

تاریخ دریافت مقاله: ۹۸/۰۲/۱۷

تاریخ پذیرش مقاله: ۹۸/۰۸/۱۶

ارائه راهکاری بر خط جهت تخصیص بهینه منابع در ردیابی سیگنالهای مغزی با معماری مه

غلامرضا حیدری

دانشکده مهندسی برق و کامپیوتر- دانشگاه قم - قم - ایران
پست الکترونیکی: ghr.heydari@stu.qom.ac.ir

دادمهر رهبری*

دانشکده مهندسی برق و کامپیوتر- دانشگاه قم - قم - ایران
پست الکترونیکی: d.rahbari@stu.qom.ac.ir

محسن نیکرأی

دانشکده مهندسی برق و کامپیوتر- دانشگاه قم - قم - ایران
پست الکترونیکی: m.nickray@qom.ac.ir

چکیده:

میانگین انرژی مصرفی دستگاه‌های موبایل کاهش یابد. واژه‌های کلیدی: پردازش مه، زمان‌بندی وظایف، یادگیری ماشین، دسته‌بندی بیزین.

اینترنت اشیا باعث افزایش حجم داده‌های تولیدشده در شبکه‌های رایانه‌ای شده است. هدف پردازش مه کنترل این حجم از داده است. زمان‌بندی در محیط‌های توزیع‌شده مسئله‌ای NP-hard است. در این مقاله به زمان‌بندی در مه به وسیله دسته‌بندی بیزین پرداخته شده است. از دسته‌بندی بیزین برای به دست آوردن نیازمندی‌های پردازشی وظایف استفاده شده است. پس از دسته‌بندی، متناسب با هر دسته، ماشین‌های مجازی به صورت پیش‌بینی شده ایجاد خواهند شد. شبیه‌ساز ifogsim برای بررسی عملکرد روش پیشنهادی در محیط مه استفاده شده است. الگوریتم‌ها روی برنامه سیستم ردیابی سیگنال‌های مغزی مورد ارزیابی قرار گرفته‌اند. بر طبق نتایج، تصمیم‌گیری هوشمند در محیط ابر ترکیب شده با مه باعث شده است تا انرژی مصرفی در ابر، هزینه اجرای وظیفه در ابر و

۱- مقدمه

اینترنت اشیا نسل آینده شبکه‌ها مانند شبکه‌های موبایل را متراکم، ناهمگن و ناپایدار خواهد کرد. از این رو فن‌های ایستای زمان‌بندی کنونی برای نسل‌های جدید شبکه‌ها مناسب نیستند. در حال حاضر بهره‌مندی از هوش مصنوعی و یادگیری ماشین زمینه مورد علاقه پژوهشگران برنامه‌ریزی شبکه شده است [۱]. با پدیدار شدن مفهوم مه تأخیر اندک، توزیع بزرگ مقیاس و جغرافیایی، تحرک، آگاهی از مکان، انعطاف‌پذیری، ناهمگنی، مقیاس‌پذیری فراهم شد و با ظهور اینترنت اشیا، مدل پردازش ابر، این ویژگی‌ها را برآورده نکرد و مه به منظور توسعه الگوی

پردازش ابری به سمت لبه شبکه ظهور کرد [۲، ۳، ۴]. نتایج [۵] اثبات کرده است هنگام استفاده از شبکه‌های مه در ترکیب با ابر زمان انتظار کمتر و نرخ داده بیشتر است. سیسکو استفاده از شبکه‌های مه را در سه سناریو پیشنهاد داده است [۶]: ۱) داده‌ها از لبه شبکه جمع‌آوری می‌شوند. ۲) دستگاه‌های ارسال کننده داده بسیار زیاد است. ۳) پردازش داده و تصمیم‌گیری باید در کمتر از یک ثانیه انجام شود. پردازش مه بیشتر با فراهم‌کنندگان سرویس و مالکان انواع تجهیزات در لایه مه مانند دستگاه کاربر نهایی، نقاط دسترسی، مسیریاب‌های لبه‌ای و غیره مرتبط است [۷]. با توجه به پویایی بسیار زیاد شبکه‌های مه توزیع عادلانه و بهینه منابع میان وظایف نیازمند الگوریتم‌های زمان‌بندی بهینه است. چگونگی و محل تأمین منبع پردازش در لبه شبکه در [۸] به تفصیل بررسی شده است. با توجه به نتایج مثبت استفاده از الگوریتم‌های یادگیری ماشین در پردازش ابر [۹] و با عنایت به بررسی جامع پیچیدگی زمانی روش‌های یادگیری ماشین برخط و غیربرخط در [۱۰]، در این مقاله نیز سعی شده است استفاده از الگوریتم دسته‌بندی بیزین برای زمان‌بندی برخط در پردازش مه بررسی شود. شبیه‌ساز iFogSim و برنامه آن به نام VRGAME^۱، روش پیشنهادی را مورد ارزیابی قرار می‌دهند. در ادامه این مقاله در بخش ۲ کارهای مرتبط، در بخش ۳ مدل سیستم و برنامه، در بخش ۴ روش پیشنهادی، در بخش ۵ نتایج شبیه‌سازی ارزیابی و در نهایت در بخش ۶ نتیجه‌گیری و پیشنهادها بیان شده است.

۲- کارهای مرتبط

زمان‌بندی در پردازش ابری دیدگاهی بدیع برای محاسبات توزیع‌شده و پردازش موازی برای ارائه سرویس بر پایه پرداخت به ازای استفاده است و زیرساخت، بُن‌سازه و نرم‌افزار را به‌عنوان سرویس ارائه می‌دهد [۱۱، ۱۲]. پایداری و کارایی سرویس‌های مه به فاکتورهایی مانند زمان‌بندی وظایف، منبع و جریان‌کار

1-VRGAME: Virtual Reality Game

وابسته است. مراکز داده، وظایف را منطبق با توافق‌نامه سطح سرویس^۲ و سرویس‌های درخواست شده دسته‌بندی می‌کنند [۱۱]. برای مهیا کردن سرویسی بهتر و استفاده بهینه از منابع، وظایف روی ماشین‌های مجازی بارگذاری می‌شوند و منابع به اشتراک گذاشته می‌شوند. در بسیاری از برنامه‌های کاربردی با دامنه خاص مانند سناریوهای برنامه‌های کاربردی صنعتی پردازش ابری قادر به پاسخ به موقع به کاربران نیست [۷]. براساس [۱۳] راه ارتباطی میان ابر و دستگاه‌های اینترنت اشیا گره‌های مه هستند.

