مقاله پژوهشی

تخلیه امن مبتنی بر یادگیری مشارکتی در محیط رایانش مه با استفاده از شبکه نرم‌افزارمحور

محمدرضا شرفی هویدا، محمدرضا ملاحسینی اردکانی و وحید آیت‌اللهی تفتی

چکیده: در اینترنت اشیا به دلیل حجم بالای داده‌ها، چالش‌های قابل توجهی در پردازش و ذخیره‌سازی اطلاعات هست که نیاز به توجه دارد؛ از جمله تأخیر، آگاهی از مکان و پشتیبانی از تحرک در زمان واقعی. محاسبات مه به عنوان یک راهکار مؤثر برای این چالش‌ها شناخته می‌شود. این مقاله به بررسی روش‌های مختلف تخلیه امن مبتنی بر یادگیری مشارکتی در محیط رایانش مه با استفاده از شبکه نرم‌افزارمحور می‌پردازد و چهار روش بهینه‌سازی شامل SDN، SA+GA، OLB-LBMM و Round-Robin را مورد تحلیل و مقایسه قرار می‌دهد. هدف اصلی این تحقیق، بهبود عملکرد و امنیت در فرایند تخلیه داده‌ها با توجه به چالش‌های موجود است. روش شبکه نرم‌افزارمحور به عنوان یک چارچوب منعطف برای مدیریت منابع و داده‌ها در شبکه‌های اینترنت اشیا، عملکرد بهتری نسبت به سایر روش‌ها ارائه می‌دهد و با کاهش تأخیر و بهینه‌سازی تخصیص منابع، موجب افزایش رضایت کاربران و درآمد ارائه‌دهندگان خدمات ابری می‌شود. همچنین الگوریتم‌های SA+GA و OLB-LBMM بهبودهایی در کارایی و امنیت ارائه می‌دهند، اما با چالش‌هایی در تأخیر و پیچیدگی محاسباتی مواجه هستند. نتایج نشان می‌دهند که یادگیری مشارکتی در ترکیب با شبکه نرم‌افزارمحور می‌تواند بهبود قابل توجهی در تخلیه امن داده‌ها و مدیریت منابع شبکه ایجاد کند.

کلیدواژه: **شبکه نرم‌افزارمحور، یادگیری مشارکتی، محاسبات مه، اینترنت اشیا.**

۱- مقدمه

پیشرفت سریع اینترنت اشیا[1] و برنامه‌های شبکه‌های اجتماعی منجر به رشد نمایی داده‌های تولیدشده در لبه شبکه شده است. پیش‌بینی می‌شود که نرخ تولید داده‌ها در آینده نزدیک از ظرفیت فعلی اینترنت فراتر خواهد رفت [۱]. با توجه به رشد نمایی داده‌های تولیدشده در لبه شبکه، پیش‌بینی می‌شود که بیش از ۹۰ درصد از داده‌ها به‌صورت محلی پردازش شوند [۲]. این داده‌ها نیاز به منابع محاسباتی قوی برای پردازش بی‌درنگ دارند و این امر می‌تواند منجر به مصرف انرژی بالا بر روی تجهیزات محدود منابع IoT شود. برای حل این مشکل، تخلیه وظیفه با استفاده از محاسبات لبه به‌عنوان یک راه‌حل امیدوارکننده مطرح گردیده است [۳] و [۴].

۱-۱ چالش‌ها و نیازها

در این راستا، محاسبات مه به‌عنوان یک راهکار مؤثر شناخته می‌شود. این مدل به‌عنوان یک الگوی برنامه‌نویسی و ارتباطی تعریف می‌شود. این مدل منابع ابری را به‌صورت فیزیکی به دستگاه‌های اینترنت اشیا نزدیک می‌کند؛ به عبارت دیگر، محاسبات مه به‌عنوان رابطی بین ابر و اینترنت اشیا عمل می‌کند و به آنها در برقراری ارتباط کمک می‌کند. با گسترش کاربرد محاسبات ابری و افزایش دسترسی به منابع در اینترنت اشیا، این فناوری به‌خوبی از امکانات موجود بهره‌برداری می‌کند. تخلیه امن داده‌ها در رایانش مه یکی از چالش‌های اصلی است که نیاز به بررسی دقیق دارد [۵].

۱-۲ راه‌حل‌ها

استفاده از یادگیری مشارکتی[2] در ترکیب با شبکه نرم‌افزارمحور[3] می‌تواند به‌عنوان یک راهکار مؤثر برای بهبود تخلیه امن داده‌ها در محیط‌های رایانش مه عمل کند [۶]. طرح‌های تخلیه بار در مقالات موجود عمدتاً بر توان محاسباتی و مصرف انرژی متمرکز شده‌اند، بدون آنکه بار شبکه بر روی مسیر بین دستگاه و کارگزار لبه را در نظر بگیرند؛ بنابراین بار دینامیکی شبکه باید در حین تصمیم‌گیری تخلیه مد نظر قرار گیرد. این امر با اتخاذ معماری شبکه نرم‌افزارمحور که کنترل متمرکز منطقی و نمای کلی شرایط شبکه را فراهم می‌کند، قابل دستیابی است و همچنین به بهینه‌سازی تخصیص منابع در محیط‌های رایانش مه کمک می‌کند [۷]. همچنین مدیریت شبکه توسط SDN امکان جمع‌آوری اطلاعات شبکه از دستگاه‌های بی‌سیم ناهمگن را در میان فناوری‌های مختلف فراهم می‌کند. به این ترتیب، کنترل‌کننده SDN قادر به اتخاذ تصمیمات بهینه در زمینه تخلیه وظایف خواهد بود.

۱-۳ اهداف تحقیق

موضوع بهینه‌سازی به بیشینه‌سازی و کمینه‌سازی مربوط می‌شود. هدف از بهینه‌سازی به‌طور دقیق‌تر، یافتن مقادیر متغیرهایی است که مقدار یک تابع مشخص را به حداکثر یا حداقل برسانند. در بسیاری از موارد، متغیرها باید شرایط جانبی مانند معادلات یا نابرابری‌ها را برآورده کنند که در این صورت اصطلاح بهینه‌سازی مقید به‌درستی به کار می‌رود. در غیر این صورت، مسئله بهینه‌سازی نامقید نامیده می‌شود [۸].

این مقاله در تاریخ ۱۵ آبان ماه ۱۴۰۳ دریافت و در تاریخ ۱ فروردین ماه ۱۴۰۴ بازنگری شد.

محمدرضا شرفی هویدا، گروه کامپیوتر، واحد میبد، دانشگاه آزاد اسلامی، میبد، ایران، (email: mohamadreza_sharafi@yahoo.com).

محمدرضا ملاحسینی اردکانی (نویسنده مسئول)، گروه کامپیوتر، واحد میبد، دانشگاه آزاد اسلامی، میبد، ایران، (email: mr.mollahoseini@iau.ac.ir).

وحید آیت‌اللهی تفتی، گروه کامپیوتر، واحد تفت، دانشگاه آزاد اسلامی، تفت، ایران، (email: vahid.ayat@gmail.com).

1. Internet of Things

2. Federated Learning

3. Software-Defined Network

بسیاری از مسائل بهینه‌سازی ناشی از مهندسی و علوم حاوی روابط ترکیبی و غیرخطی هستند. این گونه مسائل به‌وسیله برنامه‌نویسی غیرخطی مختلط- صحیح مدل‌سازی می‌شوند که ترکیبی از قابلیت‌های برنامه‌نویسی خطی صحیح مختلط و برنامه‌نویسی غیرخطی است. توانایی مدل‌سازی دقیق مسائل دنیای واقعی، MINLP[1] را به یک حوزه تحقیقاتی فعال با کاربردهای صنعتی متنوع تبدیل کرده است [۹].

۱-٤ نوآوری‌های یادگیری مشارکتی

یادگیری مشارکتی به عنوان یک فناوری نوظهور [۱۰]، توجه بسیاری از محققان را جلب کرده است تا پتانسیل و کاربردهای آن را کشف کنند [۱۱] و [۱۲]. FL به دنبال پاسخ به این پرسش اصلی است که آیا می‌توانیم مدل را بدون نیاز به انتقال داده‌ها به یک مکان مرکزی آموزش دهیم [۱۳]. در چارچوب FL، تمرکز بر همکاری است که همیشه از طریق الگوریتم‌های یادگیری ماشین استاندارد حاصل نمی‌شود [۱۴]. علاوه بر این، FL به الگوریتم‌ها اجازه می‌دهد تا تجربه کسب کنند که این نیز چیزی است که همیشه از طریق روش‌های یادگیری ماشین سنتی تضمین نمی‌شود [۱۵] و [۱۶]؛ بنابراین ما یک طرح آموزشی توزیعی مبتنی بر یادگیری مشارکتی را پیشنهاد می‌کنیم تا بار آموزشی را بر روی هر دستگاه کاهش دهیم.

