

## توزیع بهینه بار کاری در پردازش‌های لبه شبکه بر پایه استفاده از سیستم‌های دسته‌بند یادگیر حافظه‌دار

مینا یعقوبی کیا<sup>۱\*</sup>، مهدی عباسی<sup>۲</sup>، میلاد رفیعی<sup>۳</sup>

\*نویسنده مسئول، دریافت: ۹۸/۱۲/۱۳، بازنگری: ۹۹/۰۲/۱۱، پذیرش: ۹۹/۰۳/۲۸

<sup>۱</sup> کارشناسی ارشد، مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران

<sup>۲</sup> استادیار، مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران

<sup>۳</sup> دانش‌آموخته کارشناسی ارشد، مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، ایران

### چکیده

همراه با رشد روزافزون دستگاه‌های هوشمند، مفهوم اینترنت اشیاء نیز توسعه پیدا کرده است. افزایش تعداد این اشیاء هوشمند سبب افزایش تولید حجم داده‌ها و بارهای محاسباتی در مقیاس‌های وسیع شده است. به همین دلیل رایانش ابری، به‌عنوان راه‌حل اصلی جهت کنترل این بارها استفاده می‌شود. با این حال، زمان بر بودن پردازش بارها در ابر، هنوز به‌عنوان مسئله اصلی در حوزه شبکه‌های توزیع‌شده مطرح است. پردازش بارهای کاری در لبه‌های شبکه می‌تواند موجب کاهش این زمان پاسخ شود؛ اما از سوی دیگر با آوردن پردازش بارها از مراکز داده‌ها (متصل به برق) به سمت لبه‌های شبکه، منجر به محدودیت انرژی می‌شود. بنابراین لازم است بارهای کاری به شکلی متوازن میان ابرها و لبه‌های شبکه توزیع شوند. در این مقاله به‌منظور ایجاد تعادل میان مصرف انرژی در لبه شبکه و تأخیر بارهای کاری در ابرها، روشی مبتنی بر سیستم‌های دسته‌بند یادگیر ارائه شده است. نتایج حاصل از شبیه‌سازی نشان می‌دهد که روش پیشنهادی در مقایسه با روش‌های پیشین، به شکل متعادل‌تری بارها را توزیع می‌کند. روش پیشنهادی سبب کاهش ۴۲ درصدی تأخیر بارهای کاری و همچنین کاهش مصرف انرژی در لبه شبکه می‌شود.

**کلمات کلیدی:** اینترنت اشیاء، پردازش لبه، یادگیری ماشین، انرژی تجدید پذیر

### ۱- مقدمه

با وارد شدن اینترنت اشیاء در زندگی روزمره ما و توسعه آن در بسیاری از حوزه‌ها و محیط‌ها، سبب بهبود کیفیت زندگی ما شده است. این فناوری در خانه، هنگام مسافرت، هنگام بیماری، در محل کار، هنگام دویدن و در ورزشگاه به کار بسته شده است. این محیط‌ها در ابتدا و بیشتر اوقات، هیچ‌گونه قابلیت ارتباطی نداشته‌اند. این فناوری امکان برقراری ارتباط اشیاء با یکدیگر و تشریح اطلاعات درک شده از محیط را فراهم کرده است. با در نظر گرفتن گوناگونی محیط‌های قرارگیری این اشیاء، طیف بسیار گسترده‌ای از برنامه‌های کاربردی در آن‌ها مستقر شده‌اند را می‌توان در حوزه‌های زیر دسته‌بندی کرد [۲]:

- حوزه حمل‌ونقل و تدارکات.
- حوزه بهداشت و درمان.
- حوزه محیط‌های هوشمند (خانه، اداره، کارخانه).
- حوزه شخصی و اجتماعی.

اینترنت اشیاء<sup>۱</sup> این امکان را فراهم می‌کند تا میلیاردها شیء شامل حسگرها<sup>۲</sup>، محرک‌ها<sup>۳</sup> و سایر اشیاء متصل به اینترنت، بتوانند از محیط یاد بگیرند و با آن در تعامل باشند [۱]. نمونه‌های اولیه این اشیاء شامل سیستم‌های کنترل گرمایشی، سرمایشی و تهویه هوا است که خانه را هوشمند می‌کند. حوزه‌های دیگری در اینترنت اشیاء نیز وجود دارد که می‌تواند کیفیت زندگی ما را ارتقا بخشد.

این فناوری به انسان‌ها و رایانه‌ها این امکان را می‌دهد که بتوانند از میلیاردها شیء که شامل حسگرها، محرک‌ها، خدمات و سایر اشیاء متصل به اینترنت است اطلاعات دریافت کنند و با هم در تعامل باشند [۱]. همچنین شرایطی فراهم می‌کند تا اشیاء فیزیکی نیز، اطلاعات خود را با یکدیگر به اشتراک بگذارند و در تصمیم‌گیری‌ها با یکدیگر هماهنگ باشند.

### • قابلیت جابجایی

از آنجاکه بیشتر خدمات به کاربران از طریق موبایل ارائه می‌شود قابلیت جابجایی به‌عنوان یک چالش مطرح می‌شود. در واقع قابلیت جابجایی اتصال دائم کاربران به خدمات موردنظر خود در هنگام حرکت است. چالش اصلی هنگام حرکت این دستگاه‌ها از یک دروازه به دروازه دیگر ایجاد می‌شود که دو راهکار کلی برای جلوگیری از قطع ارتباط وجود دارد [۷]: ذخیره‌سازی و تونل زنی. این دو روش به‌صورت موقتی، به برنامه‌ها امکان دسترسی به داده‌های اینترنت اشیاء را می‌دهد.

### • کارایی

همواره ارزیابی عملکرد خدمات اینترنت اشیاء یک چالش بزرگ محسوب می‌شود زیرا این معیار به مؤلفه‌ها زیادی بستگی دارد. سیستم‌های مبتنی بر اینترنت اشیاء مانند سایر سیستم‌ها برای برآورده کردن نیازهای مشتری باید خدمات خود را به‌طور مداوم توسعه داده و بهبود بخشد. دستگاه‌های اینترنت اشیاء باید تحت نظارت و ارزیابی قرار گیرند تا بهترین عملکرد ممکن را با قیمت مناسب برای مشتریان فراهم کنند. چند نمونه از معیارهایی که می‌توانند برای ارزیابی عملکرد اینترنت اشیاء استفاده شوند: سرعت پردازش، سرعت ارتباطات و هزینه دستگاه‌های ارتباطی است [۸].

### • مدیریت

اتصال میلیاردها یا تریلیون دستگاه هوشمند از ارائه‌دهندگان خدمات مدیران را با مسائل پیچیده‌ای از جمله خطایابی، پیکربندی، عملکرد و امنیت این دستگاه‌ها روبرو می‌کند؛ بنابراین لازم است پروتکل‌های مدیریتی سبک‌تری توسعه پیدا کنند تا مدیریت دستگاه‌های اینترنت اشیاء آسان‌تر شود [۹].

### • مقیاس‌پذیری

مقیاس‌پذیری اینترنت اشیاء به امکان اضافه کردن دستگاه‌ها، خدمات و عملکردهای جدید برای مشتریان اشاره دارد بدون اینکه روی کیفیت خدمات موجود تأثیر منفی بگذارد. به‌خصوص در صورت وجود سیستم‌عامل‌های سخت‌افزاری متنوع و پروتکل‌های ارتباطی مختلف افزودن وظایف جدید روی دستگاه‌ها کار آسانی نیست [۱۰]. به این منظور یک معماری سه لایه‌ای در [۱۱] ارائه شده که شامل لایه‌های: اشیاء مجازی<sup>۱۳</sup>، اشیاء مجازی ترکیبی<sup>۱۴</sup> و لایه خدمات است. ارائه این لایه‌ها با استفاده از خودکارسازی، هوشمندی در هر شیء، دو قابلیت مقیاس‌پذیری و همکاری در محیط اینترنت اشیاء را تضمین می‌کند.

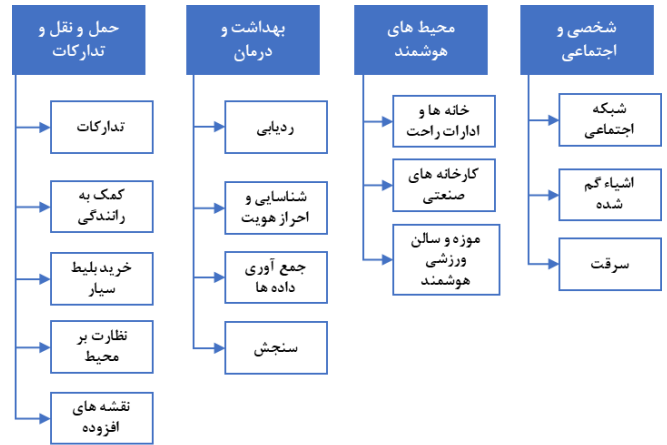
### • قابلیت همکاری

قابلیت همکاری انتها به انتها<sup>۱۵</sup>، یکی دیگر از چالش‌های اینترنت اشیاء است. این چالش به دلیل تنوع تعداد دستگاه‌ها و ناهمگونی سیستم‌عامل‌های مختلف آن‌هاست. توسعه‌دهندگان برنامه‌ها و دستگاه‌های اینترنت اشیاء باید همواره محصولات خود را مطابق با استانداردهای مشخص شده تولید کنند تا از ارائه خدمات برای همه مشتریان صرف‌نظر از مشخصات سیستم‌عامل و سخت‌افزاری که استفاده می‌کنند به‌درستی صورت گیرد. به‌عنوان مثال، امروزه بیشتر تلفن‌های هوشمند امروزه از فناوری‌های ارتباطی مشترکی مانند WiFi، NFC و GSM پشتیبانی می‌کنند تا قابلیت همکاری در سناریوهای مختلف را تضمین کنند [۱۲].

### • امنیت و حریم خصوصی

به دلیل عدم وجود استانداردها و معماری مناسب برای امنیت داده‌ها و دستگاه‌های اینترنت اشیاء، این ویژگی یک چالش مهم محسوب می‌شود. در شبکه‌های ناهمگونی مانند اینترنت اشیاء، تضمین امنیت و حفظ حریم شخصی کاربران آسان نیست. عملکرد اصلی اینترنت اشیاء بر مبنای تبادل اطلاعات بین میلیاردها یا حتی تریلیون اشیاء اتصال به اینترنت است. برای مثال توزیع نامناسب کلیدها بین دستگاه‌ها از مشکلاتی که امنیت اینترنت اشیاء را دچار چالش می‌کند و در استانداردها موردتوجه قرار نگرفته است [۱۳].

در میان این برنامه‌های کاربردی ممکن است فقط با برخی از آن‌ها به‌طور مستقیم تعامل داشته باشیم و برخی دیگر را فقط می‌توانیم ایده فرض کنیم؛ زیرا ممکن است برای استقرار و پیاده‌سازی آن‌ها هنوز فن‌آوری‌های کافی نداشته باشیم. در شکل ۱ به‌طور خلاصه مجموعه کلی از برنامه‌های کاربردی برای هر یک از این دسته‌ها را نشان داده‌ایم.



شکل ۱- حوزه‌های کاربردی و سناریوهای اصلی در اینترنت اشیاء

تحقق بخشیدن به فناوری اینترنت اشیاء سبب بروز چالش‌های بسیاری شده که برطرف کردن آن‌ها کار آسانی نیست. چالش‌های اصلی شامل دسترس‌پذیری<sup>۴</sup>، قابلیت اطمینان<sup>۵</sup>، قابلیت جابجایی<sup>۶</sup>، کارایی<sup>۷</sup>، مقیاس‌پذیری<sup>۸</sup>، قابلیت همکاری<sup>۹</sup>، امنیت<sup>۱۰</sup>، مدیریت<sup>۱۱</sup> و اعتماد<sup>۱۲</sup> است. بنابراین ارائه‌دهندگان خدمات اینترنت اشیاء و برنامه‌نویسان برنامه‌ها برای برطرف سازی این چالش‌ها، باید خدمات و برنامه‌های خود را به‌صورت کارآمد پیاده‌سازی کنند. به‌عنوان مثال، به دلیل حساس بودن اطلاعات شخصی کاربران، حفظ امنیت و حریم خصوصی آن‌ها نقش بسزایی در همه بازارهای جهانی دارند که باید برنامه‌نویسان برای تأمین این نیاز باید تمام نکات لازم را در نظر بگیرند [۳]. در ادامه مختصراً در مورد چالش‌های اساسی مراحل توسعه و استقرار اینترنت اشیاء می‌پردازیم.