۲-۱ زمان‌بندی در ابر و یادگیری ماشین

در [۱۴] سهولت مدیریت منبع در دستگاه‌های توزیع‌شده کلان‌مقیاس توسط یادگیری ماشین بحث شده است و براساس آن استفاده از شبکه‌های عصبی هزینه خنک‌کنندگی مراکز داده گوگل را ۴۰ درصد و یادگیری عمیق در لبه شبکه مقدار داده منتشرشده برگشتی به مراکز داده را کاهش داده است. بهینه‌سازی پارامترهای راندمان انرژی و طیف، تأخیر، پایداری و امنیت در مرحله عملیات شبکه‌ای نیاز به الگوریتم‌های تصمیم‌گیری و یادگیری بیدرنگ دارد [۱]. مهندسی ویژگی گام اولیه هر تحلیل مبتنی بر یادگیری ماشین به منظور انتخاب صحیح معیارهای داده است تا به‌عنوان ورودی به الگوریتم‌های یادگیری ماشین ارائه گردند [۱۳]. چون تعداد ماشین‌های مجازی و کارها در محیط پردازش ابری بسیار زیاد است در برخی متون از یادگیری موازی چند عامله برای سرعت دادن به طرح بهینه زمان‌بند استفاده شده است [۱۵]. زمان‌بندی در پردازش ابری به‌طور عمده بر روی بهبود استفاده از منابع و همچنین کاهش زمان اتمام کار تمرکز دارد [۱۲] و قابل‌تعمیم به پردازش مه نیز هست.

۲-۲ استفاده از یادگیری ماشین

در یادگیری ماشین دستیابی به نتایج بسیار دقیق با کاوش کامل مقادیر پارامترها ممکن است از نظر محاسباتی امکان پذیر نباشد. هدف کلیه برنامه‌های یادگیری ماشین،

2- SLA: Service Level Agreement

حداقل کردن تفاوت‌ها میان مفهوم هدف و خروجی تولید شده توسط مدل‌های آموزش دیده است [۱۰]. در مقایسه با هیوریستیک اقتضایی، رویکردهای یادگیری ماشین می‌توانند در تخصیص هوشمند منبع و انتخاب عمل بر اساس حالت‌های مفهومی و فاکتورهای محیطی در زمان بندی مفید باشند [۱۴]. بر اساس [۱۶] اثرات استفاده از مفاهیم یادگیری ماشین مانند سازوکار تخصیص منبع خودکار، زمان بندی و مدیریت هوشمند منابع در محیط‌های ابری بررسی شده است. برای این منظور برخی از روش‌های یادگیری ماشین مانند "SmartSLA" بررسی شده‌اند. بر اساس [۱۵] یک طرح زمان بندی وظیفه بر پایه یادگیری تقویتی طراحی شده است و زمان ورود تا اتمام کار و میانگین زمان انتظار را تحت محدودیت‌های منابع ماشین مجازی و ضرب‌الاجل حداقل کرده است. بر اساس [۱۲] در مسائل زمان بندی منبع به دلیل تمرکز راه‌حل‌های موجود روی یک هدف خاص مثل حداقل سازی زمان اجرا یا بارکاری و عدم استفاده از ویژگی‌های پردازش ابری، یک درخت دسته بندی و رگرسیون و یک الگوریتم بهینه سازی تغذیه باکتری اصلاح شده پیشنهاد شده است. بر اساس [۱۷] یک تکنیک انتخاب منبع چند عامله مبتنی بر شبکه عصبی که قادر به تقلید سرویس‌های کاربر متخصص در سیستم‌های توزیع شده مانند گریدها و ابرها است پیشنهاد شده است.

۲-۳ زمان بندی در مه و یادگیری ماشین

بر اساس [۱۸] داده‌کاوی و یادگیری ماشین در ابزارهای شبکه‌هایی مانند شبکه‌های حسگر بیسیم و بسیاری از مطالعات اخیر مدل‌های پیشگویانه‌ای را در شبکه‌هایی که ناپایداری و پویایی ویژگی اساسی آنان است ایجاد می‌نمایند و مدل یادگیری توزیع شده روی دستگاه حسگر و شبیه سازی جریان داده در مه نشان داده است که ترکیب پردازش مه و ابر برای برنامه‌های کاربردی اینترنت اشیا صنعتی سودمند است. در [۱۳] استفاده از یادگیری ماشین کم‌منبع روی دستگاه‌های مه که در نزدیکی وسایل

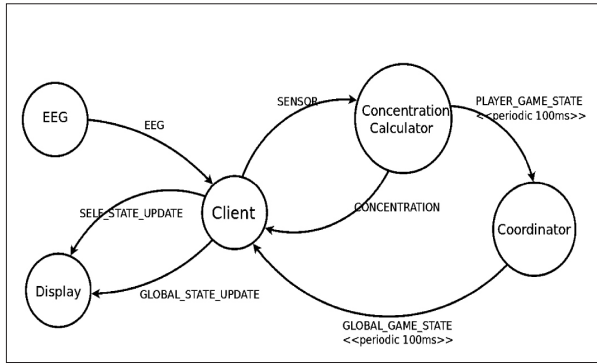
پوشیدنی مراقبت از سلامت نگهداری می‌شوند نشان داده است که یک رایانه با توانایی محدود داده‌کاوی می‌تواند داده‌های جمع‌آوری شده از حسگرهای پوشیدنی مختلف را برای برنامه مراقبت از سلامت راه دور تحلیل کند. بر اساس [۵] یادگیری تقویتی مبتنی بر سازوکار تخلیه کد برای اطمینان از ارائه خدمات کم‌تأخیر برای مشتریان سرویس موبایل پیشنهاد و زمان اجرا، تأخیر دسترسی به سرویس موبایل و مصرف انرژی دستگاه‌های موبایل را کاهش داده است. در [۱۹] سیستم بهینه سازی انتقال مبتنی بر یادگیری ماشین به عنوان بخشی از JAMScript که یک زبان برنامه نویسی چندزبانه برای اینترنت‌اشیاء است ارائه شده است بر اساس [۲] یک پیش بینی کننده داده در یک مدل مبتنی بر مه بر روی پروتکل سبک وزن MQTT^۳ ساخته شده است و انرژی دستگاه‌های انتهایی را با کاهش تبادل داده، ذخیره می‌نماید. بر اساس [۲۰] یک طرح زمان بندی انرژی-کارآمد بر اساس مدل Q_Learning برای وظایف متناوب در سیستم‌های بیدرنگ پیشنهاد شده است و در [۲۱] اینترنت اشیا شناختی مبتنی بر رادیو^۴ راه‌حل امیدوارکننده‌ای برای برنامه‌های کاربردی اینترنت اشیا است و زمان بندی انتقال مبتنی بر Q_Learning با استفاده از یادگیری عمیق برای CIoT پیشنهاد شده است.