۱-۵ ساختار مقاله

در این مقاله به مسئله چگونگی استفاده مؤثر از منابع محاسباتی و ارتباطی محدود در لبه برای بهینه‌سازی عملکرد یادگیری می‌پردازیم. ما یک معماری محاسبات لبه معمولی را در نظر می‌گیریم که در آن گره‌های لبه با ابر از راه دور از طریق عناصر شبکه مانند دروازه‌ها و روترها مرتبط هستند. این ارتباط با استفاده از مدیریت شبکه توسط SDN امکان‌پذیر است و اطلاعاتی از قبیل توپولوژی، خودکارسازی شبکه، انتزاع از زیرساخت و کاهش هزینه‌های عملیاتی را از دستگاه‌های ناهمگن جمع‌آوری می‌کند؛ بنابراین کنترل‌کننده SDN قادر به اتخاذ تصمیمات بهینه در زمینه تخلیه وظایف خواهد بود و همچنین از دیگر مزایای SDN در این معماری، انعطاف‌پذیری شبکه و پیاده‌سازی مدیریت شبکه و زیرساخت‌های مبتنی بر مه است [۱۲]. در این معماری با توجه به استفاده از الگوریتم یادگیری مشارکتی، باید سناریویی را در نظر بگیریم که در آن کارهای تخلیه بار باید از طریق مسیرهای چندگانه به سمت دستگاه‌های لبه انجام شود.

۲- کارهای مرتبط

محاسبات رایانش در مه که بر روی لبه‌های شبکه انجام می‌گیرد، به‌عنوان یک نوع محاسبه توزیعی در نظر گرفته می‌شود. Fog Mobile به‌عنوان یک نمونه اولیه از این نوع محاسبات، برای ارائه خدمات آگاه از تأخیر به کاربران تلفن همراه طراحی شده است. تخلیه محاسبات در تلفن همراه به دلیل نیاز به منابع موقت و وجود دستگاه‌های ناهمگن، چالشی جدی محسوب می‌شود. به همین دلیل، یک مکانیسم تخلیه محاسباتی مبتنی بر یادگیری تقویت‌کننده برای اطمینان از ارائه خدمات به مشتریان تلفن همراه ارائه شده است [۱۷]. این مکانیسم از الگوریتم یادگیری تقویتی توزیع‌شده استفاده می‌کند و هدف آن، تخلیه بلوک‌های پایه به‌صورت غیرمتمرکز است تا کدهای تلفن همراه در فضاهای توزیع‌شده جغرافیایی توزیع شوند. روش پیشنهادی به‌طور قابل توجهی منجر به کاهش زمان اجرا و تأخیر دسترسی به خدمات تلفن همراه می‌شود و در عین حال مصرف انرژی دستگاه‌های تلفن همراه را نیز کاهش می‌دهد. این طرح پیشنهادی یک روش توزیع‌شده با چند عامل برای تخلیه محاسبات در فضاهای پراکنده رایانش مه تلفن همراه ارائه می‌دهد.

یکی از رویکردهای کاربردی که در سال ۲۰۱۶ ارائه شده است [۱۸]، مدل توزیع‌شده ADMM برای تخلیه داده در رایانش مه می‌باشد که بهینه‌سازی عملکرد الگوریتم تخلیه و تضمین کیفیت خدمات برای کاربر نهایی را نتیجه می‌دهد. با افزایش قابل توجه ترافیک داده در شبکه، حجم محاسبات بالا می‌رود و این امر منجر به افزایش تأخیر می‌شود. برای حل این مسئله از رایانش مه استفاده می‌شود که می‌تواند ذخیره‌سازی، حافظه و شبکه محاسباتی را فراهم کند، اما نمی‌تواند جایگزین سرور ابری[2] شود. Fog با گسترش ابر، هدف خود را نزدیک‌ترکردن به کاربر نهایی محاسباتی قرار می‌دهد تا بهره‌وری را بهبود بخشد و میزان داده‌هایی را که برای تجزیه و تحلیل، پردازش و ذخیره‌سازی نیاز به انتقال به ابر دارند، کاهش دهد. در واقع، تعدادی از گره‌های مه وجود دارند که هر کدام دارای محدودیت‌هایی در ذخیره‌سازی هستند و تعدادی از کاربران می‌خواهند داده‌ها را به CS (سیستم ذخیره‌سازی) منتقل کنند.

در چارچوب دیگری که برای تخلیه بار در مراکز داده رایانش ابری ارائه شده است [۱۹]، فرض بر این است که مراکز داده در لبه شبکه نصب می‌شوند و اگر یک درخواست به یک مرکز داده منتقل شود، آن را با در نظر گرفتن یک احتمال (p_i) به یک مرکز داده همسایه هدایت می‌کند. فرض بر این است که مراکز داده دارای تعداد زیادی سرور هستند و ترافیک در برخی از آنها سبب اشباع می‌شود. در این حالت، مراکز داده ممکن است با پذیرش برخی از درخواست‌های ردشده به این مسئله کمک کنند. هدف دستیابی به کیفیت از طریق همکاری میان مراکز داده همسایه است. در رایانش مه، مراکز داده با ظرفیتی کمتر از مواردی که در محاسبات ابری وجود دارد، نیاز است و بنابراین احتمال فرسودگی بیشتری دارند؛ از این رو برای کاهش احتمال مسدودشدن درخواست‌ها، مراکز داده در محاسبات رایانش مه باید همکاری کنند. به‌عنوان مثال، زمانی که سرور یک درخواست را نمی‌تواند اجرا کند، باید یکی دیگر از آنها را جایگزین کند و ممکن است درخواست به یکی دیگر ارسال شود. همچنین اگر مراکز را در لبه یک شبکه حلقه‌ای در نظر بگیریم، احتمال ارسال درخواست به یک مرکز داده همسایه $i-۱$ یا $i+۱$ بیشتر خواهد بود. بخش اصلی تحلیل مربوط به تأثیر همکاری دو مرکز داده است. برای $i \in \{۱,۲\}$، روند ورود درخواست‌ها به مرکز داده i با توزیع پواسون و پارامتر λ_i توصیف می‌شود. اگر یکی از سرورهای C_i پس از ورود درخواست بیکار باشد، درخواست توسط این مرکز داده پردازش می‌شود و در غیر این صورت، اگر مرکز داده همه اشباع باشند، با احتمال p_i، درخواست به مرکز داده‌های دیگر نیز ارسال می‌شود.

چالش اصلی در سرویس‌های ابر برای تلفن همراه، به حداقل رساندن زمان انتقال داده‌ها و اجرای وظایف کاربر نهایی است. این چالش نیاز به سیاست‌هایی دارد که بتوانند برای تخلیه محاسباتی تصمیم‌گیری کنند [۲۰]. در این مدل، چهار عامل مهم وجود دارد: کاربر، عامل، ابر محلی و ابر از راه دور. کاربران تلفن همراه با استفاده از ابر از طریق دستگاه‌های خود که به‌وسیله نقاط دسترسی WiFi یا اینترنت به ابر دسترسی دارند، تعامل می‌کنند. هنگامی که یک برنامه پیچیده اجرا می‌شود، دستگاه‌های

1. Mixed-Integer Linear Programming

2. Cloud Server

موبایل با محدودیت‌هایی مواجه می‌شوند و باید از سیستم ابر کمک بگیرند که به‌طور مستقیم توسط کاربران قابل دسترسی است و می‌تواند از تأخیرهای ارتباطی اضافی جلوگیری کند. ابر از راه دور می‌تواند منابع محاسباتی فراوانی فراهم کند، اما به دلیل فاصله فیزیکی از کاربران، باعث تأخیر قابل توجهی در انتقال می‌شود. عامل به‌عنوان یک مرکز کنترل عمل می‌کند که می‌تواند اطلاعات منابع را جمع‌آوری کند. بر اساس این اطلاعات، عامل تصمیم می‌گیرد که وظایف بر روی دستگاه تلفن همراه اجرا شوند، به ابر محلی یا به ابر از راه دور تخلیه شوند؛ بنابراین هنگامی که یک کار جدید توسط یک کاربر ایجاد می‌شود، می‌توان آن را بر روی دستگاه تلفن همراه اجرا کرد. اگر دستگاه تلفن همراه قادر به انجام این کار نباشد، عامل تصمیم می‌گیرد که این کار را از طریق لینک بی‌سیم به ابر محلی ارسال کند. اگر ابر محلی نتواند این وظیفه را انجام دهد یا زمان‌بر باشد، این وظیفه از طریق اینترنت به ابر از راه دور ارسال می‌شود.

یکی دیگر از مطالعات انجام‌شده در زمینه تخلیه بار محاسبات [۲۱] بیان می‌کند که با افزایش روابط اجتماعی، انرژی مصرفی دستگاه‌های تلفن همراه افزایش می‌یابد. این موضوع می‌تواند در طراحی تخلیه محاسباتی رایانش در مه با هدف به حداقل رساندن هزینه‌های اجرا در گروه‌های اجتماعی مورد توجه قرار گیرد. این مدل از روش تئوری بازی حمایت می‌کند و یک برنامه تخلیه محاسباتی پویا را برای فرایند تخلیه در سیستم محاسبات مه ارائه می‌دهد.