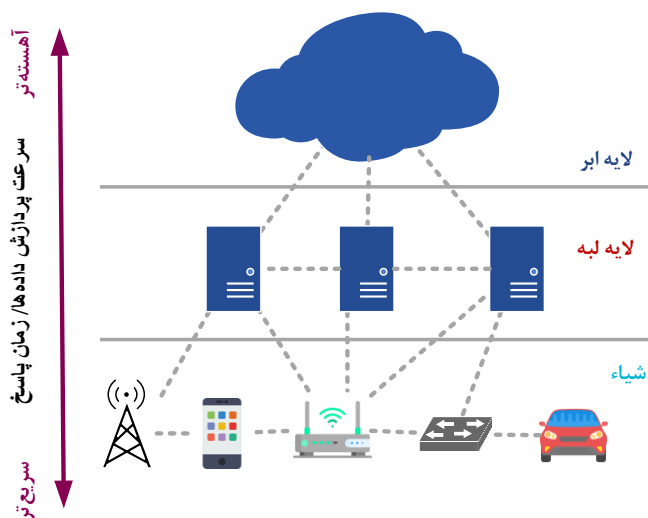
### • دسترس‌پذیری

در دسترس بودن اینترنت اشیاء به این معناست که این فناوری باید در دو سطح سخت‌افزاری و نرم‌افزاری قابل پیاده‌سازی باشد تا خدمات در هر زمان و هر مکانی برای مشتریان قابل‌ارائه باشد. در دسترس بودن نرم‌افزار یعنی برنامه‌های اینترنت اشیاء، خدمات را به همه افراد در مکان‌های مختلف به‌طور هم‌زمان ارائه دهد. در دسترس بودن سخت‌افزار یعنی دستگاه‌های موجود همیشه با عملکردها و پروتکل‌های اینترنت اشیاء سازگار باشند [۴].

### • قابلیت اطمینان

قابلیت اطمینان به کارکرد مناسب سیستم بر اساس ویژگی‌های آن گفته می‌شود. هدف از قابل‌اطمینان بودن یک سیستم، افزایش میزان موفقیت در ارائه خدمات اینترنت اشیاء است. این ویژگی با دسترس‌پذیر بودن رابطه بسیار نزدیکی دارد، زیرا با داشتن قابلیت اطمینان، دسترسی به اطلاعات و خدمات پس از گذشت مدتی تضمین می‌شود؛ اما زیرا در شرایط اضطراری که سیستم باید به‌طور دقیق و سریع به واکنش نشان دهد قابلیت اطمینان اهمیت بیشتری می‌یابد [۵]. این ویژگی باید به شکل نرم‌افزاری و سخت‌افزاری در تمام لایه‌های اینترنت اشیاء پیاده‌سازی شود. برای داشتن یک شبکه اینترنت اشیاء کارآمد، باید ارتباطات زیربنایی قابل‌اعتماد باشند، زیرا در صورت قابل‌اعتماد نبودن سیستم، فرایند جمع‌آوری، پردازش و انتقال داده‌ها می‌تواند منجر به تأخیرهای طولانی، از بین رفتن داده‌ها و درنهایت تصمیمات اشتباه شود و یک اتفاق به فاجعه‌بار روی دهد [۶].

این "دستگاه‌های لبه" و با پردازش درخواست‌ها به صورت محلی، نیازی به مداخله ابر نیست. لایه لبه بین دستگاه‌های انتهایی و ابر قرار گرفته و با کاهش تأخیر درخواست‌ها اجازه می‌دهد در سیستم‌های اینترنت اشیا حساس به تأخیر درخواست‌ها به شکل بلادرنگ پاسخ داده شود.



شکل ۲- لایه‌های پردازش داده‌ها در اینترنت اشیا

شکل ۲ لایه‌های پردازش داده‌ها را به طور کلی نشان می‌دهد. با توجه به شکل در پایین‌ترین لایه، دستگاه‌هایی قرار دارند که داده‌های قابل پردازش تولید می‌کنند. این دستگاه‌ها نیز می‌توانند از منابع پردازشی خود استفاده کرده و عملیات پردازشی را نزدیک به کاربران انجام دهند. با اینکه به دلیل نزدیک بودن به کاربران، درخواست‌ها تأخیر انتشار بسیار کمی خواهند داشت و با سرعت بیشتری پاسخ داده می‌شوند، قدرت پردازش کمتری نسبت به ابرها دارند. در لایه بالاتر مسیرهای پر قدرت و سرورهای نزدیک به لبه‌های شبکه قرار دارند که می‌توانند درخواست‌ها را بدون ارسال به ابر پردازش کنند [۱۸]. هر چه از اشیا به سمت مراکز داده‌ها حرکت می‌کنیم تأخیر انتشار بیشتر خواهد شد. در نهایت در بالاترین لایه، مراکز داده‌های عظیم به صورت ابرها در سراسر دنیا توزیع شده‌اند که قابلیت پردازش و ذخیره‌سازی بسیار بالایی دارند. با دور شدن این منابع از کاربران، تأخیر زیادی به درخواست‌ها تحمیل می‌شود [۱۹]. به علاوه این مراکز داده‌ها به علت حجم پردازش و قدرت بالا، انرژی الکتریکی به شدت زیادی را مصرف می‌کنند. در صورتی که اکثر دستگاه‌های لبه شبکه می‌توانند با باتری یا انرژی بسیار کمی کار کنند.

با این وجود تأمین برق لازم برای انجام این محاسبات بسیار پرهزینه و گاهی غیرممکن است. زیرا این دستگاه‌ها به صورت دوره‌ای برای شارژ باتری به برق وصل شوند، و شارژ شدن آن‌ها با برق از تحرک آن‌ها جلوگیری می‌کند. از سوی دیگر ممکن است تعویض باتری دستگاه‌های اینترنت اشیا روشی جایگزین باشد اما معمولاً هزینه بالایی دارد و حتی ممکن است خطراً فرین باشد. به همین جهت بهتر است دستگاه‌های اینترنت اشیا با استفاده از انرژی سبز و قابلیت شارژ بی‌سیم استقلال و پایداری خود را حفظ کنند [۲۰]. بنابراین گره‌های محاسباتی لبه اغلب انرژی مورد نیاز خود را از طریق باتری‌های قابل شارژ تأمین می‌کنند [۲۱]. چون این نوع منبع انرژی بسیار محدود است و پس از مدتی باید مجدداً شارژ شود؛ استفاده از انرژی‌های تجدید پذیر، به عنوان یک منبع تغذیه اصلی یا حتی تنها منبع انرژی در لبه شبکه الزامی است. در این راستا در این پژوهش به دلیل کمبود انرژی در لبه شبکه، با استفاده از انرژی‌های تجدید پذیر تلاش می‌شود باتری‌ها در لبه شبکه به حداکثر میزان خود برسند.

اتصال تعداد زیادی اشیا فیزیکی به اینترنت سبب تولید حجم زیادی از داده‌ها می‌شود. به این ترتیب برای ذخیره، پردازش و بازیابی داده‌های بزرگ به مکانیسم‌های هوشمند و کارآمدی نیاز است؛ اما تعداد این داده‌ها به حدی است که از توانایی‌های سخت‌افزاری و نرم‌افزاری ابزارهای رایج برای مدیریت و پردازش آن‌ها در مدت زمان قابل قبول کافی نیست. این داده‌های قابل پردازشی موجب بروز محدودیت منابع محاسباتی و ذخیره‌سازی می‌شوند [۱۴]. در نتیجه برای جمع‌آوری و پردازش داده‌هایی که توسط این اشیا تولید می‌شوند، معماری و فناوری‌های متعددی ارائه شده است که ارائه خدمات اینترنت اشیا به کاربران را بهبود می‌بخشد.

رایانش ابری<sup>۱۶</sup> یک مکانیسم جدید مدیریتی برای پردازش حجم زیادی از داده‌ها را ارائه می‌دهد [۳]. پایه و اساس رایانش ابری مبتنی بر مراکز داده‌ها<sup>۱۷</sup> است که دارای قدرت ذخیره‌سازی و پردازش داده‌ها در مقیاس بزرگ هستند. این مراکز داده که غالباً از طریق شبکه‌های نوری با یکدیگر در ارتباط هستند، می‌توانند به عنوان یک منبع واحد برای کاربر نهایی ظاهر شوند. ظهور رایانش ابری محدودیت فضای ذخیره‌سازی و پردازشی برطرف گردید. [۱۵]. اما طی دهه گذشته انتقال بارهای محاسباتی و حجم داده‌ها در ابرها روند صعودی داشته است. نتایج این محاسبات و داده‌های کنترلی دائماً در مراکز داده‌های متمرکز و هسته‌های شبکه جابجا می‌شوند [۱۶]. در نتیجه، رایانش ابری با چالش‌های جدیدی روبرو می‌شود:

۱. **تأخیر:** ارسال داده‌ها از دستگاه‌های اینترنت اشیا به سمت مراکز داده‌ها، به دلیل طولانی بودن فاصله، تأخیر زیادی به بارهای کاری تحمیل می‌کند.
  ۲. **محدودیت پهنای باند:** با افزایش سریع تعداد اشیا متصل به اینترنت، سرعت تولید داده‌ها نیز با نرخ نمایی رشد پیدا می‌کند. به عنوان مثال یک ماشین می‌تواند در هر ثانیه چند مگابایت داده تولید کند. بنابراین افزایش نرخ ارسال و حجم داده‌های تولید شده، در لینک‌های ارتباطی محدودیت پهنای باند ایجاد می‌کند.
  ۳. **محدودیت منابع دستگاه‌ها:** بیشتر دستگاه‌های اینترنت اشیا منابع بسیار محدودی دارند. به عنوان مثال حس گر، محرک‌ها، کنترل کننده‌ها، دوربین‌های مدار بسته و... به دلیل محدودیت منابع قادر به تکیه بر خود نیستند تا تمام نیازهای محاسباتی خود را برآورده سازند. از سوی دیگر وابستگی دائم این دستگاه‌ها به ابر بسیار هزینه‌بر است، زیرا اغلب این تعاملات نیاز به منابع پردازشی و پروتکل‌های پیچیده دارد.
  ۴. **امنیت:** دستگاه‌ها برای دستیابی به اطلاعات ذخیره شده یا منابع پردازشی در ابر لازم است شناسایی و احراز هویت شوند. از سوی دیگر سایر افراد یا دستگاه‌ها نباید به این منابع دسترسی داشته باشند یا پیام‌ها را شنود کنند. مراکز داده‌هایی که در قلب این ابرها قرار دارند غالباً از طریق شبکه‌های نوری با یکدیگر در ارتباط هستند که تأخیر بسیار پایین آن‌ها سبب شده تا شبکه‌های مرکز داده (DCN)<sup>۱۸</sup> به عنوان یک منبع واحد برای کاربران ظاهر شوند. با این حال، برخی از حوزه‌های سیستم‌های اینترنت اشیا نیاز است که پردازش داده‌های حسگرها به صورت بلادرنگ انجام شود تا محرک‌ها سریع تصمیم را اعمال کنند. پس حتی اگر ارتباطات داخلی DCN تأخیر بسیار کمی داشته باشند، تأخیر ارتباط بین دستگاه‌های انتهایی و DCN‌ها برای سیستم‌های اینترنت اشیا بلادرنگ یک گلوگاه ایجاد می‌کند [۱۵]. به این ترتیب لازم است پردازش‌ها نزدیک کاربران نهایی انجام شود تا محدودیت سیستم‌های اینترنت اشیا حساس به تأخیر، برطرف یا تعدیل شود.
- سرانجام مدل جدیدی به نام رایانش لبه<sup>۱۹</sup> ارائه شد تا پردازش داده‌ها را با تأخیر کم و نزدیک به کاربران در لبه شبکه، امکان پذیر کند. هدف اصلی از ارائه این مدل، آوردن منابع محاسباتی و ذخیره‌سازی نزدیک به کاربر نهایی در شبکه است [۱۷]. محاسبات لبه با به کارگیری ظرفیت ذخیره‌سازی و پردازشی تعداد زیادی دستگاه اینترنت اشیا، یک لایه میانی بین دستگاه‌های انتهایی و ابر ایجاد می‌کند. با حضور

شبکه معرفی می‌گردد. در بخش چهارم روش پیشنهادی پیاده‌سازی شده و نتایج به‌دست‌آمده با توجه به معیارهای ارزیابی بررسی و مقایسه می‌شود. در انتها نتیجه‌گیری و ایده‌هایی جهت ادامه این پژوهش ارائه شده است.

