلکام	استفاده از داده‌های ساختگی	از مجموعه دیتاست‌های Fashion-MNIST استفاده شده که در سایت https://github.com/zalandoresearch/fashion-mnist قابل دسترسی است.
محیط شبیه‌سازی	CPU frequency of IoT device is 1 GHz while the CPU frequency of the edge server is 4.4 GHz. The wireless channel bandwidth is set to be 1 MHz.	شبیه‌سازی در سیستم با مشخصات Win10 64-bit Operating System based on JAVA with a I7-9750H 2.60GHz Intel processor	classical ResNet-18
هدف	۱) بهینه‌سازی منابع، ۲) کاهش مصرف انرژی و ۳) رعایت محدودیت‌های طول عمر اطلاعات	۱) بهینه‌سازی منابع، ۲) کاهش مصرف انرژی و ۳) کاهش زمان محاسباتی	۱) خدمات شبکه با پهنای باند بالا و تأخیر کم، ۲) کاهش مصرف انرژی و ۳) کاهش اختلال در سیستم همکاری لبه و ابر

OBJECTIVE VALUE	AVERAGE DELAY	AVERAGE BACKLOG OF MCC SERVERS AVERAGE BACKLOG OF MEC SERVERS
The J. of Supercomputing	IEEE Internet of Things J.	Cluster Computing The J. of Networks, Software Tools and Applications
26 Sep. 2022	9 Jun. 2022	3 Mar. 2022
بارگذاری وظیفه برای محاسبات لبه در اینترنت صنعتی با فشرده‌سازی داده‌های مشترک و حفاظت امنیتی	تخصیص منابع و تحلیه وظایف مشترک برای اینترنت صنعتی چندهاب	یک الگوریتم تخلیه کار برای سیستم مشارکتی لبه ابری مبتنی بر بهینه‌سازی لیاپانوف
۱) طراحی الگوریتم انتقال بر مبنای الگوریتم جریان ذرات ترکیبی تقلیدی بهبودیافته (ISAPSO)، ۲) ترکیب فشرده‌سازی داده با محاسبات لبه و ۳) ترکیب تصمیمات انتقال تسک‌ها و تخصیص منابع	یک الگوریتم مبتنی بر روش ADMM با استفاده از متغیرهای کمکی و تکنولوژی RLT طراحی‌شده	با استفاده از نابرابری مارکوف و روش بهینه‌سازی لیاپانوف، مسئله به سه زیرمسئله تجزیه شده و یکی‌یکی حل می‌گردد. الگوریتم ارائه‌شده در این مقاله می‌تواند اطمینان حاصل کند که سیستم همکاری سمت ابر در حالت تراکم خفیف قرار دارد و مصرف انرژی کلی سیستم کم است.
۱) تأخیر، ۲) مقادیر هدف در استراتژی تخلیه، ۳) مقادیر هدف در امنیت و ۴) مقادیر هدف در امنیت	۱) تأخیر در سرعت محاسبات	۱) استراتژی تخلیه وظیفه برای به حداقل رساندن مصرف انرژی سیستم همکاری سمت ابر و ۲) بهینه‌سازی صف وظایف سرورهای ابری و سرورهای لبه و تخصیص منابع سرورهای ابری و سرورهای لبه
مشکل تخلیه وظیفه را در یک سیستم محاسبات لبه چندکاربره با فشرده‌سازی داده‌ها و حفاظت از امنیت بررسی کرده و یک مدل محاسباتی، یک مدل ارتباطی، یک مدل فشرده‌سازی و یک مدل امنیتی ایجاد کرده‌اند.	۱) افزایش قدرت پردازش سرورهای لبه و ۲) کاهش زمان محاسبه	قدرت پردازشی سرورهای لبه در مقایسه با الگوریتم‌های متمرکز، ۱٪ پایین‌تر است.
۱) مدت زمان اجرای الگوریتم نسبت به سایر الگوریتم‌های قابل استفاده زیاد می‌باشد، ۲) مصرف بالای انرژی و ۳) نامناسب برای تسک‌های حساس به زمان	استفاده از داده‌های ساختگی	داده‌های واقعی
شبیه‌سازی در سیستم با مشخصات Java programming language. CPU Intel(R) Core(TM) i5-10750H CPU @ 2.30 GHz, the OS 64-bit, memory 8GB	MATLAB R2014a	استفاده از داده‌های واقعی
۱) مدل بهینه‌سازی برای انتقال تسک‌ها با فشرده‌سازی داده، ۲) حفاظت امنیت مشترک و ۳) فشرده‌سازی بهینه داده	۱) ایجاد مدل لینک برای برقراری ارتباط، ۲) ارتباط بین قابلیت اطمینان و تأخیر انتقال، ۳) کمینه کردن تأخیر پردازش وظایف و ۴) تخصیص وظایف	اشاره‌ای نشده است.
		۱) کاهش مصرف انرژی و ۲) تخصیص منابع محاسباتی و کاهش تراکم شبکه

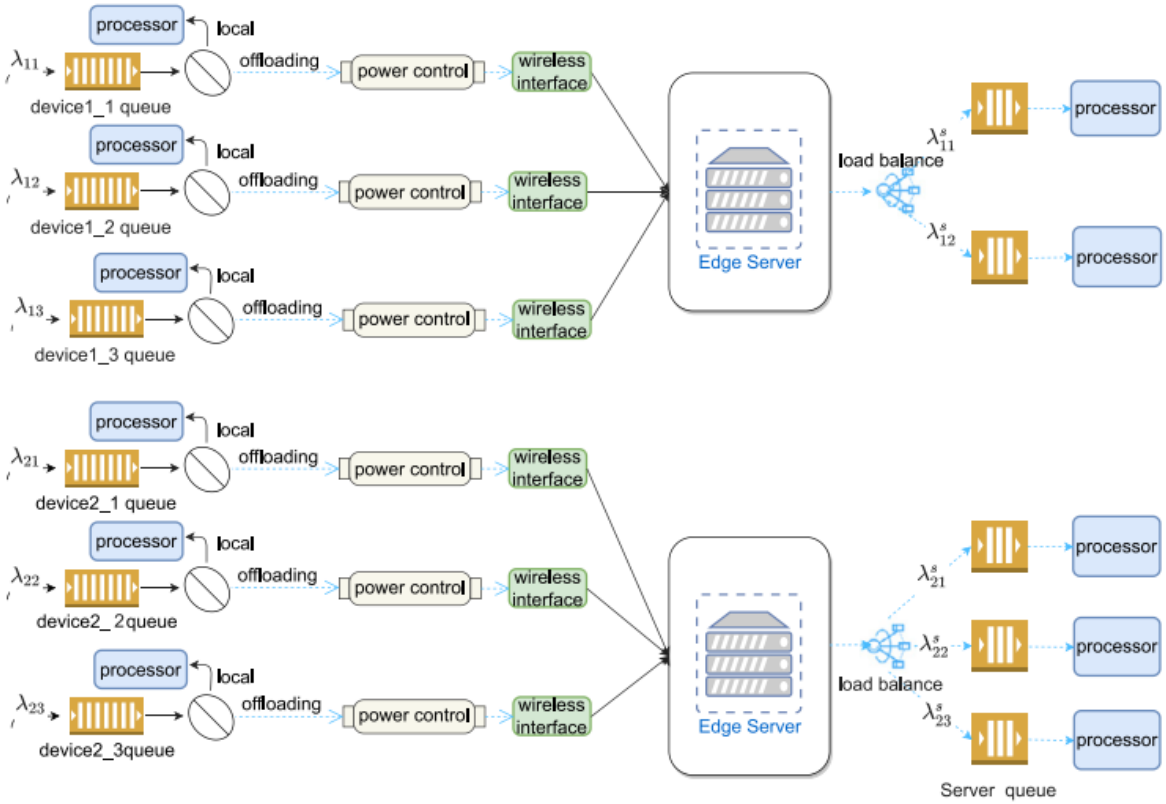
۳- مدل سیستم و فرموله‌بندی

۳-۱ سناریو

در این مقاله، یک سناریوی IIoT را مورد بررسی قرار می‌دهیم که در آن چندین دستگاه IoT و سرورهای لبه وجود دارند و در شکل ۱ نشان داده شده است. هر دستگاه IoT از طریق شبکه بی‌سیم به یکی از سرورهای لبه متصل می‌شود. فرض می‌کنیم که ارتباط بین دستگاه IoT و سرورهای لبه از طریق یک روش FDMA[1] انجام می‌گردد و کانال‌ها از پیش اختصاص داده می‌شوند؛ بنابراین هیچ تداخلی بین دستگاه‌های مختلف IoT وجود ندارد. فرض می‌کنیم هیچ منطقه‌ای بین ایستگاه‌های پایه تداخلی وجود ندارد و بنابراین مدیریت شبکه و برنامه‌ریزی وظیفه به صورت موازی قابل انجام است. هر گروه شامل مجموعه‌ای از $N=\{1,2,\ldots,N\}$ دستگاه‌های IoT و یک سرور لبه است که در آن یک مجموعه از $C=\{1,2,\ldots,c\}$ ماشین‌های مجازی برای پردازش داده‌ها مستقر شده است.