در [۲۲]، یک طرح تخلیه وظیفه برای شبکه‌های SDN پیشنهاد شده که در آن دستگاه‌های IoT با نقاط دسترسی چندگانه IoT به گره‌های محاسباتی مه متصل می‌شوند (نقاط دسترسی و گره‌های مه ثابت در نظر گرفته شده‌اند). این تحقیق همچنین مشکل تخلیه وظیفه در یک شبکه SDN را بررسی می‌کند که در آن دستگاه‌های VoIP به گره‌های محاسباتی مه از طریق APs[1] متصل می‌شوند. از آنجا که مسئله غیرخطی تخلیه بار با چالش‌هایی مواجه می‌شود، از یک تکنیک خطی برای ارائه یک فرمول خطی عددی صحیح[2] برای حل مسئله استفاده شده است. راه‌حل حریصانه، تأخیر، مصرف انرژی، مسیرهای چندرشته‌ای و شرایط شبکه پویا از قبیل کاربرد لینک و قدرت قانون‌پذیری SDN را در نظر می‌گیرد. در این تحقیق، تخلیه بار با استفاده از کوتاه‌ترین مسیر به‌سوی نود مه بر اساس تعداد پرش‌ها (hop) انجام می‌شود تا تأخیر متوسط را در هنگام رعایت محدودیت‌های انرژی به حداقل برساند.

برای فراهم‌کردن مدیریت توزیعی و مقیاس‌پذیری، گره‌های مه از تکنیک‌های شبکه‌سازی SDN استفاده می‌کنند که در آن صفحه کنترل قادر به تصمیم‌گیری است، در حالی که صفحه اطلاعات به‌سادگی وظایف ارسال و پردازش را ارائه می‌دهد [۲۳]. در این مطالعه برای حل مسائل مقیاس‌پذیری، یادگیری تقویتی با یک شبکه عصبی به نام شبکه عصبی عمیق[3] ترکیب می‌شود که سرعت یادگیری و عملکرد را بهبود می‌بخشد. در این مقاله، یک الگوریتم تخلیه کار همگن مشترک و تخصیص منابع طراحی شده که هدف آن به حداکثر رساندن وظایف پردازش تکمیل‌شده با حداقل نرخ سرریز از الگوریتم پیشنهادی است. نتایج ادغام زمانی مشاهدات از یک شبکه تکراری به گره‌ها اجازه می‌دهد که در انتخاب‌های خود هماهنگ باشند؛ بدون آنکه نیاز به دانستن حالت‌های صریح و مجموعه‌های عملیاتی دیگران داشته باشند که الگوریتم پیشنهادی مبتنی بر DRQN[4] را نسبتاً مقاوم در برابر محیط دینامیک نسبتاً قابل مشاهده می‌سازد. این نتایج همچنین نشان می‌دهند که منابع توزیع هوشمندانه با نیاز به محدودیت‌های مختلف تأخیر، تأثیر قابل توجهی بر عملکرد کلی سیستم می‌گذارند.

در [۲۴] یک رویکرد یادگیری متحد مبتنی بر بلاک‌چین پیشنهاد گردیده که در آن به‌روزرسانی‌های محلی از دستگاه‌های مختلف جمع‌آوری می‌شوند و به‌صورت امن در بلاک‌چین ذخیره می‌گردند. این روش به گونه‌ای طراحی شده است که ضمن حفظ حریم خصوصی داده‌ها، امکان یادگیری مدل‌های مشترک را برای چندین کاربر فراهم می‌آورد. در این رویکرد، هر دستگاه به‌صورت محلی داده‌های خود را پردازش کرده و مدل‌های به‌روزرسانی‌شده را به شبکه ارسال می‌کند. این به‌روزرسانی‌ها سپس در بلاک‌چین ثبت می‌شوند تا از صحت و یکپارچگی آنها اطمینان حاصل شود.

از سوی دیگر، [۲۵] و [۲۶] یک سیستم سلسله‌مراتبی یادگیری مشارکتی مبتنی بر معماری client-edge را پیشنهاد کردند که هدف آن کاهش هزینه‌های ارتباطات نسبت به روش‌های سنتی FL است. در این سیستم به‌جای اینکه همه داده‌ها به یک سرور مرکزی منتقل شوند، فرایند یادگیری به دو سطح تقسیم می‌شود: سطح کلاینت و سطح لبه. در این رویکرد، دستگاه‌های کلاینت به‌طور محلی مدل‌های یادگیری خود را آموزش می‌دهند و تنها به‌روزرسانی‌های مدل را به نودهای لبه ارسال می‌کنند. این نودهای لبه که معمولاً شامل سرورهای محلی یا گره‌های نزدیک به کاربر هستند، به‌عنوان واسطه‌ای عمل می‌کنند که به جمع‌آوری و تجمیع به‌روزرسانی‌های مدل از چندین کلاینت می‌پردازند. سپس نودهای لبه این به‌روزرسانی‌ها را به‌صورت دوره‌ای به سرور مرکزی ارسال می‌کنند.

۳- مدل مسئله

ما در این تحقیق، یک طرح تخلیه بار دینامیک در شبکه‌های SDN ارائه دادیم که در آن دستگاه‌های IoT به گره‌های محاسباتی مه از طریق سرویس‌های ابری مختلف متصل می‌شوند. به‌طور خاص در این معماری جنبه‌های زیر را در نظر می‌گیریم:

۱) تصمیم‌گیری محاسباتی: تصمیم‌گیری می‌تواند به‌صورت محلی در دستگاه‌های IoT یا از طریق سرورهای دور انجام شود.

۲) انتخاب مسیر بهینه: انتخاب مسیر بهینه برای انتقال داده‌ها و بار محاسباتی بین گره‌ها

۳) استفاده از الگوریتم یادگیری مشارکتی: به‌کارگیری الگوریتم FL برای بهبود فرایند یادگیری و پردازش داده‌ها

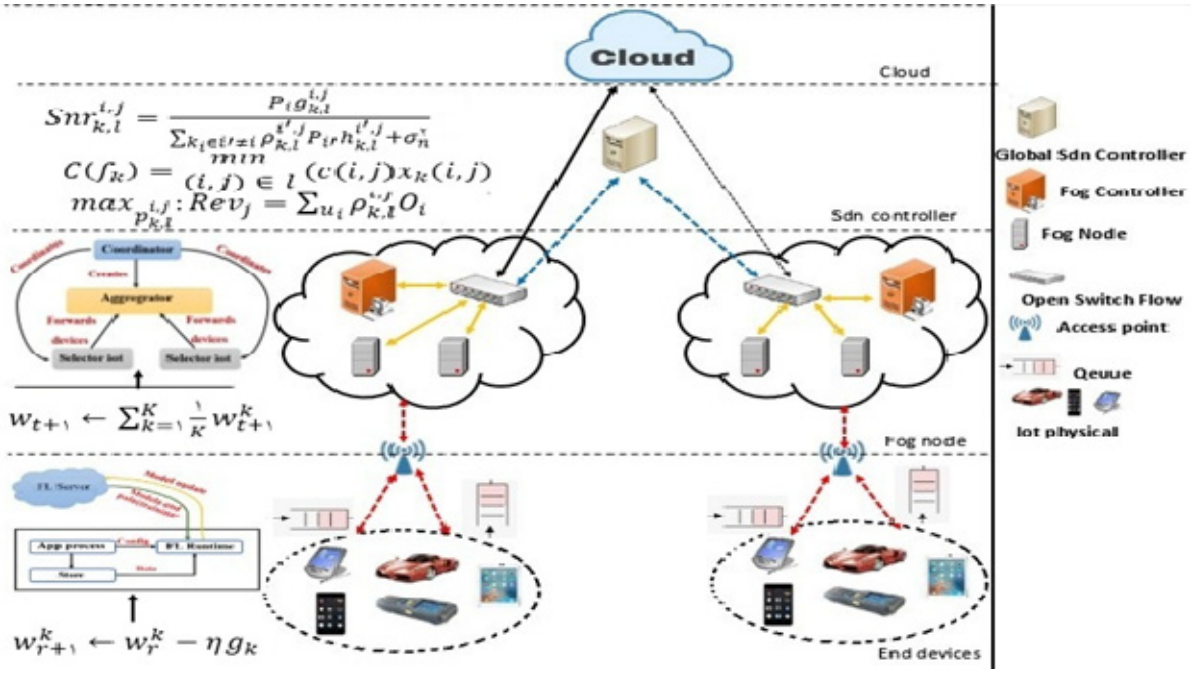
۴) تخلیه بار بهینه: انتخاب بهترین روش برای تخلیه بار محاسباتی به گره‌های مه

به‌طور خلاصه، معماری این کار به شرح زیر است:

در مسئله خود، همان طور که در شکل ۱ مشاهده می‌شود، از یک معماری شبکه‌ای پنج‌لایه استفاده کرده‌ایم که در آن فرایند تخلیه بار به‌طور مؤثر مدیریت می‌شود و به‌طور مختصر در ذیل تشریح شده است:

لایه IoT: این لایه شامل نقاط دسترسی و تجهیزات متنوعی است از قبیل حسگرها، موبایل‌های هوشمند، تبلت‌ها و دیگر دستگاه‌ها که از نظر توانمندی‌های فضای ذخیره‌سازی، پردازش و روابط ارتباطی غیرمتجانس می‌باشند. دسترسی از این لایه به سرورهای مه، معمولاً از طریق خطوط

1. Access Points
2. Integer Linear Program
3. Deep Q-Network
4. Deep Recurrent Q-Network



شکل ۱: چارچوب پیشنهادی تخلیه وظایف مبتنی بر یادگیری مشارکتی در محیط رایانش مه.