## ۲- کارهای پیشین

در سال‌های اخیر (به‌خصوص از ۲۰۱۶ به بعد) پژوهش‌های متعددی در زمینه‌ی توزیع بار محاسباتی در لبه‌های شبکه صورت گرفته است. در ادامه تعدادی از کارهای انجام‌شده در حوزه بهبود مصرف انرژی و هزینه پردازش در رایانش مه‌آلود بررسی شده است:

اولین پژوهش‌ها در سال ۲۰۱۶، امکان ترکیب انرژی‌های تجدید پذیر در محاسبات لبه‌های شبکه تلفن همراه را بررسی می‌کنند [۲۷]. این پژوهش‌ها، چند الگوریتم یادگیری، برای مدیریت منابع و تخصیص بار محاسباتی ارائه می‌دهند. یکی از الگوریتم‌های مذکور، سیاست تخلیه‌ی بارهای ۲۰ کاری پویا (به ابر متمرکز) و تهیه کارگزار لبه را یاد می‌گیرد. هدف از ارائه این الگوریتم، به حداقل رساندن هزینه‌های درازمدت سیستم (شامل هر دو تأخیر سرویس و هزینه‌های عملیاتی) است. الگوریتم مذکور، سبب بهبود میزان یادگیری و عملکرد زمان اجرا آن الگوریتم، در مقایسه با الگوریتم‌های دیگر می‌شود. نتایجی حاصل شبیه‌سازی نشان می‌دهد که میانگین مصرف انرژی این الگوریتم نسبت به الگوریتم‌های مشابه ۲۵٪ کاهش پیدا می‌کند، اما همچنان سرعت یادگیری آن پایین می‌ماند.

در این مقطع زمانی باوجود ارائه روش‌هایی برای بهبود مصرف انرژی، همچنان سرعت روش‌های پیشنهادی، پایین بود. پس در پژوهش‌های آتی لازم بود تا علاوه بر مصرف انرژی به هزینه زمانی (سرعت) روش‌های ارائه‌شده، توجه بیشتری شود. در نتیجه در سال ۲۰۱۷ آقای جی زو<sup>۲۱</sup> و همکارانش، الگوریتم یادگیری پیشین خود را با تغییر مدل‌های یادگیری، بهبود بخشیدند [۲۸]. آن‌ها برای یادگیری سریع، یک الگوریتم یادگیری تقویتی مبتنی بر PDS<sup>۲۲</sup> ارائه دادند، تا تخلیه بار و خودکارسازی تخصیص منابع به‌صورت بهینه انجام شود. الگوریتم مذکور، حتی با پارامترهای ناشناخته سیستم هم به‌درستی کار می‌کرد. آن‌ها سپس، الگوریتم پیشنهادی خود را، هم به‌صورت شبیه‌سازی و هم در محیط واقعی، ارزیابی کردند. این نتایج نشان می‌داد که طرح پیشنهادی می‌تواند به‌طور قابل‌توجهی، موجب بهبود عملکرد محاسبات لبه شود. تخصیص بارکاری متناسب با منابع انرژی تجدید پذیر غیرقابل‌پیش‌بینی، از نقاط قوت این الگوریتم است.

الگوریتم فوق باوجوداینکه توانست مصرف انرژی بارهای محاسباتی در لبه‌های شبکه را بهبود بخشد، اما نمی‌توانست این بارهای کاری را به‌صورت متعادل در میان گره‌های پردازشی توزیع کند.

در سال ۲۰۱۸ هانگ وو<sup>۲۳</sup> و همکارانش [۲۱]، الگوریتمی به نام GLOBE را ارائه دادند. آن‌ها در این الگوریتم، دو مکانیسم تعادل بار جغرافیایی و کنترل ورودی بارها را برای بهینه‌سازی عملکرد ایستگاه‌های لبه شبکه، باهم ترکیب کردند. الگوریتم ذکرشده بدون نیاز به اطلاعات آینده سیستم، به‌صورت آنلاین عمل می‌کند و به چالش‌های مهمی ناشی از وضعیت باتری و محدودیت‌های انرژی را پاسخ می‌دهد. نتایج شبیه‌سازی‌ها نشان می‌دهد: الف) GLOBE در مقایسه با الگوریتم‌های آفلاین دیگر هزینه را تا ۵۰٪ کاهش می‌دهد (منظور از الگوریتم آفلاین، الگوریتم‌هایی است که اطلاعات کاملی در مورد وضعیت آینده دارند). ب) این الگوریتم یک تعادل بین ظرفیت باتری و عملکرد سیستم ایجاد می‌کند. الگوریتم ذکرشده توزیع بارهای محاسباتی در میان گره‌های شبکه را بهبود بخشید، اما این متعادل‌سازی هنوز به حالت بهینه نرسیده بود.

با توجه به نقایص موجود در کار آقای هانگ در سال ۲۰۱۸ آقای جیافو وان<sup>۲۴</sup> و همکارانش [۲۹]، یک روش متعادل‌سازی و برنامه‌ریزی بار برای بهبود مصرف انرژی مبتنی بر رایانش مه‌آلود را پیشنهاد می‌کنند. آن‌ها ابتدا، فرمول‌هایی برای

همان‌طور که در مطالب فوق ذکر شد، لازم است تصمیم‌گیری شود به هر لایه چه میزان حجم کار تخصیص یابد. تصمیم‌گیری‌ها بر اساس معیارهایی صورت می‌گیرد که می‌تواند این تخصیص منابع و توزیع بارکاری را بهبود دهد [۲۲]. این معیارها عبارت‌اند از:

۱- **تأخیر:** هرچند تأخیر فقط با زمان محاسبات اندازه‌گیری نمی‌شود، برای کاهش تأخیر حجم بار بهتر است این محاسبات در نزدیک‌ترین لایه که قدرت محاسباتی کافی دارد انجام شود.

۲- **پهنای باند:** عمل پیش‌پردازش داده‌ها قبل از ارسال به شبکه سبب کاهش حجم داده و در نتیجه سبب کاهش مصرف پهنای باند می‌شود.

۳- **انرژی:** در لبه‌ی شبکه لازم است یک مصالحه بین انرژی مصرفی پردازش داده‌ها و انرژی مصرفی برای ارسال آن به نقطه‌ای دیگر ایجاد شود.

۴- **هزینه:** سرمایه‌گذار مسئول ارائه خدمات، ساخت و نگهداری اشیاء در هر لایه است. ارائه یک مدل جدید برای هزینه، باید تضمین کند که برای ارائه‌دهنده خدمات سودآور است و برای کاربران نیز مقبولیت به همراه می‌آورد.

در انتها می‌توان نتیجه گرفت که با افزایش سطح خدمات‌رسانی در لبه شبکه، درخواست‌ها می‌توانند زمان پاسخ کوتاه‌تری داشته باشند [۲۳]. به‌علاوه، با پردازش محلی داده‌ها (به‌جای بارگذاری در ابر) در مصرف پهنای باند نیز می‌توان صرفه‌جویی کرد. اما از سوی دیگر این افزایش پردازش محلی داده‌ها، لبه شبکه را با محدودیت انرژی و پهنای باند روبرو می‌کند. بنابراین لازم است تا تعادلی میان پردازش محلی و ارسال داده‌ها به ابر صورت گیرد تا علاوه بر کاهش تأخیر داده‌ها، لبه شبکه با محدودیت انرژی روبرو نشود [۲۴]. پس به دلیل محدود بودن منابع تولید انرژی در لبه‌های شبکه، نمی‌توان کل بارهای ورودی را به‌صورت محلی پردازش کرد و باید بخشی از بار ورودی را به سمت ابر ارسال نمود [۲۵]. از طرفی در صورت ارسال بیش‌ازحد بارها به ابر، شاهد افزایش تأخیر بارهای کاری شده و کارایی سیستم‌های حساس به تأخیر کاهش می‌یابد و حتی عملکرد برخی دچار اختلال می‌شود.

بنابراین تخصیص بار، کار آسانی نیست زیرا معیارهای تخصیص منابع (انرژی، تأخیر، پهنای باند و هزینه) بسیار نزدیک و وابسته به یکدیگرند. پس باید معیارهایی برای بارهای کاری مختلف تعیین کرده، سپس آن‌ها را اولویت‌بندی کرد تا بتوان یک استراتژی تخصیص متناسب با معیارها، انتخاب کرد؛ بنابراین سه معیار تأخیر، انرژی و هزینه در استفاده از منابع توسط بارهای کاری هم‌زمان، باید مدنظر قرار گرفته شود [۲۶] که در ادامه این پژوهش سعی در برطرف سازی این چالش داریم.

در این پژوهش برای اولین بار جهت به دست آوردن حالت بهینه در متعادل‌سازی بارهای کاری از سیستم‌های دسته‌بند یادگیر استفاده شده است. سیستم‌های دسته‌بند یادگیر برای هدایت جستجو به سمت حالت بهتر، از یک الگوریتم تکاملی جهت جستجو در فضای حالت‌های موجود و از فرآیند یادگیری دیگری برای انتخاب استفاده می‌کنند. علاوه بر این، در این پژوهش با اضافه کردن حافظه‌ای که حالت‌های بهینه را نگهداری می‌کند، سبب بهبود عملکرد سیستم‌های دسته‌بند یادگیر شده است.

در روش پیشنهادی، بارهای کاری به‌گونه‌ای توزیع می‌شود که میزان تأخیر زمان محاسبات و تأخیر ارسال بارها بین ابر و لبه شبکه به حداقل خود می‌رسد. همچنین در روش پیشنهادی میزان مصرف انرژی بهینه‌شده است؛ به‌صورتی که با کنترل میزان استفاده از انرژی تجدید پذیر در بلندمدت سبب شارژ باتری می‌شود. با مقایسه دو معیار تأخیر بارها و میزان انرژی مصرفی در روش ارائه‌شده و کارهای اخیر، روش پیشنهادی توانسته است هر دو معیار را به‌صورت بهینه متعادل سازد.

ادامه این مقاله به‌صورت زیر سازمان‌دهی شده است: در بخش دوم کارهای مرتبط به موضوع ذکر می‌شود. سپس در بخش بعد روشی جدید برای توزیع بارها ارائه می‌شود که شامل مدل‌سازی سیستم و فرموله‌سازی مسئله است. همچنین در این بخش سیستم دسته‌بند یادگیر تغییر یافته جهت متعادل‌سازی توزیع بار در لبه

### ۳-۱- مدل سازی سیستم

برای سناریوی اصلی، یک سیستم لبه متشکل از یک ایستگاه پایه و مجموعه‌ای از سرورهای لبه در نظر گرفته شده که از نظر فیزیکی در کنار هم قرار دارند. هر یک از این منابع پردازشی در لبه‌ی شبکه (سرورهای لبه) دارای باتری با ظرفیت محدود هستند. پس یک منبع تغذیه مشترک در شبکه قرار داده شده تا در صورت کمبود باتری در سرورهای لبه، بارهای کاری به ابر ارسال شوند. بارکاری که از سمت کاربران به لبه شبکه وارد می‌شود ابتدا ایستگاه پایه آن‌ها را دریافت می‌کند. وظیفه ایستگاه پایه تصمیم‌گیری درباره میزان بار تخصیصی به لبه شبکه و ارسال باقیمانده آن به ابر است. شکل ۳ معماری کلی سیستم را نشان می‌دهد. همچنین در جدول ۱ پارامترهای اصلی معرفی شده‌اند.