۳- مدل سیستم

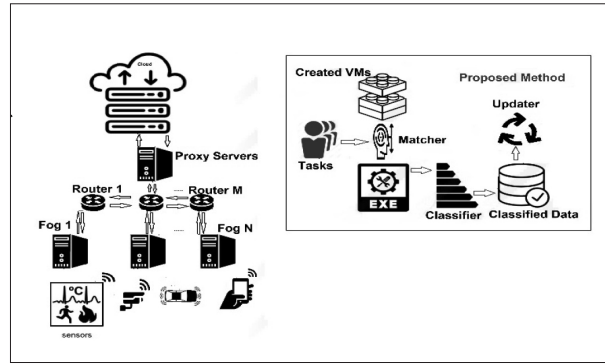
همان‌طور که در شکل ۱ مشخص است در لایه مه درخواست‌هایی از سوی دستگاه‌های انتهایی ارسال و برای اجرا در صف انتظار قرار می‌گیرند. در یک مسیر درخواست‌ها اجرا و مشخصات آنان دسته بندی و ذخیره می‌گردد و بر اساس آنان ماشین‌های مجازی مناسب به روزرسانی و ساخته می‌شوند. در مسیر دیگر تنها با مراجعه به ماشین‌های مجازی از پیش ساخته شده درخواست‌ها اجرا و نتیجه پاسخ داده می‌شود. مدل برنامه VRGAME و مؤلفه‌های آن در شکل ۲ مشخص است.

3- MQTT: Message Queuing Telemetry Transport

4- Cognitive radio-based Internet of Things



شکل ۲: مدل برنامه VRGAME در شبیه‌ساز iFogSim [۲۲]



شکل ۱: مدل سیستم پیشنهادی

۴- روش پیشنهادی

با توجه به نتایج مثبت استفاده از دسته‌بند بیزین در [۹] و همچنین با مقایسه سرعت و صحت عملکرد روش‌های مبتنی بر تئوری بیز و برخی از الگوریتم‌های یادگیری ماشین در [۲۳]، در این بخش استفاده از دسته‌بند بیزین در محیط مه به منظور زمان‌بندی وظایف بر اساس دسته‌بندی نیازمندی‌های آن‌ها مانند نیازمندی‌های پردازشی و مؤلفه‌های مقصد پیشنهاد شده است. روش پیشنهادی شکل ۱، در قالب دو الگوریتم توضیح داده شده است. در این روش $TS = \begin{bmatrix} m_1 & p_1 \\ \vdots & \vdots \\ m_n & p_n \end{bmatrix}$ آرایه‌ای دوبعدی است که بعد اول آن نام مؤلفه‌های مقصد وظایف و بعد دوم نیازمندی پردازشی آنان است. آرایه دوبعدی $CT = \begin{bmatrix} m_1 & p_1 \\ \vdots & \vdots \\ m_i & p_j \end{bmatrix}$ نتیجه اجرای دسته‌بندی بیزین روی TS است و مشخصات ابعاد آن مشابه TS است. آرایه یک‌بعدی حاوی شمارنده‌های انواع ماشین‌مجازی در ابر، $MDMC^8$ آرایه یک‌بعدی حاوی شمارنده‌های انواع ماشین‌مجازی در دستگاه‌های موبایل، dmn^9 نشان‌دهنده نام مؤلفه مقصد وظیفه، rm^10 توان پردازشی درخواست شده توسط وظیفه، nrc^{11} نشان‌دهنده تعداد وظایف در حال اجرا روی ماشین‌مجازی و $cetc^{12}$ تعداد وظایف اجرا شده از ابتدا تاکنون است.

- 5- TS: Tasks Specification
- 6-CT: Classified Tasks
- 7- CDMC: Cloud Destination Module Counters
- 8- MDMC: Mobile Destination Module Counters
- 9- dmn: Destination Module Name
- 10- rm: Requested MIPS
- 11- nrc: Number of Running Cloudlets
- 12- cetc: Current Executed Tasks Counter

۴-۱ دسته‌بند بیزین

در گام ۱۳ الگوریتم شماره ۱ دسته‌بند بیز بر روی آرایه دوبعدی TS اجرا می‌شود. آرایه‌های M و P به ترتیب ستون اول و دوم آرایه دوبعدی TS هستند که به صورت $M = TS \begin{bmatrix} m_1 \\ \vdots \\ m_n \end{bmatrix}$ و $P = TS \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}$ تعریف می‌شوند. با فرمول شماره (۱) اندیس‌های i' و j' که نشان‌دهنده تطابق میان نام مؤلفه و مقدار پردازش موردنیاز وظیفه اجرا شده هست به دست می‌آید.

$$(i', j') = \operatorname{argmax} \left(p(M_i | P_j) = \frac{p(P_j | M_i) \times p(M_i)}{p(P_j)} \right) \quad (1)$$

در (۱) $p(M_i | P_j)$ نشان‌دهنده احتمال تطبیق مؤلفه M_i به شرط نیازمندی پردازشی P_j هست. $p(P_j | M_i)$ احتمال تطبیق نیازمندی پردازشی P_j به شرط مؤلفه M_i و $p(M_i)$ احتمال وجود مؤلفه M_i نسبت به کل مؤلفه‌ها و $p(P_j)$ احتمال وجود نیازمندی پردازشی P_j نسبت به کل نیازمندی‌ها هست. در (۲) بر اساس اندیس‌های (i', j') به دست آمده از (۱)، مدخل‌های CT مقداردهی می‌شوند. در (۳) مقدار k از ۱ تا n و مقدار h از ۲ تا n و تعداد عناصر موجود در CT است و با مقایسه هر ردیف با سایر ردیف‌ها در صورت تطابق یک‌به‌یک مدخل‌ها ردیف‌های تکراری از CT حذف می‌شوند و در نهایت دسته‌بندی وظایف صورت می‌گیرد. با توجه به این‌که پیچیدگی زمانی (۱) و (۳) هر دو $O(n^2)$ است عملیات دسته‌بندی در زمان چندجمله‌ای $O(n^2)$ انجام می‌شود.

$$CT_{(i' \ 2)} = P_{j'} \text{ و } CT_{(i' \ 1)} = M_{i'} \quad (2)$$

if $CT_{(k-1)} == CT_{(h-1)}$ and $CT_{(k-2)} == CT_{(h-2)}$ then
remove CT_h from CT

۴-۲ دسته‌بندی وظایف

در الگوریتم ۱ وظایف بر اساس مؤلفه (ماشین‌مجازی) مقصد و نیازمندی پردازشی آنان دسته‌بندی می‌شوند. ورودی این الگوریتم شامل شناسه برنامه، شناسه بروکر، تعداد دپارتمان، تعداد موبایل در هر دپارتمان، حالت استقرار در ابر و آرایه دوبعدی TS است و خروجی آن آرایه دوبعدی CT هست که پس از اجرای دسته‌بند بیزین روی آرایه دوبعدی TS به دست می‌آید.