هر دستگاه IoT اطلاعات حسگری را جمع‌آوری کرده و یک سری به‌روزرسانی وضعیت سیستم را ایجاد می‌نماید. در این مقاله، یک سیستم شیار زمانی گسسته در نظر گرفته شده که زمان را با شیوه‌ای به بخش‌های مساوی تقسیم می‌کند. هدف ما در پاسخ به درخواست‌های دستگاه IoT، انتخاب پردازنده محلی یا سرور لبه برای انجام به‌روزرسانی

1. Frequency Division Multiple Access



شکل ۱: سناریوی سیستم اینترنت اشیای صنعتی [۲].

وضعیت بر اساس محیط شبکه در هر شیار زمانی t است. برای پردازش محلی، مقداری از انرژی بر روی دستگاه IoT مصرف می‌شود؛ در حالی که برای پردازش لبه، دستگاه IoT باید اطلاعات خود را به سرور لبه در ایستگاه پایه انتقال دهد و از سرور لبه برای پردازش درخواست کند که همچنین مصرف انرژی، تأخیر ارتباطی و زمان پردازش را می‌طلبد. تعادل بین این موارد باید با دقت مورد بررسی قرار گیرد تا به بهینه‌سازی کارایی انرژی برسیم و در عین حال اطلاعات را به‌روز نگه داریم [۱].

۳-۲ مدل سیستم

۳-۲-۱ مدل میزان کار

در ابتدای فاصله زمانی t، احتمال این که iامین دستگاه IoT تقاضاهای خود را به سرور لبه منتقل کند با $P_i^s(t)$ نشان داده می‌شود، جایی که $0 \le P_i^S(t) \le 1$ با توجه به خاصیت فرایند پواسون می‌دانیم که ورود وظایف به سرور لبه همچنین یک فرایند پواسون با نرخ میانگین $P_i^S(t)\lambda_i$ را دنبال می‌کند. در عین حال، نرخ ورود به‌روزرسانی وضعیت که در دستگاه IoT محلی پردازش می‌شود توسط $(1-P_i^S(t))\lambda_i$ بیان می‌شود. اندازه میانگین داده‌های تولیدشده از iامین دستگاه IoT با θ_i بیان می‌شود. مانند کارهای موجود فرض می‌کنیم اندازه بارکاری (داده) با توزیع نمایی مطابقت دارد [۱].

۳-۲-۲ مدل محاسباتی

الف) مدل محاسباتی محلی: برخی از بارکاری‌های سبک می‌توانند به صورت محلی در دستگاه IoT پردازش شوند. از آنجا که هیچ ارتباط داده‌ای با لبه وجود ندارد، تنها باید زمان پردازش صرف‌شده را محاسبه و وضعیت اطلاعات از دستگاه‌های IoT را دریافت کنیم. با استفاده از مدل بارکاری، دستگاه IoT با صف M/M/۱ مدل می‌شود. با استفاده از نظریه صف، زمان پاسخ میانگین $T_i^A(t)$ برای درخواست‌های پردازش‌شده محلی در هر دستگاه با (۱) بیان می‌شود و μ_i^A نرخ خدمت‌دهی را برای توان محاسباتی دستگاه iام نشان می‌دهد

$$T_i^A(t) = \frac{1}{\mu_i^A - (1-P_i^S(t))\lambda_i}, \quad i \in \mathcal{N} \qquad (1)$$

ب) مدل محاسباتی: چندین دستگاه IoT به سرور لبه متصل می‌شوند و در نتیجه درخواست‌های آنها ممکن است در گره لبه همگرا شوند. ما می‌دانیم که هر دستگاه IoT درخواست‌های خود را به سرور لبه با توزیع پواسون ارسال می‌کند. جمع‌آوری همه فرایندهای پواسون از دستگاه‌های IoT نیز یک فرایند پواسون است که نرخ ورودی آن به‌صورت (۲) نمایش داده می‌شود

$$\lambda_{total}^A(t) = \sum_{i=1}^{N} P_i^S(t)\lambda_i \qquad (2)$$

برای جلوگیری از ازدحام درون صف که ممکن است منجر به کاهش عملکرد غیرمنتظره شود، ممکن است یک طرح کنترل دسترسی وجود داشته باشد که حداکثر بارکاری را در سرور لبه محدود کند تا وضعیت صف را استوار نگه دارد. $\lambda_{\max}^S$ حداکثر نرخ ورودی در سرور لبه را نشان می‌دهد. بنابراین نرخ ورودی در سرور لبه به شکل (۳) قابل بیان است

$$\lambda_p^S(t) = \begin{cases} \lambda_{total}^A(t), \lambda_{total}^A(t) \le \lambda_{\max}^S \\ \lambda_{\max}^S, \lambda_{total}^A(t) > \lambda_{\max}^S \end{cases} \qquad (3)$$

باید توجه داشت سرور لبه‌دار c ماشین مجازی همگن دارد که می‌توانند به طور هم‌زمان چندین درخواست به‌روزرسانی وضعیت را از دستگاه‌های IoT پردازش کنند. فرض می‌کنیم نرخ سرویس‌دهی هر ماشین مجازی را با μ^S نشان دهیم؛ بنابراین سرور لبه‌دار می‌تواند با یک صف M/M/c مدل شود. بهره آن می‌تواند با معادله زیر محاسبه گردد

$$\rho^S(t) = \frac{\lambda_p^S(t)}{c\mu^S} \qquad (4)$$

میانگین زمان انتظار برای پردازش درخواست شامل زمان صف‌سازی و زمان سرویس‌دهی است. میانگین زمان صف‌سازی برای هر به‌روزرسانی وضعیت در سرور لبه برابر با زمانی است که یک درخواست در صف منتظر می‌ماند که به صورت متوسط توسط فرمول زیر محاسبه می‌شود

$$T_w(t)=\frac{G(t)}{c\mu^S-\lambda_p^S(t)} \quad (۵)$$

$$G(t)=\frac{\frac{(c\rho^S(t))^c}{c!}\frac{۱}{۱-\rho^S(t)}}{\sum_{k=۰}^{c-۱}\frac{(c\rho^S(t))^k}{k!}+\frac{c\rho^S(t)^c}{c!}\frac{۱}{۱-\rho^S(t)}} \quad (۶)$$

میانگین زمان سرویس در سرور لبه شبکه به صورت زیر قابل بیان است

$$T_s=\frac{۱}{\mu^S} \quad (۷)$$

از (۵) تا (۷) میانگین زمان انتظار برای انتقال محاسبات به صورت زیر قابل بیان است

$$T_{wait}^S(t)=T_w(t)+T_s \quad (۸)$$

۳-۲-۳ مدل ارتباطی

اگر دستگاه‌های IoT تصمیم به انتقال اطلاعات به سرور لبه بگیرند، زمان انتقال داده اضافی مصرف می‌شود. در فاصله زمانی t، قدرت انتقال دستگاه IoT i با $P_i(t)$ نشان داده می‌شود که $۰<P_i(t)<P_i^{\max}$ و $P_i^{\max}$ حداکثر قدرت انتقال دستگاه i است. توانایی کانال بین دستگاه i IoT و سرور لبه با h_i نشان داده می‌شود. با توجه به تئوری شانون که در ارزیابی عملکرد انتقال داده به طور گسترده‌ای استفاده شده است، می‌توانیم سرعت انتقال داده بالا به سمت بالا را برای انتقال محاسبات دستگاه IoT i به سرور لبه به شرح زیر به دست آوریم

$$R_i(t)=\omega\log_۲(۱+\frac{P_i(t)h_i}{\sigma^۲}) \quad (۹)$$

در اینجا ω پهنای باند کانال بی‌سیم و $\sigma^۲$ نویز سفید گوسی میانگین است. برای آسان‌سازی تحلیل، فرض می‌کنیم که کانال‌های بی‌سیم بین دستگاه‌های IoT با یکدیگر متعامد هستند. با استفاده از (۹) می‌توان زمان انتقال برای ارسال اطلاعات به دستگاه لبه توسط دستگاه IoT i را به صورت زیر بیان کرد

$$T_i^t(t)=\frac{(P_i^S(t)\lambda_i)\theta_i}{R_i(t)} \quad (۱۰)$$

اندازه نتایج محاسباتی نسبتاً کوچک‌تر از اندازه داده‌های اصلی است و زمان ارسال نتیجه و مصرف انرژی مربوطه می‌تواند نادیده گرفته شود.