ارتباطی بی‌سیم امکان‌پذیر است. این لایه اولین نقطه تصمیم‌گیری برای ارسال وظایف به سرورهای نزدیک یا دور است.

لایه Edge: در این لایه، دستگاه‌ها از فناوری‌های دسترسی رادیویی مانند WiFi برای برقراری ارتباط با یکدیگر استفاده می‌کنند. گره‌های لبه در یک منطقه خاص با استفاده از یک رابط کانال بی‌سیم با نودهای مه در همان محدوده ارتباط برقرار می‌کنند.

لایه مه: در این لایه، نودهای مه قرار دارند که به‌طور مستقیم با دستگاه‌های انتهایی مانند گوشی‌های هوشمند، دوربین‌های بی‌سیم و دستگاه‌های حسگر از طریق یک رابط بی‌سیم تعامل دارند. هر نود مه یک یا چند سرویس محاسباتی و یک عامل مه را اجرا می‌کند که سعی می‌کنند تا حد امکان درخواست‌های محاسباتی و ذخیره‌سازی نودهای IoT را برآورده نمایند. برای مدیریت توزیع‌شده در سطح وسیع جغرافیایی، نودهای مه به کلونی‌هایی تقسیم می‌شوند. هر کلونی وظایف خاصی دارد و دارای یک نود قوی‌تر با نام Fog Orchestration Control Node است که وضعیت کلی کلونی از جمله آزادبودن ظرفیت پردازشی نودهای سطح کلونی جاری را در خود نگهداری می‌کند. اگر درخواست دریافتی توسط یکی از نودهای پیشگام در لایه IoT امکان اجرا در آن لایه را نداشته باشد، درخواست به Fog Orchestration Control Node منتقل خواهد شد و در صف M/M/1/k قرار خواهد گرفت. از طریق Fog Orchestration Control Node می‌توانیم مجموعه‌ای از اطلاعات مربوط به هر نود را جمع‌آوری کنیم که به‌طور خاص شامل خدمات نصب‌شده، مشخصات سخت‌افزاری و منابع محاسباتی کنونی (مانند CPU، RAM و فضای ذخیره‌سازی) است. این اطلاعات توسط سرویس تخلیه بار برای انتخاب یک تخلیه بهینه استفاده می‌شود.

لایه کنترل SDN: کنترلر SDN در این لایه قرار می‌گیرد که یک رابط برنامه‌نویسی را برای مدیریت شبکه فراهم می‌کند. تصمیم‌گیری در مورد نحوه تخلیه بار توسط این کنترلر اتخاذ می‌شود.

لایه ابر: در این لایه، مراکز داده‌ای قوی با منابع نامحدود رایانشی قرار داده شده‌اند تا خدمات مناسب را ارائه دهند.

از آنجا که این تحقیق قصد دارد بر ویژگی‌های لایه SDN و لایه مه تمرکز داشته باشد، به ذکر جزئیات لایه ابر پرداخته نمی‌شود. این معماری به ما این امکان را خواهد داد که به‌طور مؤثری بار محاسباتی را در شبکه‌های SDN مدیریت کنیم و بهینه‌سازی‌های لازم را برای بهبود عملکرد و کارایی دستگاه‌های IoT انجام دادیم.

تعریف متغیرها در جدول ۱ ارائه شده است. کاربران شامل مجموعه‌ای از دستگاه‌ها هستند که مثلاً شامل گوشی‌های هوشمند، دوربین‌های نظارتی، وسایل الکتریکی و ... می‌باشند و با نماد $U = \{u_1, u_2, \ldots, u_m\}$ نمایش داده می‌شوند. این کاربران ممکن است که بار محاسباتی یا ذخیره‌سازی خاصی را به ارائه‌دهندگان خدمات ابری محول کنند که با نماد $CN = \{cn_1, cn_2, \ldots, cn_n\}$ نشان داده می‌شوند.

این ارائه‌دهندگان خدمات می‌توانند به کاربران مختلفی با ملزومات محاسباتی متفاوت از نظر اندازه داده و تأخیر خدمات، پاسخ دهند. برای کاربرانی که به تأخیر حساس نیستند، محاسبات به ابر ارسال می‌شود؛ اما در مورد کاربران حساس به تأخیر، ارائه‌دهندگان خدمات یکی از گره‌های مه نزدیک را برای انجام محاسبات اختصاص می‌دهند. گره‌های مه که به کاربر نزدیک‌تر هستند، با تأخیر کمتری مواجه خواهند شد؛ اما موقعیت جغرافیایی تنها عامل مؤثر بر تأخیر خدمات نیست.

ابتدا تأخیر ارسال/ دریافت و تأخیر پردازشی محاسبه می‌گردد. هر کاربر اطلاعاتی را با خود حمل می‌کند و به آنها این امکان داده می‌شود که نیازهای خود را بیان کنند؛ مانند نیاز به تأخیر، اندازه داده و زمان پردازش (که در آن زمان پردازش و اندازه داده رابطه خطی دارند). این اطلاعات به ارائه‌دهندگان ابری ارسال می‌شود. در مرحله بعد، ابر یک گره مه مناسب با نیاز کاربر را مشخص می‌کند تا منابع رادیویی محاسباتی را به شکل مناسب تخصیص دهد. این فرایند تخصیص منابع به ما این امکان را می‌دهد که با بهینه‌سازی استفاده از منابع موجود، کارایی سیستم‌های محاسباتی ابری و مه را افزایش و به نیازهای متنوع کاربران پاسخ دهیم.

در واقع، تأخیر خدمات شامل سه دوره زمانی است: ۱) زمان ارسال داده، ۲) زمان پردازش توسط CPU و ۳) زمان دریافت نتایج. دوره‌های زمانی ارسال و دریافت به‌عنوان زمانی که برای ارسال داده به گره مه برای پردازش و برای دریافت نتایج پردازش نیاز است تعریف می‌شوند. از طرف دیگر، زمان پردازش CPU تحت تأثیر نرخ پردازش CPU هر گره مه قرار

جدول ۱: متغیرها.

متغیر	تعریف
CN_i	سرویس‌دهنده ابری
FN^j	گره‌های مه
K	کاربران
W^i	پهنای باند
C^i	منابع محاسباتی
$snr_{\min}$	نسبت سیگنال به نویز تداخل
$snr_{k,l}^{i,j}$	مقدار سیگنال به نویز دریافت‌شده u_i در fn_l^i
p_i	قدرت ارسال
$g_{k,l}^{i,j}$	پاداش کانال کاربر u_i و گره مه fn_l^i با استفاده از کانال Bw_k^j
$\rho_{k,l}^{i,j}$	متغیر دودویی تخصیص منابع
RP^i	مجموعه جفت منابع پهنای باند و چرخه CPU
$h_{k,l}^{i,j}$	پاداش کانال تداخل سایر کاربران سیار
σ_π^γ	نویز کانال
$r_{k,l}^{i,j}$	نرخ ارسال
$t_{k,l}^{i,j}$	تأخیر خدمت
t_t	مقدار زمان ارسال
t_p	مقدار زمان پردازش
t_r	مقدار زمان دریافت
D_i	اندازه داده
DC_i	تعداد چرخه CPU
$c_{k,l}^{i,j}$	نرخ پردازش
δ_t	متغیر تصادفی بین ۰ و ۱
O_i	قیمت پیشنهادی
α	پارامتر با واحد مگابیت بر ثانیه
T_i	نیاز به تأخیر
Rev_j	درآمد کل برای هر ارائه خدمت
q_R	ظرفیت کانال
q_C	ظرفیت CPU
q_{CN}	حداکثر تعداد کاربرانی که ارائه‌دهنده می‌تواند خدمات‌رسانی کند.

دارد؛ بنابراین برای هر CN_j، هنگامی که گره مه مناسب از مجموعه $FN^j=\{fn_1^j,fn_2^j,\ldots,fn_l^j\}$ برای هر کاربر انتخاب می‌شود، منابع مشترک $Bw^j=\{bw_1^j,bw_2^j,\ldots,bw_l^j\}$ و منابع محاسباتی $C^j=\{c_1^j,c_2^j,\ldots,c_l^j\}$ به طور مناسب تخصیص داده می‌شود.

کنترل‌کننده SDN از طریق پروتکل OpenFlow با سوئیچ‌ها ارتباط برقرار می‌کند و ارتباط بین کنترلر و لایه برنامه از طریق رابط برنامه‌نویسی کاربردی شمالی[1] SDN حاصل می‌شود.