جدول ۱- نمادهای اصلی

معنا	نماد
نرخ کل بار ورودی	$\lambda(t)$
میزان بارکاری که به صورت محلی پردازش می‌شود	$\mu(t)$
تعداد سرورهای فعال در لبه شبکه	$m(t)$
هزینه تأخیر پردازش بارهای کاری در لبه شبکه	$c_{io}(t)$
هزینه تأخیر ارسال بارهای کاری به ابر	$c_{off}(t)$
میزان انرژی تجدید پذیر دریافت شده	$g(t)$
وضعیت ازدحام در شبکه	$h(t)$
هزینه کل تأخیرها	$c_{delay}(t)$
هزینه استفاده از منبع تغذیه پشتیبان	$c_{back}(t)$
مصرف انرژی کارهای عملیاتی در لبه شبکه	$d_{op}(t)$
مصرف انرژی پردازش بارها در لبه شبکه	$c_{comp}(t)$
کل انرژی مصرف شده	$d(t)$
سطح باتری در لبه شبکه	$b(t)$
وضعیت سیستم	$s(t)$

در این پژوهش سیستم پیشنهادی از چهار منظر مدل می‌شوند:

#### ۳-۱-۱- مدل بارکاری

مدل زمانی در این سناریو، گسسته و با بازه‌های زمانی یکسان فرض می‌شود که به صورت  $t = 0, 1, 2, \dots$  نشان داده خواهد شد. در هر بازه زمانی دستگاه‌های لبه باید ظرفیت محاسباتی خود را تنظیم کنند (تعداد سرورهای فعال تعیین شوند).  $\lambda(t)$  نرخ رسیدن کل بارکاری در لحظه  $t$  به لبه شبکه را نشان می‌دهد که  $\lambda(t) \in [0, \lambda_{max}]$  است ( $\lambda_{max}$  حداکثر بار ورودی ممکن به لبه شبکه است). سپس سیستم تصمیم می‌گیرد، چه مقدار بارکاری به صورت محلی پردازش خواهد شد که آن را با پارامتر  $\mu(t)$  نشان داده می‌شود و  $\mu(t) \leq \lambda(t)$  است. در نهایت مقدار بارکاری  $\lambda(t) - \mu(t)$  به سمت ابر ارسال می‌شوند. همچنین تعداد سرورهای فعال در هر بازه زمانی  $m(t) \in [0, M]$  است ( $M$  حداکثر تعداد سرورها در لبه شبکه است). نکته: تعداد این سرورها در طول هر بازه زمانی ثابت است و تغییر نمی‌کند اما ممکن است برای بازه زمانی بعدی تغییر کند. شکل ۴ فرایند رسیدن بارهای کاری به لبه شبکه را نشان می‌دهد.

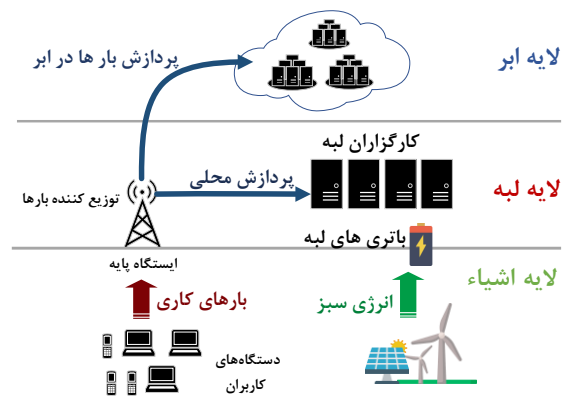
مدل مصرف انرژی مربوط به حجم کاربر روی گره‌های مه، ارائه می‌دهند، سپس یک تابع بهینه‌سازی باهدف متعادل سازی بار به صورت خوشه‌ای به دست می‌آورند. آن‌ها تابع مذکور را، با الگوریتم بهینه‌سازی دسته‌ای بهبود می‌بخشند و در نهایت یک سیستم چندمنظوره برای رسیدن به زمان بندی توزیع خوشه‌ای ارائه می‌کنند. با آزمایش روش پیشنهاد شده بر روی ربات‌ها و raspberry pi ها به دو می‌رسند: الف) در بستر ایجاد شده کارایی ربات‌ها نسبت به سایر روش‌های زمان بندی بهبود یافته، ب) بارهای کاری به صورت متوازن تری در میان آن‌ها توزیع شده است. در نتیجه چون مصرف انرژی تابعی از میزان بارکاری بر روی هر گره است؛ مصرف انرژی نیز به صورت متعادل تری در میان گره‌ها تقسیم خواهد شد.

در سال ۲۰۱۸ آقای زنگ<sup>۲۵</sup> و همکارانش [۳۰]، به بررسی چگونگی تولید انرژی‌های متفاوت در محاسبات مه‌آلود پرداختند تا به یک ساختار کارآمد برای ارائه سرویس انرژی برسند. آن‌ها شبکه مه را مبتنی بر CPS<sup>۲۶</sup> همراه با انرژی سبز طراحی کردند و سیستم مذکور را با توجه به کنترل نرخ منابع، متعادل ساز بار و سرویس‌ها توسعه دادند. آن‌ها مسئله بهره‌وری انرژی را به صورت فرمولی از برنامه‌ریزی خطی عدد صحیح مختلط ارائه دادند و ثابت کردند که یک مسئله NPhard است. سپس برای حل این پیچیدگی محاسباتی، یک پیشنهادی الگوریتم اکتشافی پیشنهاد کردند. نتایج حاصل از شبیه‌سازی نشان می‌دهد که بهره‌وری انرژی این الگوریتم، نزدیک به راه‌حل بهینه است.

در پژوهش‌های انجام شده، هیچ‌یک برای توزیع حجم بارکاری در میان گره‌های شبکه، میان دو معیار هزینه و مصرف انرژی تعادلی ایجاد نکرده‌اند. در این راستا در این پژوهش مکانیسمی جهت توزیع بارهای کاری میان ابر و گره‌های لبه به صورت متعادل ارائه شده است. این مکانیسم باهدف ایجاد تعادل میان انرژی مصرفی و تأخیر بارهای کاری، عمل می‌کند و از سوی دیگر از منابع انرژی تجدید پذیر به عنوان منبع تأمین انرژی محاسبات در لبه شبکه استفاده می‌کند. انتظار می‌رود که این منابع بتوانند باتری‌ها را در حالت شارژ نگه‌دارند.

### ۳- روش پیشنهادی

با توجه به بخش قبل، در روش‌های ذکر شده ایراداتی وجود داشت که ما را بر آن داشت تا روشی برای بهبود تخصیص تأخیر بارهای کاری و متعادل سازی مصرف انرژی در لبه شبکه ارائه دهیم. در این بخش ابتدا با استفاده از مدل ارائه شده در مقاله [۲۸] یک سیستم مبتنی بر رایانش مه، شبیه‌سازی می‌شود. این سیستم از چهار منظر مدل شده است که شامل: مدل بارکاری، مدل تأخیر، مدل مصرف انرژی و مدل باتری است. برای هر یک از این مدل‌ها فرمول‌هایی با توجه به مقاله مرجع ارائه می‌شود. سپس سیستمی مبتنی بر یادگیری تقویتی ارائه داده و با انطباق این سیستم با مدل پیشنهادی، سناریوی موردنظر شبیه‌سازی شده است.



شکل ۳- معماری سیستم لبه همراه با انرژی قابل تجدید

می‌آید که طبق رابطه ی  $d_{op}(\lambda(t)) = d_{sta} + d_{dyn}(\lambda(t))$  بدست می‌آید.

بخش اول ( $d_{sta}$ ) انرژی مصرفی ثابت در لبه شبکه است. بخش دوم ( $d_{dyn}(\lambda(t))$ ) انرژی مصرفی متغیر در لبه شبکه است که بر اساس بار ورودی به شبکه تغییر می‌کند که در این مدل به دلیل نزدیک بودن مراکز به هم، صفر در نظر گرفته می‌شود.

- انرژی لازم برای محاسبات بارهای کاری که قرار است در لبه شبکه پردازش شود با  $d_{com}(t)$  نمایش داده می‌شود. این انرژی بر اساس میزان بار داده‌شده به لبه شبکه ( $\mu(t)$ ) و تعداد سرورهای موجود در لبه شبکه ( $m(t)$ ) محاسبه خواهد شد. این انرژی طبق رابطه  $d_{com}(t) = m(t) * d_{fog-sever}$  قابل محاسبه است.

درنهایت کل انرژی مصرفی، از جمع دو انرژی بالا به دست خواهد آمد:

$$d(\lambda(t), \mu(t), m(t)) = d_{op}(\lambda(t)) + d_{com}(\mu(t), m(t)) \quad (2)$$

در این مدل انرژی‌های تجدید پذیر مثل انرژی باد و انرژی خورشیدی به‌عنوان یک منبع تولید انرژی استفاده می‌شود که با  $g(t)$  نمایش می‌دهند. این انرژی متناسب با وضعیت محیط ممکن است تغییر کند.

پیش‌بینی‌شده است تا بیش از ۵۰ میلیارد دستگاه تا سال ۲۰۲۰ به اینترنت متصل باشند. اتصال این دستگاه‌ها به اینترنت مقدار بی‌سابقه‌ای انرژی مصرف می‌کند. همچنین تخمین زده می‌شود که میزان مصرف انرژی جهانی دستگاه‌های لبه در اینترنت اشیاء تا سال ۲۰۲۵ به ۴۶TWh برسد که برابر با کل مصرف برق سالانه پرتغال در سال ۲۰۱۵ است. از سمت دیگر با کاهش منابع انرژی ناپذیر کره زمین و تأثیرات مخرب ناشی از انتشار گازهای کربن، استفاده از انرژی سبز (تجدید پذیر) برای مصرف دستگاه‌ها، مطابق با نیاز جهانی اساسی است [۲۰]. بنابراین لازم است در پژوهش‌ها این معیار در نظر گرفته شود که برای این پژوهش میزان انرژی تجدید پذیر از [۳۱] استفاده می‌کند.

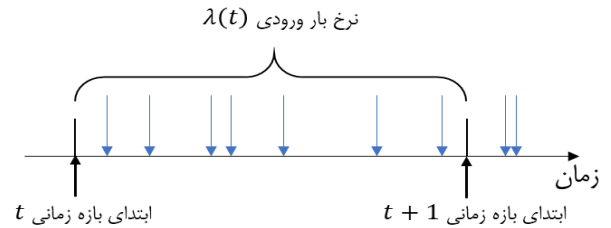
#### ۳-۱-۴- مدل باتری

همان‌طور که قبلاً ذکر شده برای هر سرور لبه یک باتری با ظرفیت محدود در نظر گرفته شده است. مجموعاً کل باتری‌های موجود در لبه شبکه  $b(t) \in [0, B]$  در نظر گرفته می‌شود که  $B$  حداکثر ظرفیت باتری در لبه شبکه است. این باتری‌ها توسط انرژی‌های خورشیدی و باد شارژ می‌شوند. در ابتدا میزان باتری صفر تنظیم شده است. از آنجا که لازم است سطح باتری در لبه شبکه کنترل شده باشد باید میزان بار در لبه  $\mu(t)$  کنترل شود. این کنترل میزان بار با شرایط زیر بررسی خواهد شد:

- هنگامی که  $b(t) \leq d_{op}(\lambda(t))$  نباید پردازشی در لبه شبکه صورت پذیرد. چون باتری موجود در لبه نمی‌تواند انرژی موردنیاز پایه را پشتیبانی می‌کند، باید همه بار  $\lambda(t)$  به ابر ارسال شود تا باتری فرصت شارژ پیدا کند. در این صورت باید از منبع تغذیه پشتیبان برای ارسال بارها به ابر استفاده شود. هزینه این عملیات از رابطه  $c_{bak}(\lambda(t)) = \varphi \cdot d_{op}(\lambda(t))$  به دست می‌آید. که  $\varphi$  ضریب هزینه منبع تغذیه پشتیبان است. برای بازه زمانی بعدی سطح انرژی باتری از طریق فرمول  $b(t+1) = b(t) + g(t)$  محاسبه خواهد شد و انرژی تجدید پذیر باعث شارژ باتری می‌شود.

حالت اول شکل ۵ این موضوع را به تصویر بیان می‌کند.