۳-۲-٤ مدل طول عمر اطلاعات (سالخوردگی اطلاعات)

تازگی اطلاعات در پردازش داده با استفاده از سالخوردگی اطلاعات (AoI)[1] پیشنهاد شده است. به طور رسمی در این مقاله، AoI را به عنوان مدت زمانی که از تولید اطلاعات وضعیت در یک دستگاه IoT تا تکمیل پردازش داده در دستگاه IoT یا سرور لبه گرفته می‌شود، تعریف کرده‌ایم.

با استفاده از مدل‌های ریاضی که در بالا ارائه شد به طور خاص با استفاده از (۱)، (۸) و (۱۰)، تازگی اطلاعات در زمان t می‌تواند به عنوان زیر محاسبه شود

$$T_{AOI}(t)=T_i(t)+T_S(t) \quad (۱۱)$$

در اینجا $T_i(t)$ زمان مصرفی برای پردازش و ارسال به‌روزرسانی‌های وضعیت به صورت محلی است که به صورت زیر بیان می‌شود

$$T_i(t)=(۱-P_i^S(t))T_i^A(t)+P_i^S(t)T_i^t(t) \quad (۱۲)$$

بخش دوم AoI در سمت راست (۱۱)، زمان صرف‌شده توسط سرور لبه جهت پردازش اطلاعات وضعیت است که به شکل زیر بیان می‌گردد

$$T_S(t)=P_i^S(t)T_{wait}^S(t) \quad (۱۳)$$

بنابراین از (۱۱) می‌توان میانگین AoI را به صورت زیر به دست آورد

$$T_{AOI}=\frac{۱}{N}\frac{۱}{T}\sum_{i=۱}^{N}\sum_{t=۱}^{T}\{T_{AOI}(t)\} \quad (۱۴)$$

$$=\frac{۱}{N}\frac{۱}{T}\sum_{i=۱}^{N}\sum_{t=۱}^{T}\{(۱-P_i^S(t))\frac{۱}{\mu_i^A-(۱-P_i^S(t))\lambda_i}+$$

$$P_i^S(t)\frac{(P_i^S(t)\lambda_i)\theta_i}{\omega\log_۲(۱+\frac{P_i(t)h_i}{\sigma^۲})}+ \quad (۱۵)$$

$$P_i^s(t)\frac{\frac{(c\rho^s(t))^c}{c!}\frac{۱}{۱-\rho^s(t)}}{\sum_{k=۰}^{c-۱}\frac{(c\rho^s(t))^k}{k!}+\frac{(c\rho^s(t))^c}{c!}\frac{۱}{۱-\rho^s(t)}}\}$$

$$\frac{۱}{c\mu^s-\lambda_p^S(t)(t)}+\frac{۱}{\mu^s}$$

۳-۲-۵ مدل انرژی

مصرف انرژی در دستگاه IoT i در اصل شامل دو بخش است: ۱) مصرف انرژی برای پردازش محلی اطلاعات وضعیت و ۲) توان مصرفی برای انتقال داده‌های حسگری به سرورهای لبه. برای بخش اول، مصرف انرژی برای پردازش داده‌ها با فرکانس پردازشی دستگاه IoT در ارتباط مستقیم است. مصرف انرژی در فاصله زمانی t می‌تواند با استفاده از (۱۶) محاسبه شود که در آن f فرکانس پردازشی دستگاه IoT و κ_i پارامتر ظرفیت مؤثر دستگاه IoT i است

$$E_i^A(t)=\kappa_i f^۳T_i^A(t) \quad (۱۶)$$

برای بخش دوم، مصرف انرژی انتقال می‌تواند با ضرب نرخ توان انتقال $P_i(t)$ در زمان انتقال $T_i^t(t)$ محاسبه شود. به طور رسمی آن را می‌توان به شکل زیر بیان کرد

$$E_i^S(t)=P_i(t)T_i^t(t)=P_i(t)\frac{P_i^S(t)\lambda_i\theta_i}{R_i(t)} \quad (۱۷)$$

با توجه به (۱۶) و (۱۷) می‌توان مصرف میانگین انرژی در سیستم IIoT را به شکل زیر بیان کرد

$$E_{ave}=\frac{۱}{N}\frac{۱}{T}\sum_{i=۱}^{N}\sum_{t=۱}^{T}\{(۱-P_i^S(t))E_i^A+P_i^S(t)E_i^S\} \quad (۱۸)$$

$$E_{ave}=\frac{۱}{N}\frac{۱}{T}\sum_{i=۱}^{N}\sum_{t=۱}^{T}\{(۱-P_i^S(t))E_i^A+P_i^S(t)E_i^S\}=$$

$$\frac{۱}{N}\frac{۱}{T}\sum_{i=۱}^{N}\sum_{t=۱}^{T}\{(۱-P_i^S(t))(\frac{\kappa_i f^۳}{\mu_i^A-(۱-P_i^S(t))\lambda_i})+ \quad (۱۹)$$

$$P_i^S(t)(P_i(t)\frac{P_i^S(t)\lambda_i\theta_i}{\omega\log_۲(۱+\frac{P_i(t)h_i}{\sigma^۲})})\}$$

1. Age of Information

۳-۳ مسئله فرموله‌شده

هدف، کاهش میانگین انرژی مصرفی توسط دستگاه‌هاست که به صورت مستقیم با دیگر مؤلفه‌ها همانند زمان انتقال و حداکثر قدرت پردازشی سرور لبه در ارتباط می‌باشد. در (۲۰) تا (۲۵) محدودیت‌هایی در فضای مسئله تعریف گردیده است. تابع هدفی که با (۲۰) تعریف شده است، کمینه‌سازی مصرف انرژی میانگین بلندمدت را مورد هدف قرار داده و نیز با استفاده از (۲۱)، محدودیتی برای مقدار سالخوردگی اطلاعات تعیین گردیده که اطلاعات جمع‌آوری‌شده به‌روز باشند. محدودیت (۲۲) نشان می‌دهد که احتمال انتقال داده‌ها بین پردازش محلی و پردازش ابری باید در محدوده ۰ تا ۱ باشد. رابطه (۲۳) محدودیتی را برای حداکثر توان انتقال برای هر دستگاه IoT تعریف می‌کند. نهایتاً (۲۴) و (۲۵) تضمین می‌کنند که صف پایدار هر دستگاه IoT و سرور ابری باشد. این محدودیت‌ها به عنوان مسئله P۱ مطرح شده‌اند

$$P1: \min_{P_i^S(t),P_i(t)} E_{ave} \quad (۲۰)$$

s.t.

$$T_{AOI}(t) \le \varepsilon \ , \ \forall i \in \mathcal{N} \quad (۲۱)$$

$$0 \le P_i^S(t) \le 1 \ , \ \forall i \in \mathcal{N} \quad (۲۲)$$

$$0 \le P_i(t) \le P_i^{\max} \ , \ \forall i \in \mathcal{N} \quad (۲۳)$$

$$\frac{1}{N}\sum_{i=1}^{N}(1-P_i^S(t))\lambda_i > 0 \quad (۲۴)$$

$$\lambda_p^S(t) < c\mu^S \ , \ \forall i \in \mathcal{N} \quad (۲۵)$$

۴- الگوریتم ژنتیک تقویتی پیشنهادی

با در نظر داشتن معایب الگوریتم‌های پیشین [۱۵] در حوزه بارگذاری درخواست‌های شبکه اینترنت اشیا می‌توان چالش‌های زیر را در این روش‌ها بیان نمود:

۱) دقت الگوریتم‌های مبتنی بر مدل‌های آماری مثل مارکوف با افزایش تعداد وضعیت‌ها (افزایش تعداد دستگاه‌های IoT) به شدت کاهش می‌یابد.

۲) الگوریتم‌های گسسته موجود مناسب کار در محیط‌های پیوسته نیست (وقتی که یک توالی منطقی میان درخواست‌های دستگاه‌های IoT وجود داشته باشد).

۳) مدل‌های بهینه‌سازی ارائه‌شده، مشکل عدم ایجاد توازن میان عامل‌ها[1] را دارند که در فضای مسئله برخی از عامل‌ها مقدار بهینه محلی[2] و برخی دیگر بهینه سراسری[3] را پویش می‌نمایند.