برای شبکه مورد نظر، معیارهای QoS مربوط به هر لینک $(i,j)\in l$ شامل تأخیر $d(i,j)$، احتمال از دست دادن بسته $l(i,j)$ و پهنای باند در دسترس $c(i,j)$ است.

ترکیب قوانین برای احتمال از دست دادن بسته پیچیده‌تر است و به همین دلیل، لگاریتم احتمال موفقیت (که خود چند برابر است) در نظر گرفته می‌شود. در نتیجه، مقدار $l'(i,j)$ طبق (۱) محاسبه می‌شود

$$l'(i,j)=\log(1-l(i,j)) \qquad (1)$$

بنابراین برای هر مسیر $p=\{i,j,k,\ldots,s,t\}$ هست. همچنین (۲) تا (۵) مقادیر تأخیر، پهنای باند و احتمال از دست دادن بسته را محاسبه می‌کنند

$$D_p=d(i,j)+d(j,k)+\ldots+d(s,t) \qquad (2)$$

$$C_p=\min\{c(i,j),d(j,k),\ldots,d(s,t)\} \qquad (3)$$

$$L_p=l'(i,j)+l'(j,k),\ldots,l'(s,t) \qquad (4)$$

$$L_p=1-\exp(L_p') \qquad (5)$$

همچنین متغیر $x_k(i,j)$ یک تابع دوحالته مانند (۶) است

$$x_k(i,j)=\begin{cases}1\\0\end{cases} \qquad (6)$$

زمانی مقدار (۱) تا (۶) برابر یک می‌شود که فرایند مسیریابی بین گره‌های i و j انجام شده باشد و در غیر این صورت مقدار آن صفر خواهد بود. همچنین جهت محاسبه تأخیر و بسته‌های ازدست‌رفته از (۷) و (۸) استفاده می‌شود

$$D(f_k)=\sum_{(i,j)\in l}d(i,j)x_k(i,j) \qquad (7)$$

$$L(f_k)=\sum_{(i,j)\in l}l'(i,j)x_k(i,j) \qquad (8)$$

از قانون ترکیب پهنای باند، ظرفیت یک مسیر که یک جریان از آن عبور می‌کند، به صورت ریاضی بر اساس (۹) تعریف می‌شود

$$C(f_k)=\min_{(i,j)}\in l(c(i,j)x_k(i,j)) \qquad (9)$$

نهایتاً جهت محاسبه پهنای باند از (۱۰) استفاده می‌شود

$$C_{res}(i,j)=c(i,j)-\sum_{f_k\in f}q_k^{bw}x_{k(i,j)} \qquad (10)$$

این روابط به ما کمک می‌کند تا تأخیر خدمات و عملکرد شبکه را به‌طور دقیق‌تری تحلیل کنیم و بهینه‌سازی‌های لازم را برای بهبود کیفیت خدمات در شبکه‌های SDN انجام دهیم.

به‌منظور تحویل داده صحیح و کامل، نسبت سیگنال به نویز تداخل باید بالاتر از مقدار حدی $snr_{\min}$ باشد. مقدار $SINR$ از رابطه زیر تعریف می‌شود

$$snr_{k,l}^{i,j}=\frac{P_i g_{k,l}^{i,j}}{\sum_{k_i\in i'\neq i}\rho_{k,l}^{i',j}P_i h_{k,l}^{i',j}+\sigma_\pi^\gamma} \qquad (11)$$

عداد تراکنش‌هایی که توسط کاربر انجام شده و توسط گره مه تأیید شده و میانگین پاداش دریافتی برای تأیید هر تراکنش $(g_{k,l}^{i,j})$ به عنوان سود کانال و نرخ ارسال از k_i تا fn_l^j با استفاده از کانال w_k^j با شرط اجرای ملزومات $SINR$ در (۱۲) بیان می‌شود

$$r_{k,l}^{i,j}=w_k^j\log(1+snr_{k,l}^{i,j}) \qquad (12)$$

همچنین تأخیر ارائه خدمت u_i وقتی که جفت منابع (w_k^j,c_l^j) استفاده می‌شود، در (۱۳) تعریف خواهد شد. هرچه تأخیر ارائه خدمت کمتر باشد، رضایت کاربر بیشتر خواهد بود

$$t_{k,l}^{i,j}=t_t+t_p+t_r=\frac{D_i}{r_{k,l}^{i,j}}+\frac{DC_i}{c_{k,l}^{i,j}}+\delta t \qquad (13)$$

1. API Northbound

ورودی: مدل اولیه (w)

خروجی: به‌روزرسانی مدل اولیه (w_{t+1})

۱) شروع حلقه: تا زمانی که کلاینت‌ها در دسترس هستند، حلقه ادامه می‌یابد.

۲) انتخاب کلاینت‌ها: در هر دور، تعدادی از کلاینت‌ها (k) برای یادگیری مشارکتی انتخاب می‌شوند.

۳) دریافت مدل: هر کلاینت مدل w را از سرور دریافت می‌کند.

۴) محاسبه و میانگین‌گیری گرادیان: کلاینت‌ها گرادیان کاهشی را محاسبه کرده و میانگین آن را بر اساس (۲۴) به دست می‌آورند.

۵) به‌روزرسانی محلی مدل: هر کلاینت مدل خود را به‌روزرسانی کرده و این به‌روزرسانی را بر اساس (۲۶) انجام می‌دهد.

۶) ارسال به‌روزرسانی به سرور: کلاینت‌ها مدل به‌روزرسانی‌شده را به سرور ارسال می‌کنند.

۷) پایان حلقه: حلقه به پایان می‌رسد و در صورت نیاز، فرایند از ابتدا تکرار می‌شود.

شکل ۲: الگوریتم ۱

در بحث سود ارائه‌دهندگان خدمات، بدون اینکه در کلیت قضیه خللی وارد شود می‌توان یک رابطه خطی بین سود و اندازه داده در نظر گرفت؛ بنابراین سود پیشنهادی هر کاربر را می‌توان در (۱۴) تعریف نمود

$$O_i = f(D_i, T_i) \quad (۱۴)$$

و در (۱۵) تابع زیر برای تعریف فوق در نظر گرفته می‌شود

$$O_i = \alpha \frac{D_i}{T_i} \quad (۱۵)$$

درآمد کل برای هر ارائه‌دهنده خدمت نیز در (۱۶) حاصل خواهد شد

$$Rev_j = \sum_{k_i \in u} \rho_{k,l}^{i,j} O_i \quad (۱۶)$$

این روابط به ما کمک می‌کند تا تأثیرات مختلف بر روی عملکرد و سودآوری ارائه‌دهندگان خدمات را تحلیل کنیم و بهینه‌سازی‌های لازم را برای افزایش کارایی و رضایت کاربران انجام دهیم؛ بنابراین مسئله بهینه‌سازی و محدودیت‌های آن به‌صورت زیر است

$$\max_{p_{k,l}^{i,j}} : Rev_j = \sum_{u_i} \rho_{k,l}^{i,j} O_i \quad (۱۷)$$

$$\text{st.}: \rho_{k,l}^{i,j} t_{k,l}^{i,j} \le T_i \ , \ \forall k_i \in k, rp_{k,l}^{j} \in RP^j, cn_j \in CN \quad (۱۸)$$

$$\rho_{k,l}^{i,j} snr_{k,l}^{i,j} \ge snr_{\min} \ , \ \forall k_i \in K, rp_{k,l}^{j} \in RP^j, cn_j \in CN \quad (۱۹)$$

$$\sum_{k \in u, fn_l^j \in FN} \rho_{k,l}^{i,j} \le q_r \ , \ \forall w_j^i \in W^j, cn_j \in CN \quad (۲۰)$$

$$\sum_{k_i \in u, w_l^j \in W^j} \rho_{k,l}^{i,j} \le q_{c'} \ , \ \forall fn_l^i \in FN^j, cn_j \in CN \quad (۲۱)$$

$$\sum_{k_i \in k, rp_{k,l}^{i,j}} \rho_{k,l}^{i,j} \le q_{cn} \ , \ \forall cn_j \in CN \quad (۲۲)$$

$$\rho_{k,l}^{i,j} \in \{0,1\} \quad (۲۳)$$

در (۱۷)، هدف سیستم به حداکثر رساندن درآمد کل برای هر ارائه‌دهنده خدمت مشخص شده است. رابطه (۱۸) نیاز تأخیر برای هر کاربر را نشان می‌دهد. همچنین (۱۹) حداقل نیاز به نسبت سیگنال به نویز تداخل را برای هر کاربر تعریف می‌کند. روابط (۲۰) تا (۲۲) به ترتیب محدودیت ظرفیت هر کانال، گره مه و ارائه‌دهندگان خدمات را برآورده می‌سازند.