- اگر میزان باتری بیشتر از انرژی مصرفی پایه باشد می‌توان بخشی از بارهای کاری را ( $\mu(t)$ ) در لبه شبکه پردازش کرد و باید باقی‌مانده را به



شکل ۴- نحوه ورود بارهای کاری در یک بازه زمانی

#### ۳-۱-۲- مدل تأخیر

این سیستم سه دسته تأخیر خواهد داشت:

- تأخیر انتقال بارها در شبکه بی‌سیم که آن با  $c_{wi}(t)$  نشان داده شده و بر اساس میزان بار ورودی به شبکه (یعنی  $\lambda(t)$ ) به دست می‌آید. اما در این سناریو به دلیل نزدیکی فیزیکی گره‌های پردازشی صفر در نظر گرفته می‌شود.
- تأخیر پردازش بارهای کاری به‌صورت محلی در لبه شبکه با  $c_{lo}(t)$  نمایش داده می‌شود. از آنجا که این معیار بر اساس پارامترهای  $m(t)$  و  $\mu(t)$  محاسبه می‌شود به‌صورت  $(c_{lo}\mu(t), m(t))$  نیز نمایش داده خواهد شد. شیوه محاسبه این تأخیر به مکانیسم مدیریت صف در سرورهای لبه بستگی دارد، که در این مدل با توجه به مقاله مرجع از سیستم مدیریت صف M/G/1 استفاده شده است. در نتیجه تأخیر پردازش در لبه شبکه از رابطه  $c_{lo}(\mu(t), m(t)) = \frac{\mu(t)}{m(t) \cdot k - \mu(t)}$  به دست می‌آید که  $k$  ظرفیت پردازش هر سرور برحسب تعداد درخواست‌ها برثانیه است.
- تأخیر ارسال باقیمانده بارها به ابر با  $c_{off}(t)$  نمایش داده شده و بر اساس وضعیت ازدحام در شبکه محاسبه می‌شود. وضعیت ازدحام در شبکه  $h(t)$  است که مجموع تأخیر انتقال رفت و برگشت<sup>۲۷</sup> و تأخیر پردازش در ابر است. تأخیر رفت و برگشت با  $ping$  به دست می‌آید. مقدار ازدحام در هر بازه زمانی طبق فرمول  $RTT(t) + delay_{cloud\_service}(t)$  به دست خواهد آمد. در نتیجه تأخیر ارسال بارهای ارسال شده به ابر بر اساس  $c_{off}(h(t), \mu(t), \lambda(t)) = (\lambda(t) - a) \cdot h(t)$  از رابطه  $h(t)$  حاصل می‌شود.

درنهایت هزینه تأخیر کل بارهای ورودی به شبکه از مجموع سه تأخیر بالا به دست می‌آید:

$$c_{delay}(h(t), \lambda(t), \mu(t), m(t)) = c_{wi}(\lambda(t)) + c_{lo}(\mu(t), m(t)) + c_{off}(h(t), \mu(t), \lambda(t)) \quad (1)$$

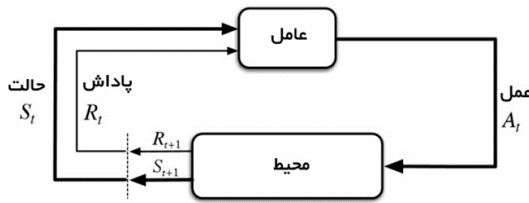
که در محاسبات مقدار  $c_{wi}(\lambda(t))$  لحاظ نخواهد شد.

#### ۳-۱-۳- مدل مصرف انرژی

در این مقاله، کل مصرف انرژی در یک بازه زمانی به دودسته تقسیم می‌شود:

- انرژی لازم برای عملیات پایه و انتقال بارها در لبه شبکه که با  $d_{op}(t)$  نمایش داده می‌شود. این نوع از انرژی به ارسال بارها به ابر یا پردازش آن‌ها در لبه شبکه بستگی ندارد و همواره فقط به میزان بار ورودی به شبکه (یعنی  $\lambda(t)$ ) بستگی دارد.  $d_{op}(t)$  از جمع دو نوع انرژی به دست

- پاداش<sup>۳۲</sup>: بازخورد از محیط
- سیاست<sup>۳۳</sup>: روشی برای نگاشت حالت عامل به عمل



شکل ۶- نمای کلی از سیستم یادگیری تقویتی

در یادگیری تقویتی هدف پیدا کردن سیاستی مناسبی است که پاداش کل<sup>۳۴</sup> سیستم را برای عامل بیشینه می‌کند. برای ساخت یک سیاست بهینه، عامل علاوه بر مسئله جست‌وجوی حالت‌های جدید، باید پاداش را به‌طور هم‌زمان بیشینه کند. به این کار موازنه جست‌وجو و استخراج<sup>۳۵</sup> گفته می‌شود. شکل ۶ نمای کلی سیستم یادگیری تقویتی را نشان می‌دهد.

در این مسئله وضعیت سیستم توسط  $s(t) \triangleq (\lambda(t), g(t), h(t), b(t))$  توصیف شده که در ابتدای هر بازه‌ی زمانی قابل‌مشاهده است. عملی که به محیط اعمال می‌کنیم و در آینده آن را حدس می‌زنیم مقدار بار قابل‌پردازش در لبه شبکه یعنی  $\mu(t)$  است.

با مشخص شدن مقدار باری که قرار است در لبه شبکه پردازش شود وضعیت باتری سیستم برای وضعیت بعدی در دو حالت ذکر شده قابل‌محاسبه است: اگر  $b(t) \leq d_p(\lambda(t))$  باشد؛

$$b(t+1) = b(t) + g(t) \quad (5)$$

در غیر این صورت؛

$$b(t+1) = b(t) + g(t) - d(\lambda(t), \mu(t), m(t)) \quad (6)$$

پس از پردازش بارها در ابر و لبه، هزینه تحمیل شده به سیستم  $c(t)$  را به‌عنوان بازخورد از محیط در نظر می‌گیریم که به شکل زیر محاسبه می‌شود: اگر  $b(t) \leq d_p(\lambda(t))$  باشد؛

$$c(t) = c_{delay}(h(t), \lambda(t), 0, 0) + c_{bak}(\lambda(t)) \quad (7)$$

در غیر این صورت؛

$$c(t) = c_{delay}(h(t), \lambda(t), \mu(t), m(t)) + c_{battery}(t) \quad (8)$$

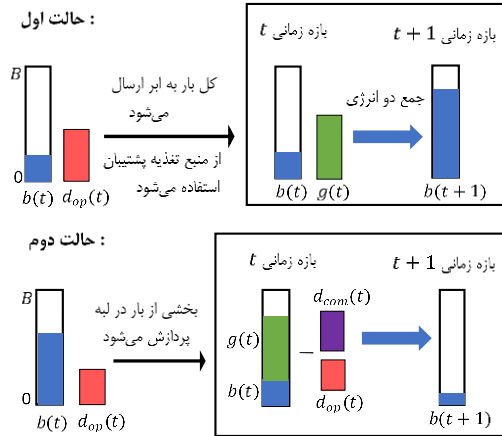
در بخش اول (اگر میزان باتری کمتر از انرژی موردنیاز باشد)  $\mu(t) = 0$  خواهد شد و برای ارسال بارها به ابر از منبع تغذیه پشتیبان استفاده خواهد شد. در بخش دوم که باتری به‌اندازه کافی داریم بجای هزینه منبع تغذیه پشتیبان، هزینه استهلاک باتری اضافه خواهد شد.

فرآیندهای تصمیم‌گیری مارکوف<sup>۳۶</sup> پارچوب‌های ریاضی هستند که برای تشریح محیط در یادگیری تقویتی استفاده می‌شوند و تقریباً همه مسائل با MDP قابل‌حل هستند. یک MDP شامل مجموعه‌ای متناهی از حالت‌های محیط<sup>۳۷</sup>، مجموعه‌ای از اعمال ممکن<sup>۳۸</sup> A در هر حالت، پاداش دارای مقدار حقیقی R و مدل انتقال  $P(s', s|a)$  است. وضعیت فعلی،  $s$  وضعیت آینده سیستم و  $a$  عملی است که به محیط اعمال می‌شود.

ابر ارسال کرد  $(\lambda(t) - \mu(t))$  پس میزان باتری برای بازه زمانی بعدی از رابطه (۳) محاسبه خواهد شد:

$$b(t+1) = b(t) + g(t) - d(\lambda(t), \mu(t), m(t)) \quad (3)$$

در این حالت باتری استهلاک خواهد داشت که از هزینه آن رابطه (۴) محاسبه می‌شود:



شکل ۵- وضعیت باتری در دو حالت

$$c_{battery}(t) = \omega \cdot \max\{d(\lambda(t), \mu(t), m(t)) - g(t), 0\} \quad (4)$$

که در آن  $\omega > 0$  هزینه استهلاک یک واحد از باتری است. حالت دوم شکل ۵ این موضوع را به تصویر بیان می‌کند.

درنهایت با توجه به چهار مدل بیان‌شده‌ی فوق، معماری سیستم پیشنهادی مطابق با شکل ۳ خواهد شد. در این شکل مجموعه درخواست‌های  $\lambda(t)$  از سمت کاربران وارد ایستگاه پایه می‌شوند که وظیفه آن توزیع بارها میان سرورهای لبه و ابر است. با استفاده از الگوریتم ارائه‌شده در بخش ۳-۳، میزان بار قابل‌پردازش در لبه شبکه  $(\mu(t))$  محاسبه شده و متناسب با آن درخواست‌ها را به سرورهای لبه و باقی‌مانده آن‌ها را  $(\lambda(t) - \mu(t))$  به سمت ابر ارسال می‌کند. ارسال بارها به ابر، شبکه را دچار ازدحام می‌کند و تأخیر بیشتری به بارها تحمیل می‌شود. بنابراین در هر بازه زمانی این تأخیر اندازه‌گیر شده  $(h(t))$  تا تصمیمات مناسب‌تری اتخاذ شود. از سوی دیگر منابع تولید انرژی‌های تجدید پذیر  $(g(t))$  در هر بازه زمانی، انرژی لازم برای محاسبات در لبه شبکه را تأمین می‌کند. در صورتی که انرژی تجدید پذیر بیش از نیاز سرورها باشد، باقیمانده آن در باتری‌های لبه شبکه  $b(t)$  ذخیره می‌شود. در صورت کمبود انرژی تجدید پذیر، از انرژی ذخیره‌شده در این باتری‌ها استفاده می‌شود.

### ۲-۳- فرموله‌سازی مسئله

در این بخش، مسئله تخصیص بارها به ابر و لبه‌ی شبکه بر مبنای یادگیری تقویتی فرموله می‌شود تا هزینه سیستم به حداقل برسد. در یادگیری تقویتی یک عامل<sup>۳۸</sup> با تعامل با محیط و استفاده از آزمون و خطاها و استفاده از بازخوردهای اعمال<sup>۳۹</sup> مسئله را یاد می‌گیرد. در نتیجه از پاداش‌ها و تنبیه‌ها به‌عنوان سیگنال‌هایی برای رفتار مثبت و منفی بهره برده می‌شود.

عناصر اصلی یک مسئله یادگیری تقویتی شامل مباحث زیر است:

- محیط<sup>۴۰</sup>: جهان فیزیکی که عامل در آن عمل می‌کند.
- حالت<sup>۴۱</sup>: موقعیت فعلی عامل

سیس، درستی هر قانون  $k_j$ ، مشخص می‌شود. در رابطه زیر، مقادیر  $\alpha$  و  $\epsilon_0$  ثابت‌هایی است که شکل تابع درستی را کنترل می‌کند.

$$k_j = \alpha \left(\frac{\epsilon_0}{\epsilon}\right)^v \text{ or } k_j = 1 \text{ if } \epsilon < \epsilon_0 \quad (12)$$

یک درستی نسبی برای هر سیستم محاسبه می‌شود.