به همین دلایل نیز پیشنهاد ما استفاده از الگوریتم ژنتیک به جای الگوریتم‌های بهینه‌سازی حاضر می‌باشد. دلایل استفاده از الگوریتم ژنتیک به شرح زیر است [۱۶]:

۱) قابلیت استفاده در فضای مسئله پیوسته و گسسته

۲) قدرت همگرایی بالا میان پاسخ‌های بهینه محلی و سراسری

۳) نتایج قابل قبول با افزایش تعداد وضعیت‌ها در فضای مسئله

در نتیجه می‌توان اذعان داشت که الگوریتم ژنتیک می‌تواند به خوبی چالش‌های الگوریتم MDP[4] را پوشش داده و مسئله P۱ را بهینه‌سازی نماید. می‌بایست توجه کرد که الگوریتم ژنتیک قابلیت ارائه به صورت یک روش تقویتی را دارا بوده و به صورت ذاتی نیز این کار را با انتخاب نسل نخبه انجام می‌دهد [۸].

۴-۱ الگوریتم ژنتیک تقویتی

مسئله P۱ که در بخش قبل ارائه شد، یک مسئله بهینه‌سازی پویای گسسته است که با توجه به وضعیت جاری سیستم و وضعیت‌هایی که می‌تواند به آن تغییر یابد، فرموله‌سازی را انجام می‌دهد. در مسئله P۱، محدودیت‌هایی وجود دارد که ارضای این محدودیت‌ها به واسطه استفاده از مؤلفه‌های تشویق و تنبیه انجام می‌پذیرد. به این روند، الگوریتم یادگیری تقویتی گفته می‌شود؛ زیرا با افزایش ادراک عامل‌ها نسبت به محیط، پاسخ صحیح (بهینه) به دست می‌آید (کاهش مصرف انرژی در ارسال درخواست‌های دستگاه‌های IoT به سرورهای لبه). در ادامه فضای حالت و نوع عملکرد الگوریتم پیشنهادی ارائه شده است.

۴-۲ فضای حالت

منظور از فضای حالت، مجموع اقداماتی است که هر عامل (کروموزوم) می‌تواند در فضای مسئله داشته باشد. در مسئله P۱ در هر بازه زمانی t، حالت سیستم به صورت دوتایی $S(t)=(\Theta_s(t),\Theta_e(t))$ تعریف می‌شود که حالت صف دستگاه‌های IoT و سرور مرکزی را نشان می‌دهد. تعریف دقیق فضای حالت به شرح زیر است.

مقدار $\Theta_s(t)$ به عنوان تعداد درخواست‌های در حال انتظار برای پردازش در هر دستگاه IoT تعریف می‌شود و می‌تواند به وسیله (۲۶) بیان گردد که در آن $Q_{si}(t)$ تعداد کارهای موجود در بافر دستگاه i IoT را نشان می‌دهد

$$\Theta_s(t)=\{Q_{s1}(t),Q_{s2}(t),\ldots,Q_{si}(t),\ldots,Q_{sN}(t)\} \quad (۲۶)$$

صف سرور مرکزی با $\Theta_e(t)$ نشان داده می‌شود و به صورت رابطه زیر تعریف می‌گردد

$$\Theta_e(t)=\{Q_{e1}(t),Q_{e2}(t),\ldots,Q_{ej}(t),\ldots,Q_{ec}(t)\} \quad (۲۷)$$

که $Q_{ej}(t)$ تعداد درخواست‌های موجود در صف j امینه مجازی روی سرور مرکزی را نشان می‌دهد.

۴-۳ مجموعه اقدام‌های هر کروموزم در انتخاب فعالیت‌های بهینه

در هر بازه زمانی t، هر کروموزوم می‌تواند اقدام‌هایی را با توجه به وضعیت شبکه انجام دهد که با $A(t)=(\gamma_t,\delta_t,\varepsilon_t)$ نشان داده می‌شود؛ بنابراین انتخاب درخواست‌های مناسب با قدرت پردازشی ماشین‌های مجازی لبه و در نظر داشتن صف درخواست‌ها در سرور، مهم‌ترین رویکرد کروموزوم‌ها خواهد بود که با توجه به وضعیت فعلی شبکه یا $S(t)$ تصمیم به offloading دستگاه‌های IoT می‌گیرد.

- γ_t نشان‌دهنده احتمال offloading دستگاه‌های IoT است که می‌تواند به صورت زیر بیان شود

$$\gamma_t=\{P_1^S(t),P_2^S(t),\ldots,P_i^S(t),\ldots,P_N^S(t)\} \quad (۲۸)$$

1. Agent
2. Exploitation
3. Exploration
4. Markov Decision Process

- δ_t بیانگر قدرت انتقال دستگاه‌های IoT است و می‌تواند به صورت زیر بیان شود

$$\delta_t = \{P_1(t), P_2(t), \ldots, P_i(t), \ldots, P_N(t)\} \quad (۲۹)$$

- ε_t سیاست‌های برنامه‌ریزی دستگاه‌های IoT را نشان می‌دهد و می‌تواند به صورت زیر بیان شود

$$\varepsilon_t = \{W_1(t), W_2(t), \ldots, W_i(t), \ldots, W_N(t)\} \quad (۳۰)$$

بنابراین فضای عمل $A(t)$ می‌تواند به صورت زیر بیان شود

$$\begin{aligned} A(t) = \{&(P_1^S(t), P_2^S(t), \ldots, P_i^S(t), \ldots, P_N^S(t)), \\ &(P_1(t), P_2(t), \ldots, P_i(t), \ldots, P_N(t)), \\ &(W_1(t), W_2(t), \ldots, W_i(t), \ldots, W_N(t))\} \end{aligned} \quad (۳۱)$$

٤-٤ تابع پاداش و تنبیه

در بازه زمانی t پس از انجام عملیات $A(t)$ در حالت $S(t)$، هر کروموزم با توجه به عملکردش ضریبی از پاداش یا $R(t)$ را دریافت می‌کند. با توجه به این که هدف مسئله P۱ کمینه‌کردن مصرف انرژی سیستم تحت محدودیت میانگین سالخوردگی اطلاعات است (محدودیت‌های فضای مسئله)، تابع پاداش به شرح زیر تعریف می‌شود

$$R(t) = \sum_{i \in \mathcal{N}} \mathbb{E}[E_{ave}(t) + T_{AOI}(t) + \Psi(t)] \quad (۳۲)$$

در (۳۲) علاوه بر مصرف انرژی متوسط $E_{ave}(t)$، مقدار سالخوردگی اطلاعات را در تابع پاداش برای کمینه‌کردن آن در الگوریتم ژنتیک خود قرار داده‌ایم. در عین حال، یک تابع جریمه $\Psi(t)$ را تعریف می‌کنیم تا محدودیت‌های مشکل P۱ را تضمین کنیم. به طور خاص اگر هر محدودیتی برآورده نشود، مقدار بزرگی مثبت به تابع پاداش اضافه می‌شود.

تابع جریمه یک عدد بزرگ با توجه به کیفیت پاسخ‌های ارائه‌شده توسط هر کروموزوم است که باعث می‌شود در نسل‌های آتی به عنوان کروموزوم نخبه انتخاب نگردند. مسئله بسیار مهم در این تابع، جلوگیری از افزایش و یا کاهش یک‌باره قدرت کروموزم‌ها در فضای مسئله است که باعث می‌شود در دورهای ابتدایی الگوریتم که انتخاب‌ها به صورت تصادفی و ناآگاهانه است، باعجله تصمیم‌گیری نشده و هر بار فرصتی به کروموزوم‌های غیربهینه داده شود (البته کروموزوم‌هایی که محدودیت‌های فضای مسئله را رعایت نکنند، به طور کلی از فضای مسئله حذف گردیده و به آنها فرصت دوباره‌ای داده نمی‌شود).

٤-٥ تابع برازندگی

تابع برازندگی در الگوریتم ژنتیک جهت ارزیابی عملکرد هر کروموزوم در تخصیص منابع به دستگاه‌های IoT استفاده می‌شود. هدف از این تابع، تعیین ارزش هر کروموزوم بر اساس عملکرد و کارایی آن در انجام تخصیص منابع بهینه است؛ بنابراین تابع بهینه مشخص می‌کند که در هر دور چه کروموزوم‌هایی تقویت شده و چه کروموزوم‌هایی تضعیف گردند.

نحوه محاسبه این تابع به این صورت می‌باشد که ابتدا برای هر کروموزوم، اقدام به تخصیص منابع مربوطه به دستگاه‌های IoT می‌شود و پس از آن با استفاده از تابع P۱ که به عنوان تابع هدف ژنتیکی شناخته می‌شود، عملکرد و کارایی تخصیص منابع توسط هر کروموزوم محاسبه می‌شود.