۴-۲-۱ مرحله آموزش

در گام اول الگوریتم ۱، برنامه و بروکر ایجاد می‌شوند. در گام ۲ تا ۷ نگاشت مؤلفه‌های برنامه بر روی دستگاه‌ها انجام می‌شود. اگر حالت استقرار در ابر باشد به ازای تعداد دپارتمان در تعداد موبایل در هر دپارتمان مؤلفه‌های Co-ordinator و Concentration_Calculator^{۱۳} در ابر مستقر می‌شوند و در غیر این صورت فقط مؤلفه Coordinator روی ابر مستقر می‌شود. سپس به ازای تمام موبایل‌ها مؤلفه Client روی آن‌ها مستقر می‌شود. در گام ۸ شبیه‌ساز اجرا می‌شود. از گام ۹ تا ۱۲ تا زمانی که وظیفه‌ای در صف وجود دارد، وظایف اجرا و مشخصات آنان مانند مؤلفه مقصد و میزان پردازنده درخواستی در آرایه دوبعدی TS درج می‌شود و پس از آن که اگر وظیفه‌ای برای اجرا وجود نداشت در گام ۱۳ دسته‌بند بیزین روی آرایه دوبعدی TS اجرا و خروجی آن در آرایه دوبعدی CT ذخیره می‌شود و در پایان در گام ۱۴ شبیه‌ساز متوقف می‌شود. روندنمای ۱ مرحله آموزش را ترسیم نموده است.

۴-۲-۲ مرحله آزمایش

در مرحله آزمایش از الگوریتم ۱ در گام ۱۵ متغیرهای شمارنده وظایف اجراشده و شمارنده iteration^{۱۴}

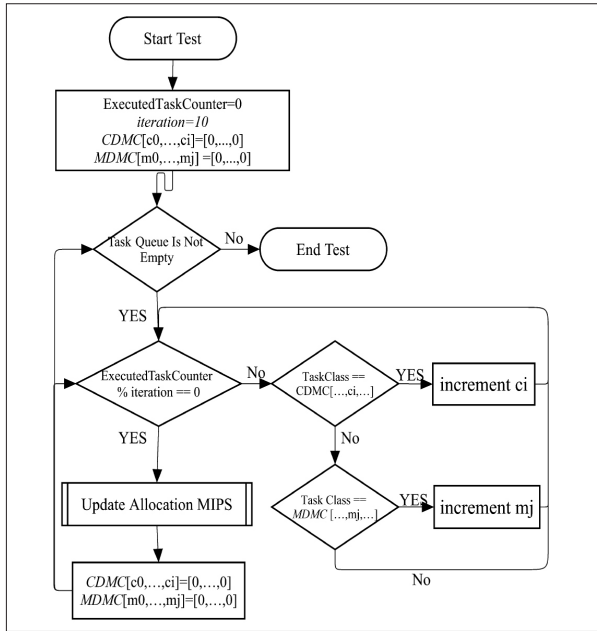
مقداردهی اولیه می‌شوند. شمارنده وظایف اجراشده تعداد وظایفی که از ابتدای مرحله آزمایش تاکنون اجرا شده‌اند را می‌شمارد و مقدار اولیه آن صفر است و شمارنده iteration مشخص می‌کند عملیات به‌روزرسانی پس از اجرای چه تعداد وظیفه انجام شود. در شبیه‌سازی iteration با ۱۰ مقداردهی اولیه شده است. در گام ۱۶ با توجه به دسته‌بندی صورت گرفته در گام ۱۳، دو آرایه تعریف می‌شوند که حاوی شمارنده‌هایی هستند که تعداد وظایفی که مؤلفه مقصد آنان با دسته‌بندی‌های حاصل از گام ۱۳ تطابق دارد شمارش می‌شوند. در گام ۱۷ تا ۲۱ تا زمانی که وظیفه‌ای در صف وجود دارد و تعداد وظایف اجراشده برابر iteration باشد تخصیص پردازنده به مؤلفه‌ها به‌روزرسانی می‌شود و مقادیر شمارنده‌های موجود در دو آرایه CDMC و MDMC با صفر مقداردهی می‌شوند. اگر تعداد وظایف اجراشده برابر iteration نباشد در گام‌های ۲۲ و ۲۳ مقدار شمارنده‌های موجود در آرایه‌های CDMC و MDMC منطبق با نوع دسته‌بندی وظیفه اجراشده افزایش می‌یابند تا مشخص گردد در تعداد iteration وظیفه اجراشده قبلی از هر دسته چه تعدادی وجود داشته است و بر اساس این مقادیر، تابع به‌روزرسانی تخصیص‌ها و مقادیر را به‌روزرسانی کند. روندنمای ۲ مرحله آزمایش را ترسیم نموده است.

۴-۳ تخصیص منابع

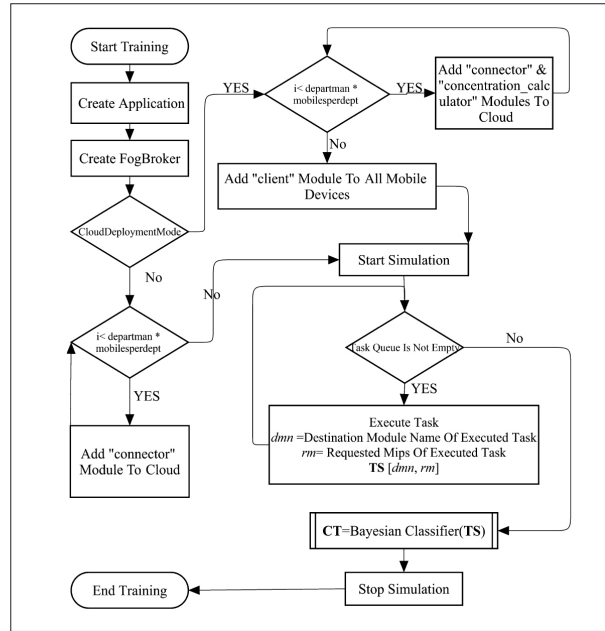
الگوریتم ۲ به‌روزرسانی تخصیص پردازنده به مؤلفه‌های موجود بر روی دستگاه را بر عهده دارد. ورودی آن شامل آرایه دوبعدی CT حاصل از الگوریتم ۱، شمارنده وظایف اجراشده، iteration، متغیر availablemips^{۱۵}، آرایه‌های CDMC و MDMC و خروجی آن مؤلفه‌های ساخته‌شده بر اساس دسته‌بندی‌های حاصل از الگوریتم ۱ است. اگر تعداد وظایف اجراشده برابر iteration باشد گام به ازای تعداد دسته‌بندی‌های حاصل از الگوریتم ۱ که دستگاه مقصد آنان ابر یا مه باشد و منطبق با مقدار شمارنده موجود در آرایه‌های CDMC و MDMC چنانچه