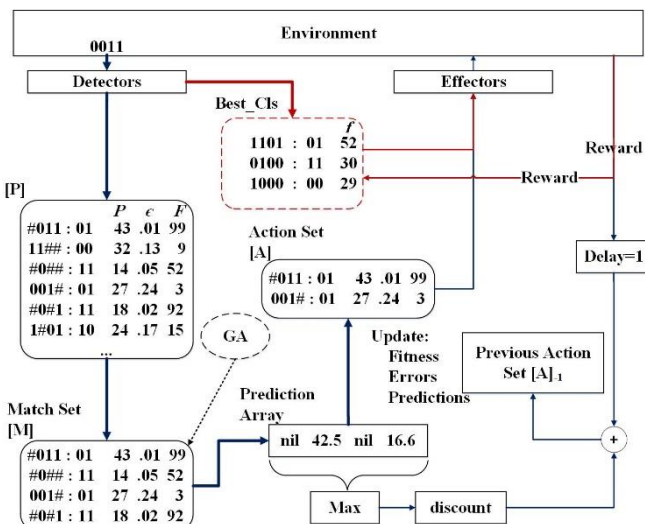
$$K'_j = \frac{K_j}{\sum_{x \in [A]} K_x} \quad (13)$$

سیس درستی نسبی برای به‌روز ساختن مقدار برازندگی طبقه بند استفاده می‌شود و اگر برازندگی به اندازه  $\frac{1}{\beta}$  بار اصلاح‌شده باشد از رابطه (۱۴) استفاده می‌شود:

$$F_j = F_j + \beta(k'_j - F_j) \quad (14)$$

در غیر این صورت برازندگی برابر مقدار متوسط درستی‌های نسبی تا لحظه فعلی می‌گردد. به بیان خلاصه، مقدار برازندگی در XCS به‌صورت معکوس تابعی از خطای پیش‌بینی پاداش است که در آن خطاهای کمتر از مقدار  $\epsilon_0$ ، میزان برازندگی را تغییر نمی‌دهند.

بیشینه آرایه پیش‌بینی سیستم  $P(a_i)$  توسط ضریبی به نام  $\gamma$  تخفیف یافته و برای به‌روز ساختن قوانین از مرحله زمانی قبل استفاده می‌شود. بنابراین در XCS نوعی از Q-learning در روند یادگیری تقویتی استفاده می‌شود. الگوریتم ژنتیک ابتدا روی [M] اعمال می‌شود اما ویلسون آن را به [A] اعمال نمود تا از ترکیب نامناسب قوانین جلوگیری کند. به‌ویژه زمانی که ناهمگنی زیادی در فضای کلیات هر کنش در [M] دیده می‌شود [۳۸]. در این روش، دو قانون بر اساس معیار برازش از لیست [A] انتخاب می‌شوند. جایگزینی قانون به‌صورت عمومی بوده و بر اساس اندازه هر مجموعه کنش، یک قانون باهدف متوازن ساختن منابع در بین ساختارهای بنیادی در آن شرکت می‌کند. الگوریتم ژنتیک در درون مجموعه کنش مفروض بر اساس متوسط فاصله زمانی از آخرین باری که قوانین بنیادی در الگوریتم شرکت کرده‌اند تحریک می‌شود.



شکل ۷- نمای کلی XCS

شکل ۷ نمایشی از XCS را نشان می‌دهد. در این مثال ورودی چهار بیتی و کنش دوبیتی است. تعداد کنش‌های تعریف‌شده در این مثال چهار عدد است. detector ورودی 0011 را از محیط دریافت کرده و با توجه به این ورودی و مجموعه [p]،

برای به حداقل رساندن هزینه بلندمدت سیستم باید در هر بازه زمانی هزینه در شبکه کاهش پیدا کند، بنابراین با استفاده از MDP و معادله بلمن به فرمول زیر می‌رسیم:

$$C^*(s) = \min_{a \in A} \left\{ c(s, a) + \sigma \cdot \sum_{s' \in S} P(s', s|a) \cdot C^*(s') \right\} \quad (9)$$

در اینجا که  $\delta < 1$  عامل تخفیف است که ثابت در نظر گرفته می‌شود، و این موضوع را نشان می‌دهد که هزینه‌های آینده وزن کمتری نسبت به هزینه فعلی دارد. با توجه به فرمول بالا، سیاست تخصیص بار، میزان  $\mu(t)$  را طوری انتخاب می‌کند که در آینده کمترین میزان هزینه به سیستم اعمال شود.

### ۳-۳- سیستم دسته‌بند یادگیر

سیستم‌های دسته‌بندی کننده یادگیر، سیستم‌هایی مبتنی بر قانون هستند. در این سیستم‌ها قوانین عمدتاً به فرم "IF condition THEN action" هستند. از یک الگوریتم تکاملی یا روش‌های شهودی می‌توان برای جستجو در فضای قوانین موجود و در همان زمان از فرآیند یادگیری دیگر برای انتساب نحوه استفاده به قوانین موجود می‌توان استفاده نمود که در واقع باعث هدایت جستجو به سمت قوانین بهتر می‌شود [۳۲، ۳۳]. اصطلاح LCS نخست توسط هلند معرفی شد و در ابتدا توسعه‌ای برای الگوریتم‌های ژنتیکی بود. سال‌ها بعد، با همکاری رایتمن، سیستم‌های شناختی سطح یک (CS-1) ارائه شد [۳۳]. سیس، هلند کار قبلی را اصلاح کرد تا به شکل استاندارد درآید. با این وجود، سیستم هلند به قدری پیچیده بود که پیاده‌سازی سیستم‌های واقعی با آن دشوار بود. سال‌ها بعد، استوارت ویلسون نوعی از LCS را به نام XCS ارائه داد که در آن برازش قوانین تنها بر اساس میزان دقت کنش‌های صورت گرفته توسط آن‌ها تعیین می‌شود [۳۴]. در دو دهه اخیر این مدل LCS استفاده زیادی در حل مسائل واقعی چون داده‌کاوی داشته است [۳۵-۳۷].

مهم‌ترین تفاوت میان سیستم XCS و سیستم‌های LCS دیگر، قبل از نحوه نمایش برازندگی قانون برای الگوریتم ژنتیک تنها مبتنی بر میزان پاداشی که قانون به دست می‌آورد نبود و تنها بر مبنای میزان درستی پیش‌بینی، مقدار پاداش  $\rho$  بود. هدف اصلی در سیستم XCS، به‌جای تمرکز در نتایج بنیادین نهایی، ساخت نقشه کامل و دقیقی از فضای مسئله از طریق عمومی‌سازی‌های مؤثری است که در آن XCS یک تابع ارزشی را از کل فضای حالت-کنش می‌آموزد. در هر مرحله زمانی، یک مجموعه تطابق  $[M]^{37}$  ساخته می‌شود. سیس، یک پیش‌بینی سیستمی برای هر کنش در [M] متناظر با مقدار متوسط پیش‌بینی‌ها که با برازندگی قانون‌های لیست کنش‌ها  $[A]^{38}$  وزن دهی شده است، انجام می‌شود. سیس، کنش سیستم به‌صورت دقیق یا تصادفی انتخاب می‌شود (معمولاً به ازای هر تجربه از احتمال  $0.5$  استفاده می‌شود). در چنین سیستمی اگر لیست [M] خالی باشد از پوشش استفاده می‌شود.

یادگیری تقویتی در XCS مشتمل بر به‌روز سازی سه پارامتر  $\epsilon$ ،  $\rho$  و  $F$  برای هر قانون است. این سه پارامتر به ترتیب خطا، پیش‌بینی درستی و برازندگی قانون را نشان می‌دهند. معیار برازندگی هر قانون بر اساس درستی نسبی قانون در طی پنج مرحله به‌روز می‌شود.

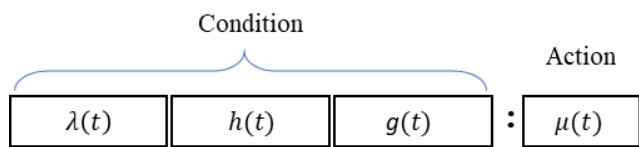
ابتدا میزان خطای هر قانون به‌روز می‌شود:

$$\epsilon_j = \epsilon_j + \beta(|reward - \rho_j| - \epsilon_j) \quad (10)$$

در گام بعدی پیش‌بینی‌های هر قانون به‌روز می‌شود:

$$\rho_j = \rho_j + \beta(reward - \rho_j) \quad (11)$$

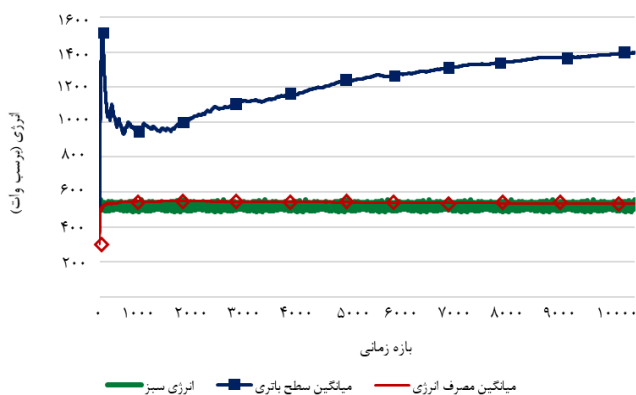
بازخورد همان هزینه ایست که به سیستم تحمیل شده و الگوریتم به دنبال کاهش این هزینه‌ها عمل خواهد کرد.



شکل ۸- نمونه‌ی یک سطر از دسته‌بند

#### ۴- پیاده‌سازی و ارزیابی

روش پیشنهادی روی سیستمی با پردازنده ۸ هسته‌ای و فرکانس ۱/۸ GHz و حافظه ۱۲ گیگابایت پیاده‌سازی شده است. در این بخش ابتدا مقادیر تنظیم شده برای متغیرهای سناریو پیشنهادی بیان شده و سپس نتایج حاصله ارزیابی می‌شود. در این سناریو بازه‌های زمانی ۱۵ دقیقه فرض شده است که تعداد بار ورودی در هر بازه زمانی  $\lambda(t)$  بین ۱۰ درخواست بر ثانیه تا ۱۳۰ درخواست بر ثانیه، به صورت تصادفی به شبکه وارد می‌شود. ازدحام در شبکه بین ۱۰ میلی‌ثانیه تا ۲۰ میلی‌ثانیه تغییر می‌کند. مقدار انرژی تجدید پذیر طبق [۲۸] از توزیع نرمال با پارامتر  $N(520W, 150)$  به سیستم وارد می‌شود. حداکثر ظرفیت باتری  $B = 2kWh$  در نظر گرفته شده که در  $b(0) = 0$  است. مصرف انرژی ثابت ایستگاه پایه  $d_{sta} = 300W$  است. حداکثر تعداد سرورها در لبه شبکه  $M = 10$  است که هر کدام از این سرورها در صورت فعال بودن  $150W$  مصرف می‌کند. حداکثر نرخ پردازشی هر سرور ۲۰ درخواست بر ثانیه است. هزینه استهلاک هر واحد از باتری  $\omega = 0.01$  خواهد بود و ضریب هزینه هر واحد از منبع تغذیه پشتیبان  $\phi = 0.15$  است.



شکل ۹- میانگین انرژی‌ها در لبه شبکه

شکل ۹ میزان انرژی باتری، انرژی مصرفی بارهای کاری و میزان انرژی تجدید پذیر را در لبه شبکه برحسب وات در هر بازه زمانی نشان می‌دهد. همان‌طور که در شکل مشخص است میانگین مصرف انرژی بارهای کاری در لبه شبکه تقریباً برابر انرژی تجدید پذیر در لبه شبکه است، پس می‌توان نتیجه گرفت روش پیشنهادی در تلاش است بارها به‌گونه‌ای توزیع شوند، تا انرژی لازم پردازش آن‌ها بیشتر از طریق انرژی تجدید پذیر تأمین شود.