تابع تناسب در الگوریتم ژنتیک با رعایت محدودیت‌های فضای مسئله، اعضای نخبه را انتخاب می‌کند؛ بنابراین می‌بایست یک ساختار مثل آرشیو در نظر گرفته و کروموزم‌هایی را که نخبه هستند در آن ذخیره‌سازی نمود.

مهم‌ترین چالش در الگوریتم‌های بهینه‌سازی گسسته، پویش وسیع فضای مسئله در کنار فرار از بهینه محلی است. در خصوص بهینه محلی می‌توان گفت که کروموزوم‌ها به صورت کاذب ضریب تشویق دریافت نموده و گمان بر این است که پاسخ ارائه‌شده، بهینه‌ترین پاسخ ممکن در فضای مسئله است؛ در حالی که پاسخ بهینه نیز وجود دارد.

استفاده از (۳۲) جهت ایجاد تعادل میان کروموزوم‌های نخبه و غیرنخبه منجر به ارائه فرصت مجدد به کروموزوم‌ها می‌شود؛ زیرا به دلیل گسسته‌بودن فضای مسئله و پویایی آن، این احتمال وجود دارد که با تغییر ژن برخی از کروموزوم‌ها، فضای مسئله حالت بهینه را به خود بگیرد. این تغییر در ژن‌های کروموزوم‌ها توسط نرخ جهش و عملگر ترکیب عرضی یا Cross Over انجام می‌پذیرد که باعث می‌شود جمعیت گوناگونی خود را حفظ نماید.

الگوریتم بهینه‌سازی ژنتیک جهت کنترل انرژی و محاسبه تخصیص منابع در شبکه‌های IoT موارد زیر را انجام می‌دهد (شکل ۲):

۱) **پارامترهای ژنتیک:** این بخش، پارامترهای مرتبط با الگوریتم ژنتیک را مقداردهی اولیه می‌کند. این پارامترها شامل اندازه جمعیت[1] کروموزوم‌ها، تعداد نسل‌ها (maxit)، میزان نرخ جهش با یک مقدار ثابت[2] و تعداد اعضای نخبه در هر نسل[3] می‌باشد.

۲) **شروع تخصیص منابع:** در این بخش، هر ژن کروموزوم‌ها به‌صورت تصادفی، درخواست‌های n دستگاه IoT را در قالب یک Episode به سرور لبه ارسال می‌نماید (شبیه‌سازی ارسال درخواست‌ها).

۳) شروع شمارنده الگوریتم از مقدار ۱ تا max_it (این تکرار می‌تواند در صورت همگرایی عامل‌ها در نقطه بهینه متوقف گردد.)

۴) اعمال تابع تناسب یا هدف بر روی کروموزوم‌های هر نسل و بررسی میزان کیفیت پاسخ‌های ارائه‌شده توسط هر کروموزوم با در نظر داشتن فضای مسئله P۱ و محدودیت‌های موجود در آن

۵) انتخاب کروموزوم‌های برتر در نسل جاری و اضافه‌نمودن آنها به فضای آرشیو (پاسخ‌هایی که میزان انرژی مصرفی در شبکه را کمینه‌سازی می‌نمایند.)

۶) اعمال ضریب تشویق بر روی کروموزوم‌های بهینه و ضریب تنبیه بر روی کروموزوم‌های ضعیف‌تر با استفاده از (۳۲)

۷) حذف کلی کروموزوم‌هایی که فضای مسئله و محدودیت‌های آن را رعایت نمی‌کنند.

۸) به‌روزرسانی ضریب تنبیه (استفاده از تابع نمایی) با توجه به موقعیت جاری الگوریتم (در این حالت هرچه الگوریتم پیش‌تر می‌رود، ضریب تنبیه نیز با حساسیت بالاتری انتخاب می‌شود؛ زیرا در دورهای انتهایی الگوریتم، عامل‌ها در اطراف فضای بهینه تجمع کرده و حرکت آنها در فضای مسئله باید با حساسیت‌های بیشتری انجام پذیرد.)

۹) ترکیب کروموزوم‌های نخبه نسل جاری با یکدیگر (اپراتور ترکیب عرضی[4]) و تولید نسل جدید

۱۰) اعمال عملگر جهش ژنتیک بر روی فرزندان تولیدشده جهت فرار از بهینه محلی و حفظ گستردگی پاسخ‌ها در فضای مسئله (بهینه سراسری)

1. Population
2. Mutation Rate
3. Ellitism
4. Cross Over

Algorithm 1 - Proposed Algorithm for Improving Offloading with Genetic Optimization (IOGO)
Input: Training episode number Tepi; training step number Tstep; model update frequency U; decay factor ζ; model update number t_epochs
Output: Offloading probability of IoT devices γt; transmission power of IoT devices δt; scheduling policy of IoT devices □t
Initialize random seed, experience replay pool D, system environment, network parameters
Set time_steps = 0
for episode = 1 to Tepi do
    Generating the initial population of chromosomes in the Genetic Algorithm.
    Set reward = 0, energy = 0
    for step = 1 to Max iteration do
        Obtain action Network from the genes of the chromosomes in the current generation
        Obtain the Elite chromosomes, calculate reward R(t)
        Store Elite chromosomes and removal of non-elite chromosomes.
        for i = 1 to t_epochs do
            Combining elite chromosomes with each other and generating new chromosomes for the next generation.
            Calculate the decay reward discount_reward = r + ζ*discount_reward
            r ← discount_reward
            Update network parameters φ
        end for
        Save the updated network parameters Πold ← Πφ
        end for
    end for
Return γt, δt, □t

شکل ۲: الگوریتم بهینه‌سازی.

جدول ۲: پارامترهای ورودی.

Notations	Definitions	Value
C	Number of VMs	۴
θ	Task size	۱MB ~ ۵MB
f	The computational frequency of IoT devices	۱ GHz
μ_S	The computational frequency of edge server	۴/۴ GHz
ω	Bandwidth of IoT device	۱ MHz
$p_i^{\max}$	The max transmission power of IoT device	۱۰۰ mW
σ^2	Gaussian white noise	−۱۰۰ dBm

۱۱) ایجاد تعادل میان پاسخ‌های بهینه سراسری و محلی (عملگرهای Exploration و Exploitation) جهت جلوگیری از تغییر مقدار یک‌باره کروموزوم‌ها در فضای مسئله و پَرِش از محدوده‌های بهینه بالقوه با استفاده از (۳۳)

$$a = ۲ - t \times \frac{۲}{T} \qquad (۳۳)$$

در (۳۳) منظور از T تعداد کل دورهای اجرایی و t دور جاری می‌باشد. به این ترتیب، نرخ حرکت عامل‌ها در فضای مسئله به یک مقدار مشخص (بر اساس مقدار اولیه a) نگاشت می‌شود که این روند به طور مستقیم در تعیین همگرایی عامل‌ها تأثیرگذار می‌باشد (مقدار a در الگوریتم پیشنهادی یک عدد کوچک مثل ۰٫۰۱ است که می‌تواند حساسیت‌های الگوریتم در پویش وسیع فضای مسئله را تضمین نماید).

۱۲) تولید نسل جدید و بازگشت به مرحله سوم

۱۳) ارائه پاسخ بهینه از کروموزوم نخبه نسل آخر الگوریتم به عنوان پاسخ قطعی از اختصاص درخواست‌ها به سرور لبه (به این حالت Offloading گفته می‌شود.)

۱۴) به‌روزرسانی وضعیت شبکه اینترنت اشیا از لحاظ میزان انرژی مصرف‌شده توسط هر دستگاه و آمادگی جهت انجام عملیات Offloading در زمان $t+۱$

در کل، این الگوریتم با استفاده از الگوریتم ژنتیک بهترین تخصیص منابع برای دستگاه‌های IoT در شبکه را با هدف بهینه‌سازی مصرف انرژی و تصمیم‌گیری‌های انتقال داده انجام می‌دهد. هدف این الگوریتم بهبود عملکرد کلی سیستم IoT است.