۱۵-میزان پردازنده باقی‌مانده براساس میلیون دستوراتعمل بر ثانیه

۱۳- مؤلفه محاسبه‌کننده میزان سطح تمرکز کاربر
۱۴-فاصله زمانی میان هر دو عملیات بروزرسانی



روندنامی ۲: مرحله آزمایش



روندنامی ۱: مرحله آموزش

$$Cost = PEC + (CC - LUUT) * RPM * LU * TM \quad (4)$$

در (۴)، PEC برابر هزینه اجرای قبلی در ابزار مه، CC برابر ساعت کلودسیم، LUUT برابر آخرین زمان به روزرسانی بهره‌وری سیستم، RPM برابر نرخ MIPS، LU برابر آخرین میزان بهره‌وری و TM برابر میزان MIPS کلی برای میزبان

است. مصرف کلی شبکه بر اساس (۵) است.

$$NetworkUsage = \frac{\sum_{i=1}^n (TL_i * TS_i)}{MST} \quad (5)$$

$$Energy = CEC + (NT - LUUT) * HLU \quad (6)$$

روابط بین مؤلفه‌ها به وسیله داده مخابره شده مؤلفه‌ها^{۱۱} مشخص می‌شود و مصرف منابع شبکه به اندازه داده مخابره شده مؤلفه‌های منتقل شده در یک زمان معین بستگی دارد. در (۵)، TL_i و TS_i برابر مجموع تأخیرها و مجموع اندازه داده مخابره شده مؤلفه‌ها، N برابر تعداد کل داده مخابره شده مؤلفه‌ها و MST برابر بیشترین زمان شبیه‌سازی است. همچنین، انرژی مصرفی شبیه‌سازی برای همبندی کامل شبکه به وسیله (۶) محاسبه می‌گردد در (۳)، CEC برابر انرژی مصرفی جاری، NT برابر زمان فعلی، LUUT برابر زمان آخرین به روزرسانی بهره‌وری سیستم و HLU برابر میزان آخرین بهره‌وری میزبان است. شبیه‌ساز در محیط jdk، eclipse نسخه ۲۰۱۸، ۱.۷.

دستگاه مقصد دارای توان پردازشی کافی باشد مؤلفه متناسب با توان پردازشی درخواستی ساخته می‌شود. سپس زمانی که تمام مؤلفه‌های درخواستی ساخته شد اگر مؤلفه وظیفه در حال اجرایی دارد یا این‌که در وضعیت اجرا و منتظر ورود وظیفه‌ای است، منطبق با توان پردازشی درخواستی، به آن پردازنده تخصیص می‌یابد و در پایان انرژی مصرفی به روزرسانی می‌شود.

۵- ارزیابی نتایج

برای تحقق توان بالقوه مدل مه و اینترنت اشیا در برنامه‌های بیدرنگ تکنیک‌هایی برای مدیریت منابع موردنیاز است [۲۴] و پردازش در لبه شبکه نقش کلیدی در اجرای برنامه‌های اینترنت اشیا دارد [۲۵]. به منظور بررسی صحت روش‌های پیشنهادی در حوزه مه و اینترنت اشیا شبیه‌ساز iFogSim [۲۳] وجود دارد و می‌تواند الگوریتم‌های پیشنهادی در این حوزه را از نظر پارامترهایی مانند انرژی مصرفی دستگاه‌ها، هزینه، پهنای باند، تأخیر و دیگر پارامترها مورد ارزیابی قرار دهد.

۵-۱ پارامترهای ارزیابی

پارامتر هزینه اجرای شبیه‌سازی بر اساس (۴) است.

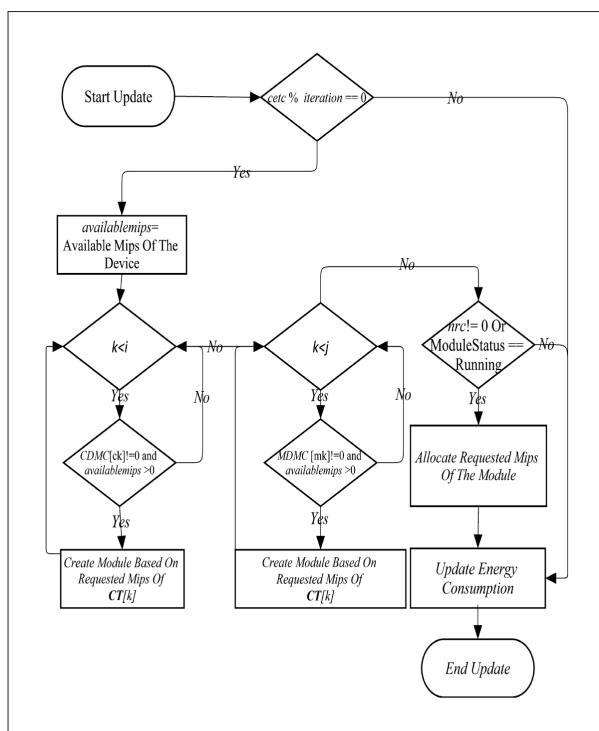
Input: AppID, BrokerID, *departman, mobilesperdept*, CloudDeploymentMode, **TS** [*dmm, rm*], **Output:** **CT** [*dmm, rm*]

```

1: Create Application, FogBroker; Module Mapping
2: if (CloudDeploymentMode) {
3:   for i=0 to i< departman * mobilesperdept {
4:     Add "coordinator" and "concentration_calculator" To Cloud }
5: Add "client" Module To All Mobile Devices } else {
6: for i=0 to i< departman * mobilesperdept {
7: } } 7: Add "coordinator " Module To Cloud
8: Start Simulation
9: while (Task Queue Is Not Empty) {
10: Execute Task; rm = Requested Mips Of Executed Task
11: dmm = Destination Module Name Of Executed Task
12: TS [dmm, rm ] }
13: CT = Bayesian Classifier(TS)
14: Stop Simulation
15: ExecutedTaskCounter, iteration
16: CDMC [c0,...,ci], MDMC [m0,...,mj]
17: while (Task Queue Is Not Empty) {
18: if (ExecutedTaskCounter% iteration == 0) {
19: Call Update Allocation Of Mips
20: CDMC [c0,...,ci] = [0,...,0]
21: MDMC [m0,...,mj] = [0,...,0] }
22: elseif (Task Class == CDMC [...,ci,...]) {
23: increment ci } elseif (Task Class == MDMC [...,mj,...]) { increment mj } }