نتایج حاصل از شبیه‌سازی گواه بر آن است که با کنترل مصرف انرژی در لبه شبکه می‌توان انرژی بیشتری در باتری‌ها ذخیره کرد و آن را در حالت شارژ نگه داشت. همان‌طور که در ابتدای بخش ذکر شده، سطح انرژی باتری‌ها در لبه شبکه در ابتدای شبیه‌سازی صفر است که بعد از مدت کوتاهی شروع به شارژ شدن می‌کند. این نتایج نشان می‌دهد سطح انرژی باتری به‌صورت صعودی افزایش پیدا کرده و پس از ۱۳ هزار بازه زمانی به بیش از ۱۴۰۰ وات می‌رسد. دلیل نوسانات اولیه خالی بودن

مجموعه  $[m]$  ایجاد شده است. به عبارتی دسته‌بندی‌هایی که شرایط آن‌ها با ورودی منطبق شده است در مجموعه  $[m]$  نگهداری می‌شوند. در این مثال کنش‌های منحصر به فردی که در مجموعه  $[m]$  قرار دارند ۲ عدد یعنی  $(01)$  و  $(11)$  است. آرایه پیش‌بینی بر اساس این مجموعه  $[m]$  ایجاد می‌شود.

نحوه محاسبه مقدار آرایه پیش‌بینی به ازای کنش  $01$  بدین صورت است. در مجموعه  $[m]$  دو دسته‌بند با کنش  $01$  وجود دارد. در هر دسته‌بند مقدار  $P$  و  $F$  آن دسته‌بند در یکدیگر ضرب شده و با دیگر دسته‌بندها جمع می‌شود. برای اولین دسته‌بند که دارای کنش  $01$  است مقدار  $F = 99$  و  $P = 43$  با یکدیگر ضرب شده و مقدار  $4275$  به دست می‌آید. این عمل برای دسته‌بند سوم که دارای کنش مشابهی  $(01)$  است انجام شده و مقدار  $81$  به دست می‌آید. حال مجموع این دو عدد تقسیم بر مجموع مقادیر  $F$  هر دو دسته‌بند  $(102)$  می‌شود. مقدار نهایی به دست آمده برابر با  $42.5$  است. این مقدار به ازای این کنش در آرایه پیش‌بینی قرار داده می‌شود. عبارت زیر نحوه محاسبه این مقدار را نشان می‌دهد.  $n$  تعداد دسته‌بندهای موجود در مجموعه  $[m]$  با کنشی مشابه است.

$$\frac{\sum_1^n F \times P}{\sum_1^n F} \quad (15)$$

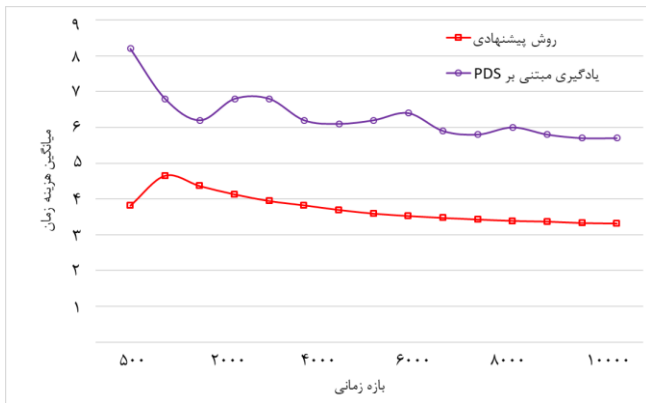
این محاسبات برای تمام کنش‌های موجود در مجموعه  $[m]$  انجام می‌شود. در آخر در صورتی که تعداد کنش‌های آرایه پیش‌بینی از کل کنش‌ها کمتر باشد عمل پوشش صورت می‌گیرد. در این مثال دو کنش موجود است و دو کنش وجود ندارد؛ بنابراین عمل پوشش دومرتبه برای ایجاد دسته‌بندی‌هایی با آن کنش‌ها انجام می‌شود. اما با گذشت زمان بهترین دسته‌بندها از مجموعه  $[p]$  حذف می‌شوند. با هدف نگهداری بهترین دسته‌بندها در طول زمان، به سیستم XCS حافظه جانبی اضافه کرده‌ایم. این حافظه در شکل ۷ به نام Best\_Cls نشان داده شده است. در این حافظه جدید، شرایط پیش‌آمده را همراه با بهترین عمل مناسب برای آن و بهترین پاداشی که آن عمل از محیط دریافت کرده، نگهداری می‌کنیم.

روند انتخاب کنش متناظر با ورودی، در این سیستم به شرح زیر است: پس از دریافت شرایط محیط از detector، برای انتخاب ادامه مسیر (استفاده از دسته‌بند ذخیره شده در حافظه یا XCS عادی) یک عدد تصادفی تولید می‌شود؛ با احتمال  $n$  و در صورتی که قبلاً دسته‌بندی مشابه با شرایط فعلی ذخیره شده باشد، از کنش موجود در حافظه جهت اعمال به محیط استفاده می‌شود (مقدار  $n$  در این آزمایش  $0.5$  در نظر گرفته شده است). همچنین با احتمال  $1 - n$  می‌توان روند عادی XCS را جهت به دست آوردن کنش مناسب انجام داد. در انتها در صورت عدم وجود دسته‌بند در حافظه، شرایط، کنش و مقدار reward به‌عنوان یک دسته‌بند جدید، به حافظه اضافه می‌شود. از طرفی در صورت وجود دسته‌بند در حافظه، با مقایسه مقدار reward دسته‌بند ذخیره شده در حافظه و reward دسته‌بند فعلی (دسته‌بند XCS) آن دسته‌بند در حافظه به‌روزرسانی می‌شود.

#### ۴-۳ کاربرد دسته‌بند یادگیر در مدل پیشنهادی

از سیستم پیشنهاد شده در بخش قبل برای کمک به تصمیم‌گیری ایستگاه پایه جهت تعیین میزان بار پردازشی در لبه شبکه استفاده می‌شود. در مدل‌سازی بارها در پردازش لبه ورودی این سیستم (شرایط) وضعیت شبکه و خروجی (کنش) آن میزان بار پردازشی در لبه شبکه است. شکل ۸ نمایی از شرایط دریافتی سیستم و کنشی است که دسته‌بند تولید می‌کند.

شکل ۸ گواه بر این است که وضعیت محیط با سه پارامتر  $\lambda(t)$ ،  $h(t)$  و  $g(t)$  قابل دریافت است. همچنین وضعیت باتری در لبه  $b(t)$  در سیستم معلوم است زیرا در مرحله قبل محاسبه شده است. در نهایت بر اساس این شرایط و با استفاده از الگوریتم ذکر شده میزان بار مناسب برای پردازش در لبه شبکه پیش‌بینی و یاد گرفته می‌شود. این میزان بار  $\mu(t)$  به‌عنوان یک کنش به محیط اعمال می‌شود. در نهایت نتیجه این عمل به‌عنوان پاداش از جانب محیط به ما بازخورد داده می‌شود. این



شکل ۱۱- میانگین هزینه زمان (تأخیر)

## ۵- نتیجه‌گیری

ارسال حجم زیادی از داده‌ها به سمت ابر تأخیر زیادی به بارهای کاری تحمیل می‌کند پس می‌توان با پردازش برخی از آن‌ها در لبه شبکه سبب کاهش تأخیر بارهای کاری شد اما از سوی دیگر در لبه شبکه با کمبود انرژی روبرو خواهیم شد. در این پژوهش روشی جهت متعادل‌سازی توزیع بار در لبه شبکه و ایجاد تعادل میان تأخیر و مصرف انرژی در لبه ارائه شده است. همان‌طور که نتایج حاصل از شبیه‌سازی نشان می‌دهد میزان تأخیر بارهای کاری کاهش قابل توجهی داشته و همچنین سطح باتری به بیشترین میزان خود می‌رسد. در مقایسه با روش پیشین که سطح باتری بالای ۶۰٪ ندارد روش مبتنی بر سیستم دسته‌بند یادگیر با حافظه سطح باتری عمدتاً به حداکثر ظرفیت خود می‌رسد. همچنین میانگین تأخیر بارها بعد از بازه زمانی ۱۰۰۰ نزولی شده و به سه ثانیه همگرا می‌شود؛ بنابراین از این مدل می‌توان در شبکه‌هایی استفاده کرد که منابع پردازشی آن‌ها به انرژی برق دسترسی ندارند و باتری تنها منبع انرژی آن‌هاست. در این مناطق می‌توان از انرژی باد یا انرژی خورشید یا هر نوع انرژی تجدید پذیر دیگری به‌عنوان منبع تغذیه کمکی و حتی تنها منبع اصلی انرژی استفاده کرد تا هزینه استهلاک باتری‌ها نیز کاهش پیدا کند. در ادامه این پژوهش می‌توان از روش‌های یادگیری ماشین دیگری همچون شبکه‌های عصبی استفاده کرد.

## ۶- مراجع

- [1] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1-20, 2016.
- [2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787-2805, 2010.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.
- [4] D. Macedo, L. A. Guedes, and I. Silva, "A dependability evaluation for Internet of Things incorporating redundancy aspects," in *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, IEEE, pp. 417-422, 2014.
- [5] N. Maalel, E. Natalizio, A. Bouabdallah, P. Roux, and M. Kellil, "Reliability for emergency applications in internet of things," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, IEEE, pp. 361-366, 2013.
- [6] J. Kempf, J. Arkkio, N. Beheshti, and K. Yedavalli, "Thoughts on reliability in the internet of things," in *Interconnecting smart objects with the Internet workshop*, vol. 1, pp. 1-4, 2011.
- [7] F. Ganz, R. Li, P. Barnaghi, and H. Harai, "A resource mobility scheme for service-continuity in the Internet of Things," in *IEEE International Conference on Green Computing and Communications*, IEEE, pp. 261-264, 2012.

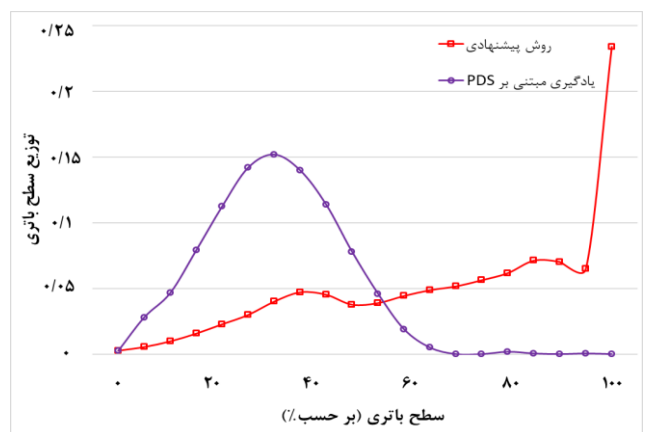
حافظه Best\_Cls است که دسته‌بند به‌صورت تصادفی و بدون پیش‌بینی توسط سیستم دسته‌بند یادگیر بدون حافظه، کنش‌ها را انتخاب و اعمال می‌کند. با اضافه شدن دسته‌بند‌های بهینه به حافظه Best\_Cls، میزان انتخاب حالت تصادفی توسط سیستم دسته‌بند یادگیر کاهش پیدا می‌کند. در نتیجه در هر مرحله احتمال انتخاب دسته‌بند بهینه از حافظه Best\_Cls بیشتر می‌شود. بدین جهت این حافظه باعث بهبود عملکرد سیستم دسته‌بند یادگیر در انتخاب حالت بهینه توزیع بار می‌شود.

در ادامه روش پیشنهادی با الگوریتم مبتنی PSD [۲۸] مقایسه شده است. این روش در میان کارهای پیشین تنها الگوریتم توزیع بار است که مبتنی بر یادگیری عمل می‌کند. این الگوریتم با ایجاد یک حالت میانی در بین حالت‌ها، بخش تکرار ارزش را از یادگیری تقویتی جدا می‌کند و در تلاش است تا هزینه‌های بلندمدت سیستم را به حداقل برساند.

شکل ۱۰ توزیع وضعیت باتری را برای ۱۰۰۰۰ بازه زمانی نمایش می‌دهد. همان‌طور که مشاهده می‌شود سطح باتری در روش پیشنهادی بیشتر از ۶۰٪ دیده می‌شود، در صورتی که در روش PSD احتمال اینکه سطح باتری به بیش از ۶۰٪ برسد تقریباً صفر است. دلیل افزایش سطح باتری، این است که در الگوریتم پیشنهادی تلاش می‌شود بیشتر انرژی موردنیاز برای پردازش داده‌ها از انرژی تجدید پذیر به دست آید تا استهلاک باتری کاهش پیدا کند و در نتیجه هزینه بلندمدت سیستم کاهش یابد. نکته‌ی دیگری که در این نمودار جلب توجه می‌کند، الگوریتم پیشنهادی در اکثر مواقع سطح باتری را در ۱۰۰ درصد نگه می‌دارد که نسبت به روش مبتنی بر PSD افزایش چشم‌گیری داشته است. دلیل این افزایش، تلاش الگوریتم پیشنهادی برای ایجاد تعادل میان هزینه‌ی استهلاک باتری و هزینه‌ی منبع تغذیه پشتیبان در طولانی‌مدت است. این الگوریتم با ایجاد تعادل میان انتخاب تصادفی کنش‌ها و دسته‌بند‌های ذخیره‌شده در حافظه کمک می‌کند تا انرژی تجدید پذیر، سطح باتری را به حداکثر ظرفیت خود برساند.