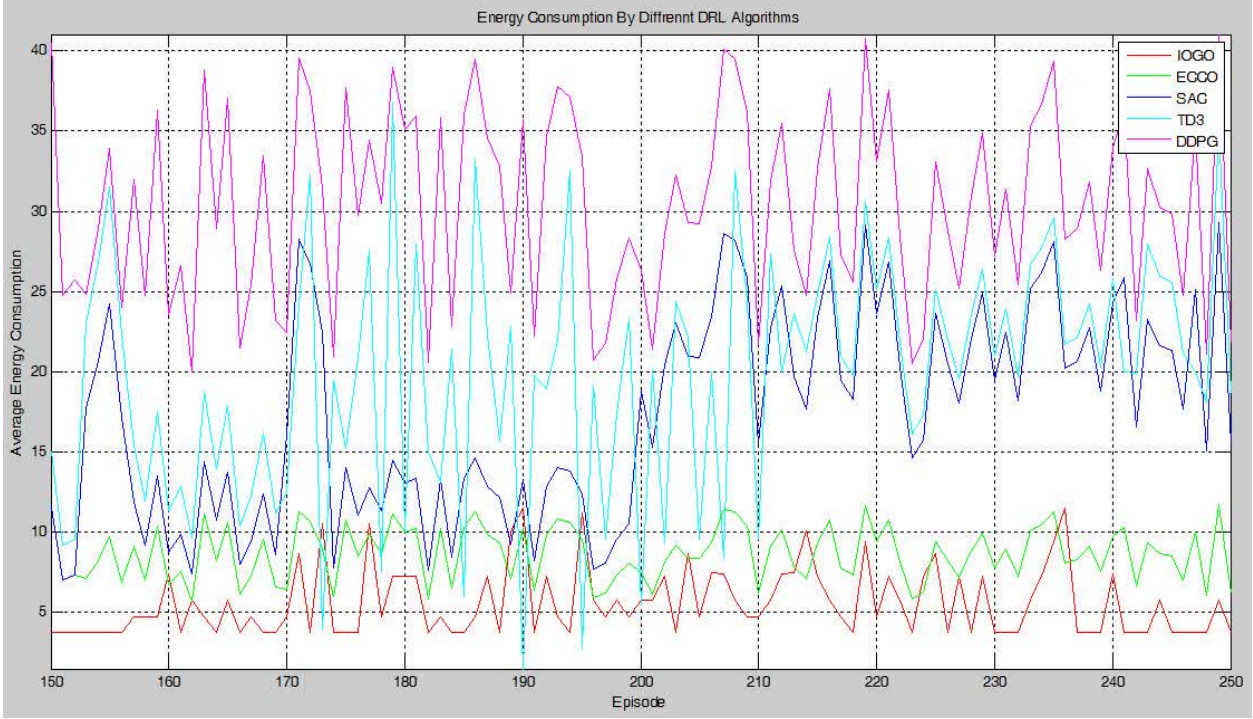
۵- ارزیابی

محیط سرورهای لبه در سطح شهر گسترده شده‌اند و اشیا اطلاعات خود را به این نودها ارسال می‌کنند. در محیط شبیه‌سازی چهار ماشین مجازی در هر سرور لبه در نظر گرفته شده است. وظایف از دیتاست تلکام گرفته شده و طول هر وظیفه (داده‌های جدید دریافت‌شده از سنسور) به صورت توزیع تصادفی یکنواخت بین ۱ تا ۵ مگابایت انتخاب گردیده است. فرکانس سرور اشیا یک و فرکانس سرور لبه چهار گیگاهرتز در نظر گرفته شده است. پهنای باند سیستم بی‌سیم یک مگاهرتز در نظر گرفته شده و پارامترهای استفاده‌شده برای ورودی و مقایسه هر دو الگوریتم طبق جدول ۲ است.

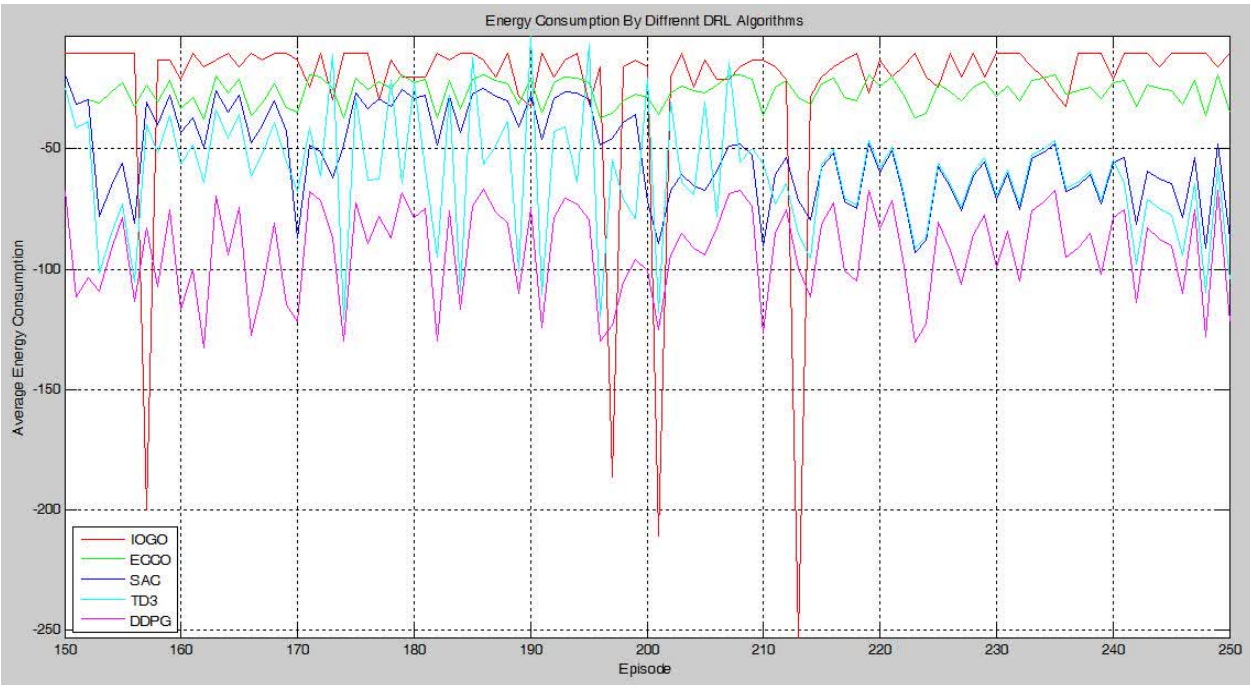
شکل ۳، عملکرد روش پیشنهادی را در مقایسه با الگوریتم‌های دیگر از جمله بهترین آنها (ECCO) در محاسبه میانگین انرژی مصرفی دستگاه‌ها و شکل ۴ ضریب پاداش در دورهای مختلف را نشان می‌دهد (هر دور یا Episode، تعداد دفعاتی است که لیست درخواست‌های پردازشی سرور لبه پر و خالی می‌شود) که برتری الگوریتم پیشنهادی نسبت به دیگر الگوریتم‌ها قابل مشاهده است.

همان طور که در شکل‌های ۳ و ۴ مشخص است، الگوریتم ژنتیک تقویتی به دلیل برخورداری از قابلیت بهینه سراسری، مناسب‌بودن برای محیط‌های پیوست و تعادل بین کاوش و بهره‌برداری، عملکرد بهتری را در کاهش مصرف انرژی (بهینه‌سازی مسئله P۱) و اختصاص ضریب پاداش برجای گذاشته است (خطوط قرمزرنگ شکل ۴ خروجی الگوریتم IOGO را با استفاده از روش ژنتیک و خطوط سبزرنگ خروجی بهترین الگوریتم قبلی یعنی ECCO را با استفاده از روش MDP نشان می‌دهند).

الگوریتم ژنتیک تقویتی و بازیگر- منتقد نرم (SAC) دو رویکرد متفاوت برای پیش‌بینی حجم کار سرور در اینترنت اشیا هستند. الگوریتم پیشنهادی یک تکنیک بهینه‌سازی مبتنی بر الگوریتم ژنتیکی است که هدف آن به حداکثر رساندن طول عمر شبکه در یک سیستم IoT چندکاربره است. از سوی دیگر SAC یک الگوریتم یادگیری تقویت عمیق (DRL) است که نسبت تقسیم وظیفه، فرکانس‌های چرخه CPU، تخصیص پهنای باند و تخصیص فرکانس سیکل CPU را بهینه می‌کند؛



شکل ۳: نتایج به‌دست‌آمده در مقایسه روش پیشنهادی و روش‌های قبلی در متوسط مصرف انرژی مصرفی.



شکل ۴: مقایسه نتایج الگوریتم‌ها.

در حالی که الگوریتم ژنتیک تقویتی بر بهینه‌سازی طول عمر شبکه تمرکز دارد. SAC قصد دارد که عملکرد سیستم را با بهینه‌سازی مشترک پارامترهای مختلف به حداکثر برساند؛ بنابراین مزیت الگوریتم پیشنهادی نسبت به SAC در پیش‌بینی حجم کار سرور در اینترنت اشیا در تمرکز خاص آن بر بهینه‌سازی طول عمر شبکه است که ممکن است برای کاربردها یا سناریوهای خاص مناسب‌تر باشد.