```

الگوریتم ۱: دسته‌بندی وظایف



روندنامی ۳: به‌روزرسانی تخصیص پردازنده

Input **CT**[DestinationModuleName, RequestedMips], *cetc*, *iteration*, *availablemips*
CDMC[*c*₀,...,*c*_{*i*}], *MDMC*[*m*₀,...,*m*_{*j*}]
Output: Created Modules

```

1: if (cetc% iteration == 0) {
2: availablemips = Available Mips Of The Device
3: for k=0 to k<i {
4: if (CDMC[ck] != 0 and availablemips > 0) {
5: Create Module Based On Requested Mips Of CT[k] } }
6: for k=0 to k<j do {
7: if (MDMC [mk] != 0 and availablemips > 0) {
8: Create Module Based On Requested Mips Of CT[k] } } }
9: if (nrc != 0 Or ModuleStatus == Running) {
10: Allocate Requested Mips Of The Module }
11: Update Energy Consumption

```

الگوریتم ۲: به‌روزرسانی تخصیص پردازنده

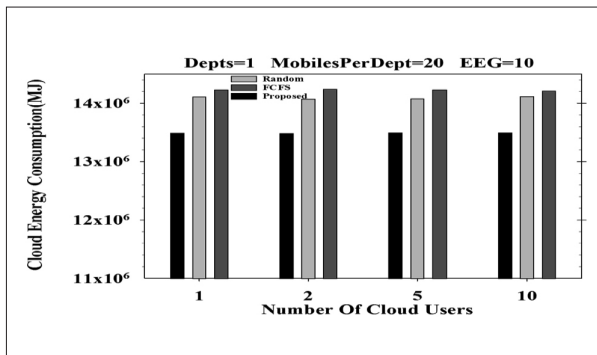
۶- نتیجه‌گیری و پیشنهادها

مه با کاهش فاصله زیاد میان فراهم‌کنندگان پردازش ابری و دستگاه‌های انتهایی زمان پاسخ در سیستم‌های بیدرنگ را کاهش داده است. هوشمندسازی زمان‌بندی، موجب کاهش هزینه‌های مدیریتی می‌شود. به همین دلیل روش زمان‌بندی مبتنی بر یادگیری ماشین بر پایه دسته‌بندی بیزین پیشنهاد شده است. نتایج نشان می‌دهند که میزان انرژی مصرفی در ابر حداقل ۴٫۴۳٪ و حداکثر ۶٫۹۴ درصد

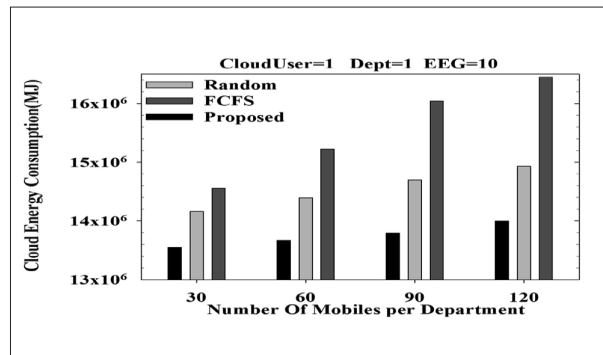
ماشینی با پردازنده Intel(R)-i5-5500U-2,6GHz، حافظه ۴ گیگابایتی، کارت شبکه Ethernet802.3 و سیستم‌عامل MicrosoftWindows7-x64 اجرا شده است و شکل‌های ۳ تا ۱۴ نتایج حاصل از شبیه‌سازی را نشان می‌دهند. در جدول شماره ۱ درصدهای بهبود انرژی مصرفی و هزینه اجرای وظایف در ابر و انرژی مصرفی در دستگاه‌های موبایل در روش پیشنهادی نسبت به الگوریتم‌های FCFS و تصادفی مقایسه شده است.

جدول ۱: درصد بهبود عملکرد روش پیشنهادی نسبت به روش‌های به ترتیب ورود و تصادفی

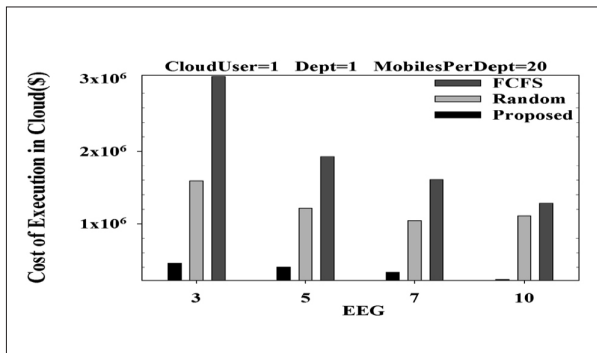
| پارامتر | نرخ ۴،۴۳ درصدی به نسبت تصادفی انرژی مصرفی در ابر روش | نرخ ۶،۹۴ درصدی به نسبت تصادفی انرژی مصرفی در ابر روش | نرخ ۴،۲ درصدی به نسبت تصادفی نرخ ۴،۲ درصدی به نسبت تصادفی | نرخ ۴،۲۹ درصدی به نسبت تصادفی نرخ ۴،۲۹ درصدی به نسبت تصادفی | نرخ ۷۱،۰۷ درصدی به نسبت تصادفی نرخ ۷۱،۰۷ درصدی به نسبت تصادفی | نرخ ۸۱،۹۳ درصدی به نسبت تصادفی نرخ ۸۱،۹۳ درصدی به نسبت تصادفی |
|------------|---|---|--|--|--|--|
| درصد بهبود | ۴،۴۳ | ۶،۹۴ | ۴،۲ | ۴،۲۹ | ۷۱،۰۷ | ۸۱،۹۳ |



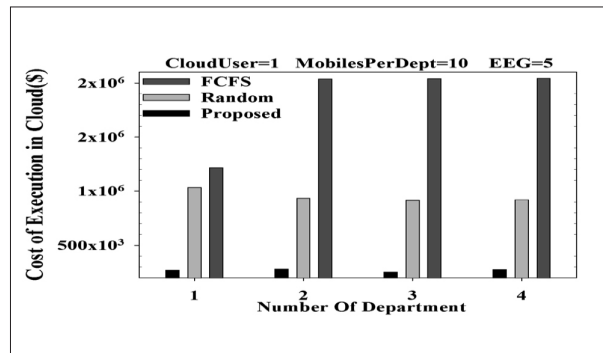
شکل ۴: انرژی مصرفی ابر به ازای EEG مختلف



شکل ۳: انرژی مصرفی ابر به ازای تعداد دپارتمان مختلف



شکل ۶: انرژی مصرفی ابر به ازای تعداد کاربر ابر مختلف



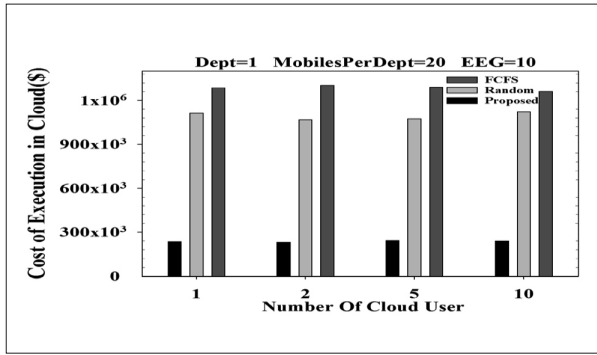
شکل ۵: انرژی مصرفی ابر به ازای تعداد موبایل مختلف

تقویتی و اجرای الگوریتم‌های مختلف در شرایط متفاوت، مدل‌های یادگیری مناسب را ایجاد کنیم و با استفاده از این مدل‌ها سیستم به مرور زمان یاد بگیرد که در شرایط مختلف استفاده از چه روشی نتیجه بهتری را خواهد داشت و هوشمندی در انتخاب روش نیز در نظر گرفته شود.