٤- حل مسئله

با توجه به اینکه هر کاربر اطلاعاتی را با خود حمل می‌کند، به آنها این امکان داده می‌شود که نیازهای خود را بیان کنند؛ از جمله نیاز به تأخیر، اندازه داده و زمان پردازش. لازم به ذکر است که اندازه داده و زمان پردازش رابطه‌ای خطی دارند. این اطلاعات به ارائه‌دهندگان خدمات ابری ارسال می‌شود. در مرحله بعد، ابر یک گره مه مناسب با نیاز کاربر را شناسایی می‌کند تا منابع پردازشی و محاسباتی را به‌طور بهینه تخصیص دهد. مثلاً اگر کاربر نیاز به پردازش داده‌های حسگرهای محیطی داشته باشد، گره مهی که نزدیک‌ترین به محل نصب حسگرها است، انتخاب می‌شود تا تأخیر در پردازش کاهش یابد.

هر دور از یادگیری مشارکتی شامل شش مرحله است:

۱) انتخاب

۲) پیکربندی

۳) به‌روزرسانی محلی

۴) تجمیع محلی

۵) تجمیع سراسری

۶) گزارش‌دهی

پس از انتخاب بهترین گره مه، یک مدل پایه عمومی در گره مه مرکزی ذخیره می‌شود و کپی‌های این مدل با دستگاه‌های کاربران به اشتراک گذاشته می‌شود. سپس مدل‌ها بر اساس داده‌های محلی که تولید می‌شود، آموزش می‌بینند. در مرحله بعد، پارامترهای به‌روزرسانی‌شده از مدل‌های آموزش‌دیده محلی به اشتراک گذاشته می‌شوند. این پارامترها با مدل پایه عمومی واقع در سرور مرکزی ترکیب می‌شوند. هنگامی که مدل مرکزی مجدداً با پارامترهای جدید آموزش داده شد، برای تکرار بعدی دوباره با دستگاه‌های کاربران به اشتراک گذاشته می‌شود. با هر چرخه، مدل‌ها حجم متنوعی از اطلاعات را جمع‌آوری می‌کنند و بدون ایجاد نقض حریم خصوصی، بهبود می‌یابند.

گره‌های مه با استفاده از یک رابط کانال بی‌سیم با کلاینت‌ها ارتباط برقرار می‌کنند، جایی که تعداد کلاینت‌ها بزرگ‌تر از تعداد گره‌های مه است $(f < k)$.

حل مسئله از اجرای دو الگوریتم تشکیل شده و با استفاده از زبان پایتون پیاده‌سازی می‌شود.

در شکل ۲:

پس از به دست آوردن پارامترهای مدل اولیه، کلاینت‌ها فرایند تقسیم داده‌های محلی خود را آغاز می‌کنند. این داده‌ها به چندین دسته با اندازه ثابت تقسیم می‌شوند و سپس هر کلاینت با استفاده از روش گرادیان کاهشی تصادفی[1] به‌طور هم‌زمان به‌روزرسانی‌های لازم را انجام می‌دهد. طبق (۲۴) میانگین گرادیان (g_k) محاسبه می‌شود

$$w_{t+1} = w_t + \alpha \frac{\partial L}{\partial W_t} \quad (۲۴)$$

که t گام زمانی، w وزن یا پارامتر مورد نظر برای به‌روزرسانی و α نرخ یادگیری است که ثابت نگه داشته نمی‌شود و به‌طور دینامیک با توجه به اندازه گرادیان‌ها تنظیم می‌گردد. همچنین $\partial L/\partial W$ گرادیان تابع زیان L است که باید نسبت به w به حداقل برسد.

در هر دسته، خروجی از لایه‌های پایین (کلاینت‌ها) به همراه برچسب‌های مربوط به داده‌ها به سرور مه ارسال می‌شود تا تجمیع داده‌ها

1. Stochastic Gradient Descent

ورودی: پارامترهای آموزش‌دیده کلاینت‌ها
خروجی: مدل آموزش نهایی
۱) شروع حلقه: تا زمانی که fog (محیط محاسباتی توزیع‌شده) وجود دارد، حلقه ادامه می‌یابد.
۲) به‌روزرسانی مدل کلاینت‌ها: هر کلاینت مدل خود را به‌روزرسانی می‌کند:
$w_{t+1}^k \leftarrow w_t$
۳) پایان حلقه: در صورت عدم وجود fog، حلقه به پایان می‌رسد.
۴) محاسبه مدل سراسری پس از آموزش: مدل سراسری به‌روزرسانی می‌شود:
$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{1}{K} w_{t+1}^k$
۵) ارسال مدل نهایی به کلاینت‌ها: مدل آموزش‌دیده نهایی به کلاینت‌ها ارسال می‌شود: w_{t+1}
۶) پایان: فرایند به پایان می‌رسد و مدل نهایی برای استفاده در دسترس کلاینت‌ها قرار می‌گیرد.

شکل ۳: الگوریتم ۲.

انجام گیرد. برای به‌روزرسانی‌های محلی مشترک، ضروری است که دستگاه‌های همراه با سرور مه ارتباط برقرار کنند. در این راستا، مقدار و اندازه دسته در زمان‌های ارتباطی (t_{batch}) طبق (۲۵) تأثیر قابل توجهی بر فرایند آموزش یادگیری بر روی داده‌های محلی دارد. این ارتباطات به بهینه‌سازی فرایند یادگیری کمک کرده و تضمین می‌کند که مدل به‌طور مؤثر از اطلاعات موجود در تمامی کلاینت‌ها بهره‌برداری کند

$$t_{batch} = \frac{D_k}{V_{batch}} \quad (۲۵)$$

در اینجا D_k نمایانگر مقدار داده‌های محلی موجود در کلاینت‌ها و V_{batch} اندازه دسته داده‌ها است. هنگامی که t به‌طور قابل توجهی بزرگ باشد، این به معنای وجود ارتباطات فراوان بین کلاینت‌ها و سرور است. در چنین شرایطی، فرایند به‌روزرسانی‌های محلی ممکن است به دلیل عدم ثبات شبکه تحت تأثیر قرار گیرد، حتی اگر پهنای باند بالا و تأخیر پایینی وجود داشته باشد. از سوی دیگر اگر V_{batch} کوچک باشد، تعداد زمان مورد نیاز برای ارتباطات محلی افزایش می‌یابد؛ در حالی که یک V_{batch} بزرگ ممکن است بر همگرایی آموزش مدل تأثیر منفی بگذارد.

پس از محاسبه گرادیان کاهش تصادفی، نوبت به به‌روزرسانی مدل و ارسال آن به سمت سرور می‌رسد (معادله (۲۶)). این مرحله از اهمیت بالایی برخوردار است؛ زیرا به‌روزرسانی مدل تأثیر مستقیمی بر عملکرد و دقت نهایی سیستم یادگیری ماشین دارد

$$w_{t+1}^k \leftarrow w_t^k - \eta g_k \quad (۲۶)$$

این الگوریتم به‌صورت سیستماتیک و مرحله‌ای، یادگیری مشارکتی بین کلاینت‌ها و سرور را انجام می‌دهد و با به‌روزرسانی‌های محلی به بهبود مدل کمک می‌کند. پس از توضیح الگوریتم در شکل ۲ به بررسی الگوریتم در شکل ۳ می‌پردازیم که فرایند هماهنگی را مدیریت می‌کند.

در شکل ۳:

سرور هماهنگ‌کننده پس از دریافت مدل‌های آموزش‌دیده از کلاینت‌ها باید ارزیابی کند که آیا فرایند یادگیری مشارکتی باید ادامه یابد یا خیر. این ارزیابی با استفاده از مجموعه داده‌های آموزش‌دیده در سرور مرکزی و انجام تست‌های مربوطه صورت می‌گیرد. اگر نتایج حاصل از این تست‌ها مطلوب باشد، نیازی به انجام تکرارهای بعدی که منجر به محاسبات بیشتر می‌شود، نخواهد بود. در این صورت فرایند به‌روزرسانی مدل به پایان می‌رسد و مدل نهایی می‌تواند به کار گرفته شود. اما اگر نتایج مطلوب نباشد، سرور هماهنگ‌کننده می‌تواند کل فرایند را مجدداً تکرار کند. در این حالت، پارامترهای مدل برای دورهای بعدی توزیع‌شده و کلاینت‌ها مجدداً به آموزش مدل می‌پردازند. این چرخه تکراری به‌تدریج مدل را بهبود می‌بخشد و نتایج بهتری به همراه دارد. مثلاً با تکرار این فرایند، ممکن است دقت مدل از ۸۰٪ به ۹۰٪ افزایش یابد.

این الگوریتم به‌طور سیستماتیک به هماهنگی و به‌روزرسانی مدل‌های آموزش‌دیده در محیط fog می‌پردازد و با استفاده از میانگین‌گیری، مدل نهایی را تولید می‌کند که می‌تواند به کلاینت‌ها ارسال شود. به‌عنوان مثال در هر دور از الگوریتم، کلاینت‌ها می‌توانند به‌طور هم‌زمان داده‌های خود را به‌روزرسانی و نتایج را به سرور ارسال کنند.