الگوریتم مدل گرادیان سیاست قطعی عمیق با تأخیر دوقلو (TD۳) در پیش‌بینی حجم کار سرور در اینترنت اشیا این است که TD۳ یک الگوریتم داده‌محور و بدون مدل است که سطح توان مداوم آنتن را به نام تخصیص توان (PA) کنترل می‌کند. از روش‌های مبتنی بر مدل مانند برنامه‌نویسی کسری (FP) و MMSE وزنی (WMMSE) و همچنین الگوریتم‌های بدون مدل مانند شبکه Q عمیق (DQN) و DDPG از نظر عملکرد نرخ جمع با قدرت تعمیم خوب پیشی می‌گیرد؛ برعکس الگوریتم پیشنهادی که بر مصرف انرژی و پیش‌بینی حجم کاری تأکید دارد.

مارکوف یک مدل احتمالاتی است که بر فرض خاصیت مارکوف متکی است که ممکن است در همه موارد درست نباشد. الگوریتم ژنتیک تقویتی با توانایی خود در انطباق و یادگیری از تغییر بارهای کاری، انعطاف‌پذیری و استحکام بیشتری را در پیش‌بینی حجم کار سرور ارائه می‌دهد و نیز این الگوریتم می‌تواند پیش‌بینی حجم کار چندمتغیره را با در نظر گرفتن پارامترهای منابع متعدد انجام دهد. این امکان پیش‌بینی جامع و دقیق‌تر حجم کار سرور را در مقایسه با مارکوف فراهم می‌کند.

در حالی که هدف هر دو الگوریتم اخیر بهینه‌سازی تصمیمات بارگذاری

کارهای اضافه می‌باشد، نشان داده شده است که الگوریتم ژنتیکی تجمیع از الگوریتم ECCO از نظر مدیریت حجم کار، کاهش زمان پاسخ و مقیاس‌پذیری برای برنامه‌های IoT پیشی می‌گیرد. مقایسه بین الگوریتم ژنتیک تقویتی و فرایند تصمیم‌گیری مارکوف (MDP) برای مسئله مورد نظر (کنترل انرژی و اجرای محاسبات در شبکه‌های بی‌سیم- ECCO) وابسته به عوامل مختلف و خصوصیات خاص مسئله است.

در زیر برخی از مزایای الگوریتم ژنتیک تقویتی پیشنهادی نسبت به MDP که بهترین پاسخ را نسبت به بقیه الگوریتم‌های مورد مقایسه تولید کرده است آورده شده‌اند:

۱) **مدیریت فضای عمل پیوسته:** الگوریتم ژنتیک به طور طبیعی مسائلی با فضای عمل پیوسته را مدیریت می‌کند که در بسیاری از مسائل بهینه‌سازی مانند تصمیم‌گیری‌های اجرایی در شبکه‌های بی‌سیم به کار می‌رود. از سوی دیگر، فرایند تصمیم‌گیری مارکوف معمولاً با فضای عمل گسسته خوب کار می‌کند و نمایش اقدامات پیوسته ممکن است چالش‌برانگیز باشد.

۲) **بهینه‌سازی سراسری:** الگوریتم ژنتیک قابلیت بهینه‌سازی سراسری را دارد؛ زیرا یک جمعیت از راه‌حل‌ها را حفظ کرده و مناطق مختلف فضای جستجو را کاوش می‌کند. این مزیت در مسائل بهینه‌سازی پیچیده که به دنبال بهینه‌سازی گلوبال هستند، مفید است.

۳) **سادگی در نمایش مسئله:** نمایش راه‌حل‌ها به عنوان کروموزوم‌ها (ژن‌ها) در الگوریتم ژنتیک معمولاً ساده‌تر و آسان‌تر برای رمزگذاری برخی مسائل است؛ به ویژه زمانی که فضای راه‌حل پیچیده یا به‌خوبی ساختاریافته نیست.

٤) **تعادل بین کاوش و بهره‌برداری:** در برخی از سناریوها، الگوریتم ژنتیک تقویتی پیشنهادی می‌تواند تعادل بین کاوش و بهره‌برداری را به طور مؤثرتری نسبت به MDP برقرار کند. فرایند تصمیم‌گیری مارکوف معمولاً نیاز به تنظیم پارامترهای کاوش مانند سیاست‌های اپسیلون- خودخواهی دارد تا تعادل مناسبی به دست آید.

به هر حال لازم به ذکر است که هر دو الگوریتم ژنتیک تقویتی پیشنهادی و MDP دارای نقاط قوت و ضعف خود هستند و انتخاب بین این دو به ویژگی‌های خاص مسئله و الزامات آن وابسته است.

مزایای فرایند تصمیم‌گیری مارکوف:

۱) **محیط‌های پویا:** فرایند تصمیم‌گیری مارکوف برای مسائلی در محیط‌های پویا که اقدامات ممکن است منجر به حالت‌های مختلف با نتایج نامعلوم شوند، مناسب است و می‌تواند با فرایندهای تصادفی به‌خوبی کار کند.

۲) **سیاست بهینه:** فرایند تصمیم‌گیری مارکوف چارچوبی را برای یافتن سیاست بهینه ارائه می‌دهد که میزان پاداش‌های تجمعی در طول زمان را بیشینه می‌کند. این مزیت زمانی مفید است که هدف اصلی، یافتن بهترین سیاست باشد.

۳) **ارزش‌یابی:** فرایندهای تصمیم‌گیری مارکوف اغلب از الگوریتم‌های کارآمد مانند Iteration ارزش‌ها و Q-Learning برای یافتن سیاست‌های بهینه در فضای حالت‌های بزرگ استفاده می‌کنند که ممکن است در برخی مسائل بهینه‌سازی محاسباتی بهتر از الگوریتم ژنتیک باشد.

٤) **تضمین همگرایی:** فرایند تصمیم‌گیری مارکوف مزیت تضمین همگرایی به یک حل بهینه را دارد که با تعداد کافی از تکرارها و کاوش‌ها به دست می‌آید.

به اختصار، انتخاب بین الگوریتم ژنتیک و فرایند تصمیم‌گیری مارکوف برای مسئله P۱ وابسته به عواملی مانند طبیعت فضای اقدامات، پیچیدگی مسئله، پویایی محیط و اهداف بهینه‌سازی خاص است. معمولاً انتخاب نهایی بر اساس آزمایش و مقایسه‌ها برای محیط خاص مسئله تعیین می‌شود. با توجه به شرایط در نظر گرفته شده کنونی الگوریتم پیشنهادی مطابق با شکل‌های ۳ و ۴ در اکثر موارد عملکرد بهتری از خود نشان داده است.