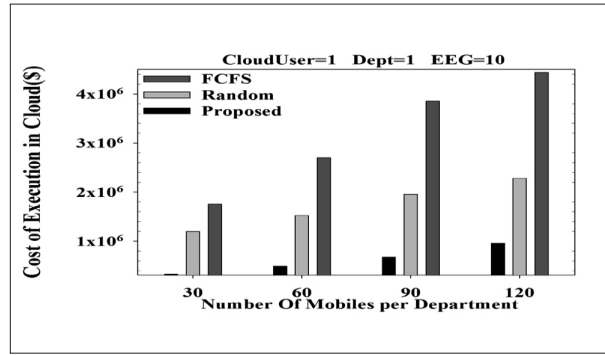
مراجع

- Bogale, Tadiilo E., Wang, X., Le, Long B., "Machine Intelligence Techniques for Next-Generation Context-Aware Wireless Networks.", arXiv preprint arXiv:1801.04223, 2018.
- Peralta, G., Iglesias-Urika, M., Barcelo, M., Gomez, R.,

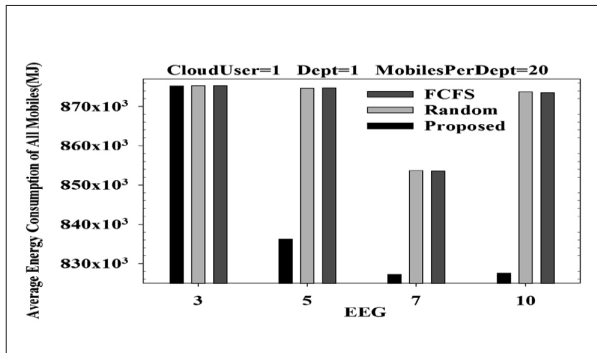
و هزینه اجرای وظیفه در ابر حداقل ۷۱،۰۷ و حداکثر ۸۱،۹۳ درصد و میانگین انرژی مصرفی دستگاه‌های موبایل ۴،۳ درصد نسبت به الگوریتم‌های تصادفی و FCFS بهبود یافته است و تأثیر منفی بر پهنای باند مصرفی و زمان اجرا نداشته است. مه و برنامه‌های کاربردی آن دارای پویایی بسیاری هستند و تغییر دائمی شرایط، جز لاینفک آن است. لذا پیشنهاد می‌شود الگوریتم‌های یادگیری مختلف در شرایط محیطی متفاوتی استفاده شوند و هوشمندی در انتخاب بهترین الگوریتم نیز در نظر گرفته شود. در آینده در نظر داریم تا به وسیله روش‌هایی چون یادگیری عمیق،



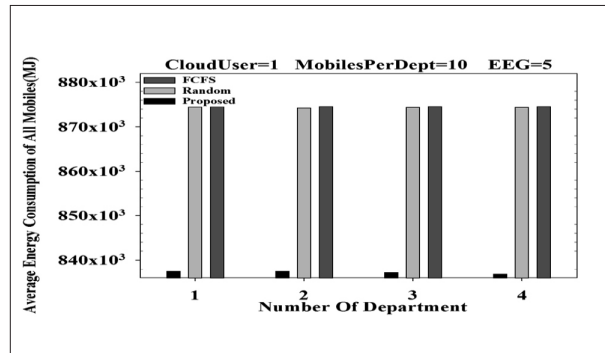
شکل ۸: هزینه اجرا در ابر به ازای EEG مختلف



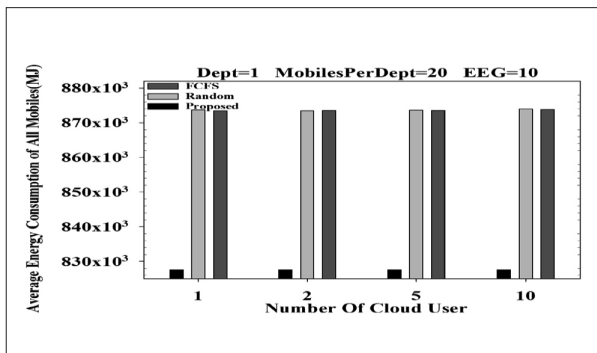
شکل ۷: هزینه اجرا در ابر به ازای تعداد دپارتمان مختلف



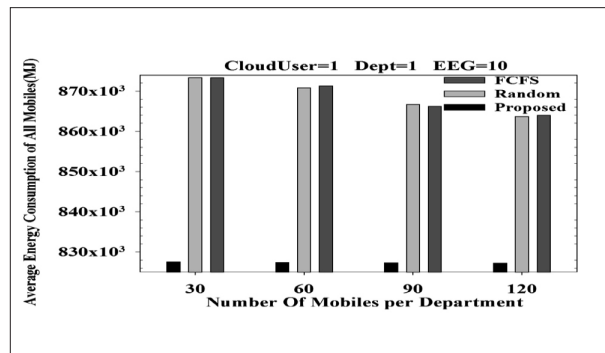
شکل ۱۰: هزینه اجرا در ابر به ازای تعداد کاربر ابر متفاوت



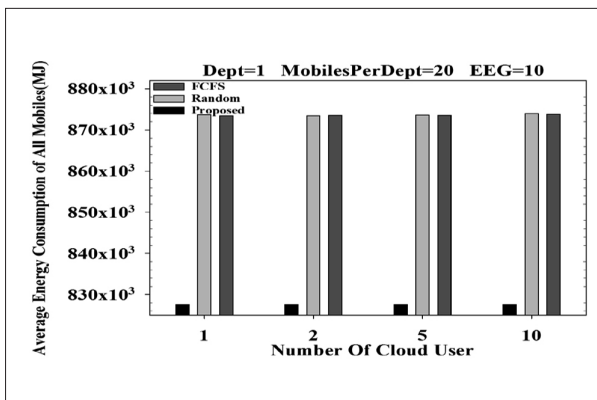
شکل ۹: هزینه اجرا در ابر به ازای تعداد موبایل متفاوت