۵- ارزیابی

در این بخش از مقاله به بررسی و ارزیابی شبیه‌سازی شبکه و سخت‌افزار مورد استفاده در آزمایش پرداخته می‌شود. هدف از این ارزیابی، تحلیل عملکرد الگوریتم پیشنهادی در مقایسه با سایر روش‌ها و درک بهتر از تأثیرات متغیرهای مختلف بر نتایج است.

جهت اجرای آزمایش با شرایط یکسان، ابتدا محیط شبیه‌سازی شبکه و سخت‌افزار مورد استفاده در آزمایش معرفی می‌شود. برای شبیه‌سازی و پیاده‌سازی الگوریتم پیشنهادی از زبان‌های برنامه‌نویسی پایتون، متلب و مینی‌نت استفاده شده است. این انتخاب زبان‌ها به دلیل قابلیت‌های متنوع آنها در شبیه‌سازی و تحلیل داده‌ها صورت گرفته است. بخش‌های اصلی شبیه‌سازی به شرح زیر است:

- طراحی معماری شبکه
- تعریف متغیرهای تحقیق
- انجام فرایند ارتباط متغیرهای تحقیق با معماری شبکه

مشخصات سخت‌افزار و شرایط تعریف‌شده برای شبیه‌سازی محیط آزمایش به شرح زیر است:

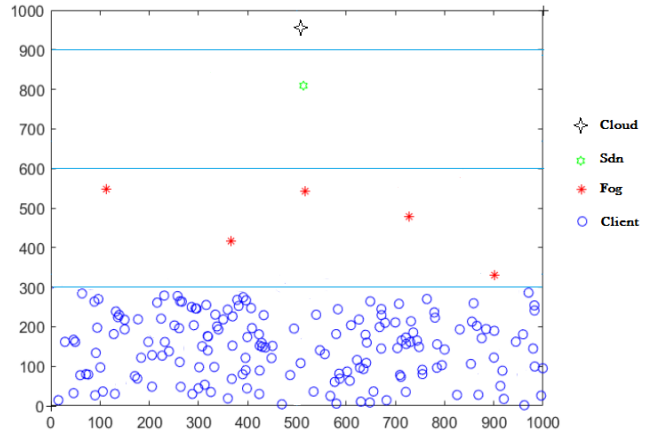
- پردازنده: سیستم اجراکننده شبیه‌سازی از یک پردازنده Core i۷ با فرکانس حداکثر ۲/۵۰ مگاهرتز استفاده می‌کند.
- حافظه: ۸ گیگابایت RAM
- سیستم عامل: لینوکس

در این آزمایش، یک شبکه با $CN = ۱$ به‌عنوان ارائه‌دهنده خدمات در نظر گرفته شده که شامل $FN = ۵$ گره‌های مه است. این گره‌ها به‌طور تصادفی در شبکه توزیع شده‌اند و از دیتاست CIRFAR−۱۰ با ابعاد ۳۲×۳۲ استفاده شده است. این دیتاست شامل ۱۰۰K داده آموزشی و ۲۰K داده تست می‌باشد که این داده‌ها شامل تصاویر رنگی از قبیل اتومبیل، سگ و کشتی هستند.

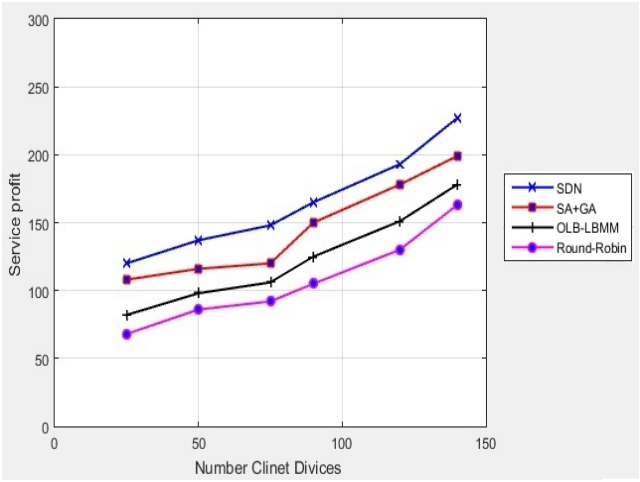
شکل ۴، توزیع شبکه چهارلایه‌ای را نشان می‌دهد که به چهار قسمت مساوی تقسیم شده است: ارائه‌دهندگان خدمات در لایه بالا، شبکه SDN در لایه بعدی و گره‌های مه و کاربران در لایه پایین. فرض می‌شود که تعداد کاربران $K = ۱۰۰$ است که به‌طور تصادفی در شبکه توزیع شده‌اند.

عملکرد الگوریتم پیشنهادی SDN-FL در مقایسه با سه الگوریتم دیگر (SA+GA، OLB-LBMM و Round-Robin) بررسی می‌شود. نتایج نشان می‌دهند که SDN-FL در ارائه سود به ارائه‌دهندگان خدمات به‌طور قابل توجهی بهتر عمل می‌کند؛ به‌ویژه با افزایش تعداد دستگاه‌های IoT. این موضوع در شکل ۵ به‌وضوح قابل مشاهده است.

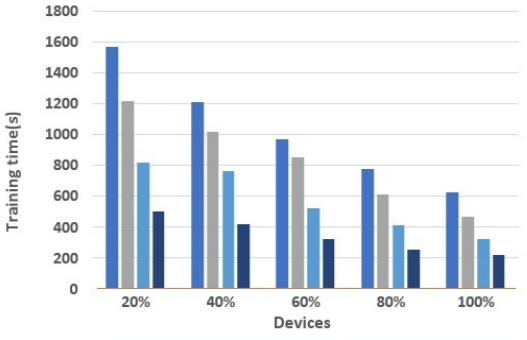
به‌منظور ارزیابی عملکرد روش پیشنهادی ارائه‌شده در این مقاله، نتایج با کارهای مرتبط اخیر از مقالات [۲۷] تا [۲۹] مقایسه می‌شود. در ادامه به معرفی پارامترهای ارزیابی می‌پردازیم.



شکل ۴: شبکه چهارلایه‌ای.



شکل ۵: سود خدمات.

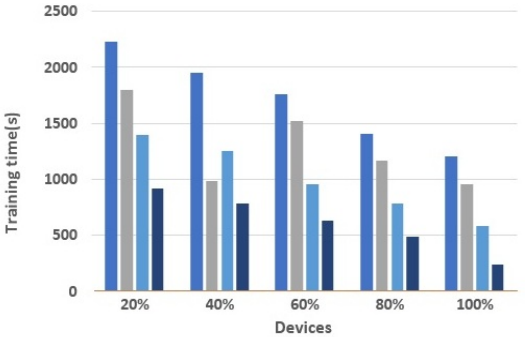


	20%	40%	60%	80%	100%
DRL	1571	1211	971	780	625
EdgeFed	1221	1020	851	610	470
FogFL	821	765	520	416	320
SDN-FL	501	418	320	253	221

شکل ۶: آموزش ۲۰٪ دیتاست.

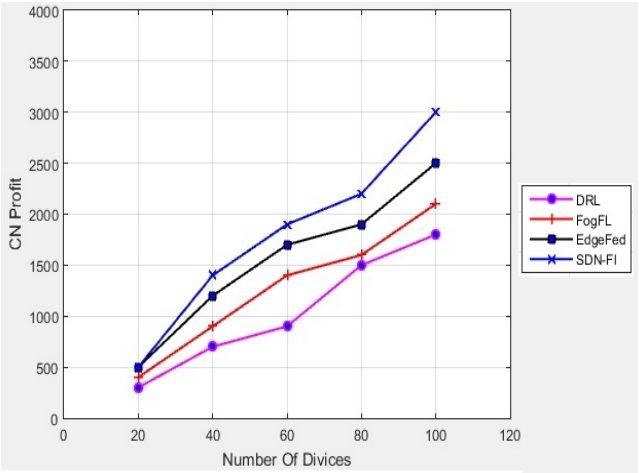
۱) تأثیر تحرک کلاینت در زمان آموزش: تحرک کلاینت تأثیر مهمی بر زمان آموزش و دقت مدل دارد. نتایج نشان می‌دهند با افزایش تحرک دستگاه‌ها، زمان آموزش کاهش می‌یابد. در شکل‌های ۶ و ۷، تأثیر تحرک بر زمان آموزش با ۲۰٪ و ۵۰٪ از دیتاست بررسی شده است. نتایج به‌دست‌آمده نشان می‌دهند SDN-FL عملکرد بهتری نسبت به سایر الگوریتم‌ها دارد و زمان آموزش را به‌طور قابل توجهی کاهش می‌دهد.

شکل ۶ اثرات تحرک‌پذیری دستگاه بر زمان آموزش را در زمانی که ۲۰٪ از مجموعه داده‌ها برای آموزش روی یک دستگاه واحد استفاده شده



	20%	40%	60%	80%	100%
DRL	2230	1955	1756	1402	1201
EdgeFed	1800	982	1520	1170	958
FogFL	1400	1258	956	780	580
SDN-FL	921	782	635	488	239

شکل ۷: آموزش ۵۰٪ دیتاست.