دومین معیار برای مقایسه، تأخیر بارهای کاری است. شکل ۱۱ میانگینی از تأخیر پردازش بارها در لبه شبکه و ارسال باقی آن‌ها به ابر را برای ۱۰ هزار بازه زمانی برحسب ثانیه نشان می‌دهد. با توجه به شکل، روش مبتنی بر PSD در ابتدا تأخیر زیادی را متحمل می‌شود.

دلیل افزایش تأخیر روش پیشنهادی، تا بازه زمانی ۱۰۰۰، خالی بودن حافظه Best\_Cls و انتخاب‌های تصادفی بدون پیش‌بینی است. اما این افزایش تأخیر به پنج ثانیه هم نمی‌رسد در صورتی که در روش PSD پس از بازه زمانی ۵۰۰، باوجود داشتن شیب نزولی، این معیار حداقل به شش ثانیه می‌رسد. دلیل افزایش کم روش پیشنهادی، تأثیر مستقیم پاداش محیط به مقدار fitness عناصر دسته‌بند است که با مشاهده افزایش ناگهانی این تأخیر سعی در کنترل آن دارد. این افزایش تأخیر بعد از مدتی متوقف شده و بعد از ۱۰۰۰ بازه زمانی شروع به کاهش می‌کند.



شکل ۱۰- توزیع وضعیت باتری

- Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361-373, 2017.
- [29] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, and C. Liu, "Fog computing for energy-aware load balancing and scheduling in smart factory," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4548-4556, 2018.
- [30] D. Zeng, L. Gu, and H. Yao, "Towards energy efficient service composition in green energy powered Cyber-Physical Fog Systems," *Future Generation Computer Systems*, 2018.
- [31] Elia: Power generation. [Online]. Available: <https://www.elia.be/en/grid-data/power-generation>.
- [32] L. Pier Luca, *Learning Classifier Systems, From Foundations to Applications*, Springer Science & Business Media, 2000.
- [33] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
- [34] S. W. Wilson, "Classifier fitness based on accuracy," *Evol. Comput.*, vol. 3, no. 2, pp. 149-175, 1995.
- [35] M. H. Arif, J. Li, M. Iqbal, and K. Liu, "Sentiment analysis and spam detection in short informal text using learning classifier systems," *Soft Computing*, vol. 22, no. 21, pp. 7281-7291, 2018.
- [36] B. Bartin, "Use of learning classifier systems in microscopic toll plaza simulation models," *IET Intelligent Transport Systems*, vol. 13, no. 5, pp. 860-869, 2019.
- [37] M. R. Karlsen and S. Moschogiannis, "Evolution of control with learning classifier systems," *Applied network science*, vol. 3, no. 1, pp. 30, 2018.
- [38] P. L. Lanzi, D. Loiacono, S. W. Wilson, and D. E. Goldberg, "Generalization in the XCSF Classifier System: Analysis, Improvement, and Extension," *Evol. Comput.*, vol. 15, no. 2, pp. 133-168, 2007.
- [8] M. Al-Garadi, and A. Mohamed, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials*, 2020.
- [9] M. Rajan, P. Balamuralidhar, K. Chethan, and M. Swarnahpriyaah, "A self-reconfigurable sensor network management system for internet of things paradigm," in *International Conference on Devices and Communications (ICDeCom)*, IEEE, pp. 1-5, 2011.
- [10] D. Uckelmann, M.-A. Isenberg, M. Teucke, H. Halfar, and B. Scholz-Reiter, "Autonomous control and the internet of things: Increasing robustness, scalability and agility in logistic networks," in *Unique radio innovation for the 21st century*: Springer, pp. 163-181, 2011.
- [11] C. Sarkar, S. A. U. Nambi, R. V. Prasad, and A. Rahim, "A scalable distributed architecture towards unifying IoT applications," in *IEEE World Forum on Internet of Things (WF-IoT)*, IEEE, pp. 508-513, 2014.
- [12] A. Dunkels, J. Eriksson, and N. Tsiftes, "Low-power Interoperability for the IPv6-based Internet of Things," in *Proceedings of the 10th Scandinavian Workshop on Wireless Ad-Hoc Networks (ADHOC'11)*, Stockholm, Sweden, pp. 10-11, 2011.
- [13] I. Ishaq, D. Carels, G. K. Teklemariam, J. Hoebeke, F. Van den Abeele, E. Poorter, I. Moerman, and P. Demeester, "IETF standardization in the field of the internet of things (IoT): a survey," *Journal of Sensor and Actuator Networks*, vol. 2, no. 2, pp. 235-287, 2013.
- [14] M. M. Mahmoud, J. J. Rodrigues, K. Saleem, J. Al-Muhtadi, N. Kumar, and V. Korotaev, "Towards energy-aware fog-enabled cloud of things for healthcare," *Computers & Electrical Engineering*, vol. 67, pp. 58-69, 2018.
- [15] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *Global Internet of Things Summit (GIoTS)*, IEEE, pp. 1-6, 2017.
- [16] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854-864, 2016.
- [17] A. Yousefpour, F. Caleb, N. Tam, K. Krishna, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, 2019.
- [18] M. Ghobaei-Arani, A. Souri, and A. A. Rahmadian, "Resource management approaches in fog computing: a comprehensive review," *Journal of Grid Computing*, pp. 1-42, 2019.
- [19] R. Basir, Q. Saad, A. Mudassar, A. Monther, I. Muhammad Ashraf, M. Aamir, and M. Gidlund, "Fog Computing Enabling Industrial Internet of Things: State-of-the-Art and Research Challenges," *Sensors*, vol. 19, no. 21, p. 4807, 2019.
- [20] X. Liu and N. Ansari, "Toward green IoT: Energy solutions and key challenges," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 104-110, 2019.
- [21] H. Wu, L. Chen, C. Shen, W. Wen, and J. Xu, "Online geographical load balancing for energy-harvesting mobile edge computing," in *IEEE International Conference on Communications (ICC)*, IEEE, pp. 1-6, 2018.
- [22] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, IEEE, pp. 73-78, 2015.
- [23] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 1, pp. 19-26, 1998.
- [24] G. C. Hunt and M. L. Scott, "A guided tour of the Coign automatic distributed partitioning system," in *Proceedings Second International Enterprise Distributed Object Computing (Cat. No. 98EX244)*, IEEE, pp. 252-262, 1998.
- [25] D. Kanapram, G. Lamanna, and M. Repetto, "Exploring the trade-off between performance and energy consumption in cloud infrastructures," in *Second International Conference on Fog and Mobile Edge Computing (FMEC)*, IEEE, pp. 121-126, 2017.
- [26] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016.
- [27] J. Xu and S. Ren, "Online learning for offloading and autoscaling in renewable-powered mobile edge computing," in *IEEE Global Communications Conference (GLOBECOM)*, IEEE, pp. 1-6, 2016.
- [28] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE*

**مینا یعقوبی کیا** مدرک کارشناسی خود را در سال ۱۳۹۶

در رشته مهندسی کامپیوتر از دانشگاه بوعلی سینا اخذ نموده و در حال حاضر دانشجوی کارشناسی ارشد گرایش شبکه‌های کامپیوتری دانشگاه بوعلی سینا می‌باشد. تحقیقات ایشان در حوزه رایانش ابری و رایانش لبه است.

آدرس پست الکترونیکی ایشان عبارت است:

[m.yaghoobikia@eng.basu.ac.ir](mailto:m.yaghoobikia@eng.basu.ac.ir)



**مهدی عباسی** مدرک دکترای خود را در سال ۱۳۹۱ در

رشته معماری سیستم‌های کامپیوتری، از دانشگاه اصفهان اخذ و در حال حاضر استادیار گروه کامپیوتر دانشگاه بوعلی سینا می‌باشند. پژوهش‌های ایشان به‌طور عمده بر روی پردازنده‌های شبکه‌ای و پردازش سیگنال متمرکز است.

آدرس پست الکترونیکی ایشان عبارت است:

[abbasi@basu.ac.ir](mailto:abbasi@basu.ac.ir)



**میلاد رفیعی** در حال حاضر مدرس و دستیار پژوهشی

دانشگاه بوعلی سینا است. او مدرک کارشناسی ارشد فناوری اطلاعات (شبکه‌های کامپیوتری) خود را از دانشکده مهندسی دانشگاه بوعلی سینا دریافت کرده است. از علایق پژوهشی او می‌توان به شبکه‌های کامپیوتری، الگوریتم‌های پردازنده‌های شبکه‌ای، پردازش موازی و برنامه‌نویسی GPU

اشاره کرد. سیستم‌های دسته‌بند یادگیر و یادگیری ماشین از دیگر علاقه‌مندی‌های پژوهشی او هستند.

آدرس پست الکترونیکی ایشان عبارت است:

[m.rafiie@alumni.basu.ac.ir](mailto:m.rafiie@alumni.basu.ac.ir)



<sup>3</sup> Actuator

<sup>4</sup> Availability

<sup>1</sup> Internet of Things (IoT)

<sup>2</sup> Sensor

---

<sup>23</sup> Hang Wu

<sup>24</sup> Jiafu Wan

<sup>25</sup> Deze Zeng

<sup>26</sup> Cyber Physical System

<sup>27</sup> Round Trip Time (RTT)

<sup>28</sup> Agent

<sup>29</sup> Actions

<sup>30</sup> Environment

<sup>31</sup> State

<sup>32</sup> Reward

<sup>33</sup> Policy

<sup>34</sup> Total Reward

<sup>35</sup> Exploration vs Exploitation trade-off

<sup>36</sup> Markov Decision Processes | MDP

<sup>37</sup> Match Set

<sup>38</sup> Action Set

<sup>39</sup> Condition

---

<sup>5</sup> Reliability

<sup>6</sup> Mobility

<sup>7</sup> Performance

<sup>8</sup> Scalability

<sup>9</sup> Interoperability

<sup>10</sup> Security

<sup>11</sup> Management

<sup>12</sup> Trust

<sup>13</sup> Virtual Object

<sup>14</sup> Virtual Object Composite

<sup>15</sup> End to End

<sup>16</sup> Cloud Computing

<sup>17</sup> Data Center

<sup>18</sup> Data Center Networks

<sup>19</sup> Edge Computing

<sup>20</sup> Task Offloading

<sup>21</sup> Jie Xu

<sup>22</sup> Post-Decision State

# **Optimum distribution of workloads in edge computing using stateful learning classifier systems**

**Mina Yaghoobikia, Mahdi Abbasi, Milad Rafiee**

Department of Computer Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran.

---

## **Abstract**

Along with the growing evolution of using smart devices, the concept of the Internet of Things (IoT) has also developed. Increasing the number of smart objects has increased the production of big data and computational workloads on high scales. As a result, cloud computing is used as the main solution to control these workloads. However, the time-consuming nature of processing the workloads in the cloud is still considered as the most important issue in the domain of distributed networks. Processing workload at the edges of the network can reduce this time, but on the other hand, bringing processing of loads of data centers (that are connected to electricity resources) to the edges of the network faces the limitation of electrical energy provision. It is therefore necessary to distribute workloads in a balanced manner between the clouds and the edges of the network. This paper aims to create a balance between energy consumption at the edge of the network and the delay of processing workloads in the clouds. The results of the simulation show that the proposed method distributes the workloads multiple times more balanced than the recent methods. The proposed method reduces a 42 % reduction in workload delay as well as reducing the energy consumption at the edge of the network.

**Keywords:** Internet of Things (IoT); Edge Computing; Machine Learning; Renewable Energy.