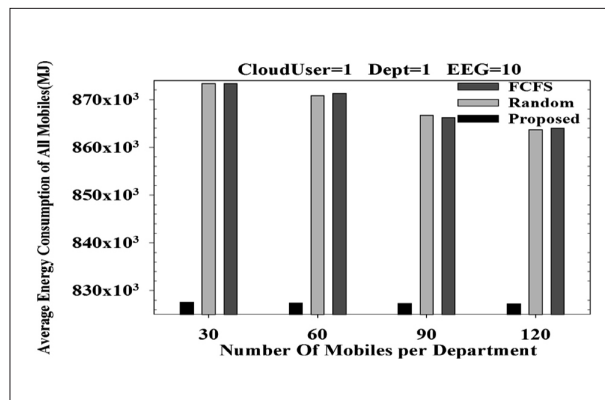
شکل ۱۲: میانگین انرژی مصرفی موبایل ها به ازای EEG متفاوت



شکل ۱۱: میانگین انرژی مصرفی موبایل ها در تعداد دپارتمان متفاوت



شکل ۱۴: میانگین انرژی مصرفی موبایل ها در کاربر ابر متفاوت



شکل ۱۳: میانگین انرژی مصرفی موبایل ها به ازای دستگاه موبایل متفاوت

- telligent Resource Scheduling at Scale: a Machine Learning Perspective.”, In Service-Oriented System Engineering (SOSE), IEEE Symposium on, pp. 132-141, 2018.
15. Cui, D., Peng, Z., Xiong, J., Xu, B., Lin, W., “A Reinforcement Learning-based Mixed Job Scheduler Scheme for Grid or IaaS Cloud.”, IEEE Transactions on Cloud Computing, 2017.
 16. Hormozi, E., Hormozi, H., Akbari, Mohammad K., Sargolzaei Javan, M., “Using of machine learning into cloud environment (a survey): managing and scheduling of resources in cloud systems.”, In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Seventh International Conference on, pp. 363-368, 2012.
 17. Ezugwu, Absalom E., Frincu, Marc E., Adewumi, Aderemi O., Buhari, Seyed M., Juniadu, Sahalu B., “Neural network-based multi-agent approach for scheduling in distributed systems.”, Concurrency and Computation: Practice and Experience, vol. 29, no. 1, p. e3887, 2017.
 18. Lavassani, M., Forsström, S., Jennehag, U., Zhang, T., “Combining Fog Computing with Sensor Mote Machine Learning for Industrial IoT.”, Sensors, vol. 18, no. 5, p. 1532, 2018.
 19. Memon, S., M. Maheswaran. “Using machine learning for handover optimization in vehicular fog computing.” In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 182-190, 2019.
 20. Zhang, Q., Lin, M., Yang, Laurence T., Chen, Z., Li, P., “Energy-efficient scheduling for real-time systems based on deep Q-learning model”, IEEE Transactions on Sustainable Computing, 2017.
 21. Zhu, J., Song, Y., Jiang, D., Song, H., “A new deep-Q-learning-based transmission scheduling mechanism for the cognitive Internet of Things.”, IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2375-2385, 2018.
 22. Gupta, H., Dashtjerdi, Amir V., Ghosh, Soumya K., Rajkumar, B., “iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Software: Practice and Experience”, vol. 47, no. 9, pp. 1275-1296, 2017.
 23. Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., Akinjobi, J., “Supervised Machine Learning Algorithms: Classification and Comparison”, International Journal of Computer Trends and Technology (IJCTT), vol. 48, no. 3, pp. 128-138, 2017.
 24. Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalahi, F., Niakanlahiji, A., Kong, J., Jue, P.J., “All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey.”, arXiv preprint arXiv: 1808.05283, 2018.
 25. Shurman, Mohammad M., Aljarah, Maha K., “Collaborative execution of distributed mobile and IoT applications running at the edge.”, In Electrical and Computing Technologies and Applications (ICECTA), International Conference on, pp. 1-5, 2017.
 3. Moran, A., Bilbao, J., “Fog computing based efficient IoT scheme for the Industry 4.0.”, In IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), pp. 1-6, 2017.
 4. Stojmenovic, I., Wen, S., “The fog computing paradigm: Scenarios and security issues.”, In Computer Science and Information Systems (FedCSIS), Federated Conference on, pp. 1-8, 2014.
 4. Rabiul Alam, Md G., Tun, Yan K., Hong, Choong S., “Multi-agent and reinforcement learning based code offloading in mobile fog.”, In Information Networking (ICOIN), International Conference on, pp. 285-290. IEEE, 2016.
 5. Etemad, M., Aazam, M., St-Hilaire, M., “Using DEVS for modeling and simulating a Fog Computing environment.”, In Computing, Networking and Communications (ICNC), International Conference on, pp. 849-854, 2017.
 6. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. CISCO, April 2015, https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf
 7. Tang, Z., Zhou, X., Zhang, F., Jia, W., Zhao, W., “Migration Modeling and Learning Algorithms for Containers in Fog Computing.”, IEEE Transactions on Services Computing, 2018.
 8. Kiani, A., N. Ansari, A. Khreishah. “Hierarchical capacity provisioning for fog computing.” IEEE/ACM Transactions on Networking. vol. 27, no. 3, pp. 962-971, 2019.
 9. Zhang, P., Zhou, M., “Dynamic cloud task scheduling based on a two-stage strategy.”, IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp. 772-783, 2018.
 10. Fei, X., N. Shah, N. Verba, K. Chao, V. Sanchez-Anguix, J. Lewandowski, A. James, Z. Usman. “CPS data streams analytics based on machine learning for Cloud and Fog Computing: A survey.” Future Generation Computer Systems, vol. 90, pp. 435-450, 2019.
 11. Mathew, T., Sekaran, K S., Jose, J., “Study and analysis of various task scheduling algorithms in the cloud computing environment.”, In Advances in Computing, Communications and Informatics (ICACCI), International Conference on, pp. 658-664, 2014.
 12. Anusha Bamini, B.E., Sharmini Enoch, M.E., “Optimized Resource Scheduling Using Classification and Regression Tree and Modified Bacterial Foraging Optimization Algorithm”, International Journal of Applied Engineering Research, vol. 10, no. 16, pp. 37170-37175, 2015.
 13. Borthakur, D., Dubey, H., Constant, N., Mahler, L., Mankodiya, K., “Smart fog: Fog computing framework for unsupervised clustering analytics in wearable internet of things.”, In Signal and Information Processing (GlobalSIP), IEEE Global Conference on, pp. 472-476. IEEE, 2017.
 14. Yang, R., Ouyang, X., Chen, Y., Townend, P., Xu, J., “In-