٦- نتیجه‌گیری

با توجه به اهمیت روزافزون استفاده از اینترنت اشیا و محدودیت انرژی اشیا و لزوم پاسخگویی در بازه زمانی مورد قبول به درخواست‌های مطرح‌شده توسط کاربران نهایی، در این مقاله محیط واقعی به صورت فرموله با محدودیت‌های که باید در نظر گرفته شود، تحت عنوان مسئله P۱ به صورت فرموله نشان داده شده است. با توجه به محدودیت مصرف انرژی توسط اشیا، این مقاله با بارگذاری اجرای درخواست به لبه‌ها سبب بهبود مصرف انرژی و بالارفتن سرعت اجرا شد و برای اطمینان از به‌روزبودن تغییرات وضعیت نودها به لبه‌ها ارسال می‌شود که همه این موارد از ارسال و تازگی اطلاعات و زمان محاسبات در حل مسئله در نظر گرفته شده است. مواردی که این موضوع به‌موقع و یا با تأخیر انجام می‌شود به عنوان تنبیه یا پاداش در الگوریتم ژنتیک پیشنهادی لحاظ شد تا به‌صورت پویا برحسب شرایط جاری محیط جهت توزیع بار تصمیم‌گیری شود. در این معماری از سرورهای لبه با قابلیت‌های محاسباتی در لبه شبکه، به‌ویژه در ایستگاه‌های پایه استفاده شد. درخواست‌های حساس به تأخیر برحسب وضعیت جاری بار سیستم از طریق کانال‌های بی‌سیم به سرورهای لبه نزدیک منتقل شدند و در نتیجه ترافیک در شبکه مرکزی و تأخیر انتقال داده کاربر به‌ویژه برای برنامه‌های صنعتی با حجم داده زیاد، کاهش داده شد. هدف حل مسئله مدیریت منابع شبکه و انتقال محاسبات و کمینه‌سازی مصرف انرژی در دستگاه‌های اینترنت اشیا با تضمین تازگی داده‌های نودهای اینترنت اشیا بود. در این محیط‌ها شبکه و کارهای ورودی متغیر با زمان و در نتیجه پویا هستند؛ بنابراین در این مقاله الگوریتم ژنتیک تقویتی برای حل مسئله بهبود محیط پویا برای مسئله بارگذاری داده‌ها و کارها در این محیط پویا با در نظر گرفتن انرژی و انتقال محاسبات در عین حفظ تازگی داده‌ها ارائه شد. از جمله مواردی که مشخص شد الگوریتم ژنتیک تقویتی عملکرد خوبی دارد، عبارت هستند از مواقعی که فضای عمل پیوسته است؛ مانند شبکه‌های بی‌سیم. یکی از دلایل بهتر عمل‌کردن الگوریتم پیشنهادی نسبت به بهترین راهکار ارائه‌شده تاکنون همین بود؛ زیرا مثلاً از تصمیم‌گیری مارکوف استفاده شده بود که بیشتر مناسب محیط گسسته است. از جمله مزایای دیگر الگوریتم ژنتیک تقویتی پیشنهادی، استفاده از بهینه‌سازی سراسری و سادگی در نمایش مسئله و نهایتاً ایجاد تعادل بین کاوش و بهره‌برداری است که در این مسئله، الگوریتم ژنتیک تقویتی پیشنهادی توانست تعادل بین کاوش و بهره‌برداری را به طور مؤثرتری نسبت به فرایند تصمیم‌گیری مارکوف که معمولاً نیاز به تنظیم پارامترهای کاوش دارد از خود نشان دهد. برای پیشنهادهای آینده تصمیم داریم تا شرایط مسئله را برای محیط‌های واقعی دیگر موجود تغییر دهیم و الگوریتم‌های مختلف را برای حل آنها بررسی نماییم.

مراجع

[1] K. Peng, *et al.*, "Intelligent computation offloading and resource allocation in IIoT with end-edge-cloud computing using NSGA-III,"

IEEE Trans. on Network Science and Engineering, vol. 10, no. 5, pp. 3032-3046, Sept./Oct. 2022.
[2] J. Huang, H. Gao, S. Wan, and Y. Chen, "AoI-aware energy control and computation offloading for industrial IoT," *Future Generation Computer Systems*, vol. 139, pp. 29-37, Feb. 2023.
[3] H. Hu, T. Wang, and F. Feng, "Research on the deep deterministic policy algorithm based on the first-order inverted pendulum," *Applied Sciences*, vol. 13, no. 13, Article ID: 7594, 15 pp., 1 Jul. 2023.
[4] W. Huo, T. Zhao, F. Yang, and Y. Cheng, "An improved soft actor-critic based energy management strategy of fuel cell hybrid electric vehicle," *J. of Energy Storage*, *pt. A,* vol. 72, Article ID: 108243, Nov. 2023.
[5] R. Liu, C. Wang, A. Tang, Y. Zhang, and Q. Yu, "A twin delayed deep deterministic policy gradient-based energy management strategy for a battery-ultracapacitor electric vehicle considering driving condition recognition with learning vector quantization neural network," *J. of Energy Storage*, vol. 41, Article ID: 108147, Nov. 2023.
[6] M. Pan, Z. Li, and J. Qian, "Energy-efficient multiuser and multitask computation offloading optimization method," *Intelligent and Converged Networks*, vol. 4, no. 1, pp. 76-92, Mar. 2023.
[7] S. Tang, *et al.*, "Computational intelligence and deep learning for next-generation edge-enabled industrial IoT," *IEEE Trans. on Network Science and Engineering*, vol. 10, no. 5, pp. 2881-2893, Sept./Oct. 2022.
[8] Z. Wang, Y. Ding, X. Jin, Y. Chen, and C. Gao, "Task offloading for edge computing in industrial Internet with joint data compression and security protection," *The J. of Supercomputing*, vol. 79, pp. 4291-4317, 2023.
[9] J. Xu, B. Yang, Y. Liu, C. Chen, and X. Guan, "Joint task offloading and resource allocation for multihop Industrial Internet of Things," *IEEE Internet of Things J.*, vol. 9, no. 21, pp. 22022-22033, 1 Nov. 2022.
[10] J. Gao, *et al.*, "A task offloading algorithm for cloud-edge collaborative system based on Lyapunov optimization," *Cluster* Computing, vol. 26, no. 1, pp. 337-348, Feb. 2023.
[۱۱] س. ا. دشتی و ح. زارع، "افزایش کارایی الگوریتم تخلیه در محاسبات مه با کمک الگوریتم بهینه سازی ازدحام ذرات"، *فصلنامه روشهای هوشمند در صنعت برق*، سال ۱۶، شماره ۶۱، صص. ۹۶-۷۹، تابستان ۱۴۰۴.
[12] S. E. Dashti and A. M. Rahmani, "Dynamic VMs placement for energy efficiency by PSO in cloud computing," *J. of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 1-2, pp. 97-112, 2016.
[13] M. Sharma, *et al.*, "Enabling security for the Industrial Internet of Things using deep learning, blockchain, and coalitions," *Trans. on Emerging Telecommunications Technologies*, vol. 32, no. 7, Article ID: e4137, Jul. 2021.
[14] D. Jiang, Y. Wang, Z. Lv, W. Wang, and H. Wang, "An energy-efficient networking approach in cloud services for IIoT networks," *IEEE J. on Selected Areas in Communications*, vol. 38, no. 5, pp. 928-941, May 2020.
[15] H. Kurniawati, "Partially observable Markov decision processes and robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 253-277, 2022.
[16] Y. Song, *et al.*, "RL-GA: a reinforcement learning-based genetic algorithm for electromagnetic detection satellite scheduling problem," *Swarm and Evolutionary Computation*, vol. 77, Article ID: 101236, Mar. 2023.

سید ابراهیم دشتی تحصیلات خود را در مقاطع کارشناسی مهندسی کامپیوتر در یزد در سال ۱۳۷۷ شروع و در سال ۱۳۸۱ با عنوان دانشجوی برتر و فعال به پایان رسانید و کارشناسی ارشد خود را در گرایش نرم‌افزار در سال ۱۳۸۴ در اصفهان به پایان رسانید، در این مدت به عنوان مسئول فناوری اطلاعات مؤسسات مختلف از جمله بیمارستان نمازی و واحدهای مختلف دانشگاه آزاد اسلامی مختلف ایفای نقش نمود، نامبرده از سال ۱۳۸۴ به عنوان هیات علمی داری پست مدیری گروهی و مسئول فن آوری اطلاعات در دانشگاه آزاد اسلامی شروع به کار کرد. ایشان دکتری مهندسی کامپیوتر را در سال ۱۳۹۳ در دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران به پایان رسانید. هم‌اکنون دانشیار دانشگاه آزاد اسلامی واحد جهرم است. نامبرده دارای طرح های پژوهشی، تالیف و ترجمه کتابهای متعدد و بیش از صد مقاله در این سال‌ها بوده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: رایانش ابری، اینترنت اشیا، رایانش فراگیر، محاسبات نرم و کاربردهای آن، یادگیری ماشین، سیستم عامل و شبکه‌های بی‌سیم و سنسوری.

فاطمه مؤیدی در سال ۱۳۸۳ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه شیراز با کسب رتبه نخست دریافت نمود. در سال‌های ۱۳۸۷ و ۱۳۹۳ به ترتیب موفق به اخذ مدرک کارشناسی ارشد و دکتری در رشته مهندسی کامپیوتر با گرایش هوش مصنوعی از دانشگاه شیراز شد. از سال ۱۳۹۵ تاکنون به عنوان عضو هیأت علمی در مجتمع آموزش عالی لارستان مشغول به فعالیت می باشد. همچنین در برخی پروژه‌های تحقیقاتی با دانشگاه شیراز و علوم پزشکی شیراز همکاری داشته اند. زمینه‌های علمی مورد علاقه نامبرده عبارتند از: بینایی ماشین، پردازش تصویر، یادگیری ماشین، داده کاوی، شناسایی الگو و بازی‌های جدی.

عادل سالمی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی نرم‌افزار به ترتیب در سال‌های ۱۴۰۰ و ۱۴۰۳ از دانشگاه آزاد اسلامی واحد شیراز به پایان رسانده است. نامبرده دارای مدارک معتبر مایکروسافت و میکروتیک و سیسکو می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه‌های کامپیوتری، محاسبات ابری، سیستم‌های اینترنت اشیاء و سیستم‌های اینترنت اشیاء صنعتی.