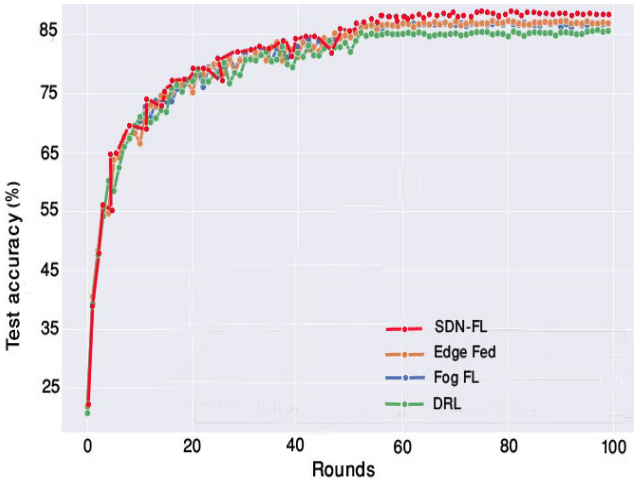
شکل ۸: سود ارائه‌دهندگان.

است، نشان می‌دهد. حرکت دستگاه در زمان‌های ۴۰٪، ۶۰٪، ۸۰٪ و ۱۰۰٪ آموزش‌ها کامل شده است. شکل ۶ نشان می‌دهد که SDN نسبت به الگوریتم‌های دیگر عملکرد بهتری دارد. هنگامی که به سمت بازه زمانی ۵۰٪ حرکت می‌کنیم، زمان آموزش تا ۳۷٪ در هر دور کاهش می‌یابد. با این حال، زمانی که با مجموعه داده‌ها حرکت می‌کنیم، در زمان ۸۰٪ که آموزش در حال تکمیل‌شدن است، زمان آموزش تا ۴۶٪ در هر دور کاهش می‌یابد.

شکل ۷ نشان می‌دهد هنگامی که ۵۰٪ از مجموعه داده‌ها برای آموزش روی یک دستگاه واحد مورد استفاده قرار می‌گیرد، همچنین حرکت دستگاه در زمان‌های ۲۰٪، ۴۰٪، ۶۰٪، ۸۰٪ و ۱۰۰٪ از آموزش تکمیل می‌شود. در شکل ۷ مشاهده می‌شود که زمان آموزش روی دستگاه‌ها بیشتر از دستگاه‌های شکل ۶ است و مشخص است که استفاده از ۵۰٪ مجموعه داده‌ها برای آموزش دستگاه‌های تلفن همراه، پایدارتر و قابل مقایسه‌تر است.

۲) تأثیر تحرک در دقت سراسری: در این بخش، دقت مدل سراسری در حین حرکت دستگاه‌ها بررسی شده است. نتایج نشان می‌دهند SDN-FL توانسته که دقت را در حین تحرک حفظ کند و این موضوع در شکل ۸ و ۹ به‌خوبی نمایان است. این نکته حائز اهمیت است که SDN-FL با حفظ دقت، زمان آموزش را نیز می‌کاهد.

شکل ۸ دقت تست را به‌عنوان تابعی از دوره آموزشی نشان می‌دهد. ما این آزمایش را برای مجموعاً ۱۰۰ دور انجام دادیم که یک دستگاه تلفن



شکل ۹: آنتروپی دقت آموزش.

همراه ۲۰٪ و ۵۰٪ از مجموعه داده‌ها را در اختیار دارد. دستگاه در دوره‌های مختلف در سراسر ۱۰۰ دور، شامل ۱۰۰، ۸۰، ۶۰، ۴۰ و ۲۰ قرار می‌گیرد.

در شکل ۹ به‌وضوح نشان داده شده که این الگوریتم‌ها همگی دقت را حفظ می‌کنند و بدین دلیل است که پارامترهای مدل از سرور مرکزی دریافت شده و آموزش در کارگزار لبه مقصد انجام می‌شود. عملکرد SDN-FL تحت تأثیر عواملی مانند تعداد داده‌های متعادل و نامتعادل، فرکانس دریافتی با حرکت دستگاه‌ها، انتخاب بهترین گره مه و مراحل آموزش مدل قرار دارد. این عوامل به‌طور مستقیم بر زمان آموزش و دقت مدل تأثیر می‌گذارند.

عملکرد SDN-FL تحت تأثیر عواملی مانند تعداد داده‌های متعادل و نامتعادل، فرکانس دریافتی با حرکت دستگاه‌ها، انتخاب بهترین گره مه و مراحل آموزش مدل قرار دارد. این عوامل به‌طور مستقیم بر زمان آموزش و دقت مدل تأثیر می‌گذارند.

نتایج حاصل از آزمایش‌ها نشان می‌دهند روش پیشنهادی SDN-FL در مقایسه با سایر روش‌ها عملکرد بهتری دارد. به‌ویژه، زمان آموزش را برای مجموعه داده‌های متعادل و نامتعادل به‌طور قابل توجهی کاهش می‌دهد و هیچ تلفات دقتی ندارد. همچنین سربار ناشی از انتقال داده بین سرورهای حاشیه‌ای در مقایسه با زمان آموزش ناچیز است. این نتایج می‌توانند به بهبود عملکرد سیستم‌های مبتنی بر IoT و یادگیری فدرال کمک کنند و نشان‌دهنده پتانسیل بالای SDN-FL در این زمینه هستند.

٦- نتیجه

با توجه به ناهمگن‌بودن محیط‌های اینترنت اشیا، هجوم داده‌ها در شبکه‌های سنتی می‌تواند به عملکرد نامناسب شبکه و کاهش کیفیت خدمات منجر شود. شبکه‌های نرم‌افزارمحور به دلیل قابلیت بررسی مستقل هر جریان و تعیین قوانین متناسب برای آن، توجه محققان را به خود جلب کرده و این امکان را فراهم می‌آورند که از ویژگی‌های این نوع شبکه‌ها در محیط‌های ناهمگن اینترنت اشیا بهره‌برداری شود.

در این مقاله با استفاده از دید جامع SDN نسبت به کل شبکه، به بررسی توازن بار پرداخته شده است. در شبکه‌های کامپیوتری سنتی، توزیع بار به‌طور ناهمگون و غیریکنواخت، همراه با استفاده از روش‌های تصادفی در جریان‌های ورودی، منجر به ایجاد سیستمی می‌شود که فاقد توازن بار است. این عدم توازن بار می‌تواند به کاهش کیفیت خدمات (QoS)، از بین رفتن تعدادی از جریان‌های ورودی، افزایش سربار، اشغال ظرفیت شبکه، ناپایداری و در نهایت کاهش کارایی شبکه منجر شود؛ بنابراین این مقاله به توزیع عادلانه ترافیک در بخش‌های مختلف شبکه و فرایند تخصیص کارآمد منابع موجود با استفاده از شبکه‌های SDN پرداخته است. همچنین سیستم یادگیری مشارکتی با دو مسئله عمده محدود می‌شود: زمان آموزش و دقت. این مسائل زمانی چالش‌برانگیزتر می‌شوند که یک دستگاه در حین آموزش FL در حال حرکت باشد. در این راستا، مقاله حاضر الگوریتم SDN-FL را معرفی کرده که برای اولین بار به چالش تحرک دستگاه‌ها در حین آموزش FL، به‌خصوص در محیط‌های مبتنی بر لبه می‌پردازد.

نتایج به‌دست‌آمده نشان می‌دهند که زمان آموزش نسبت به مقالات مورد بررسی بهبود یافته است. مثلاً نتایج نشان می‌دهند که زمان آموزش با استفاده از SDN-FL تا ۳۰٪ نسبت به الگوریتم‌های دیگر کاهش یافته است. این بهبود در زمان آموزش ناشی از این است که الگوریتم SDN-FL به جستجوی راه‌حل‌های بهتری برای مسائل می‌پردازد، در حالی که در میان تعداد زیادی از راه‌حل‌های ممکن قرار دارد. برخلاف سایر تکنیک‌های بهینه‌سازی، SDN-FL نیازی به تنظیم پارامترهای الگوریتمی ندارد که این امر موجب ساده‌ترشدن اجرای آن می‌شود.

به‌طور کلی، نتایج این تحقیق نشان‌دهنده پتانسیل بالای SDN-FL در بهبود عملکرد سیستم‌های یادگیری فدرال در محیط‌های ناهمگن اینترنت اشیا است و می‌تواند به‌عنوان یک راهکار مؤثر برای مدیریت بهینه ترافیک و منابع در این زمینه مورد استفاده قرار گیرد. این نتایج می‌توانند به طراحان سیستم‌های IoT کمک کنند تا با استفاده از SDN-FL، کارایی و کیفیت خدمات را در محیط‌های پیچیده و ناهمگن بهبود بخشند.

نهایتاً تحقیقات آینده می‌توانند به بررسی تأثیرات SDN-FL در شرایط مختلف شبکه و با استفاده از داده‌های واقعی بپردازند. همچنین پیشنهاد می‌شود که به بررسی بهبودهای ممکن در الگوریتم و آزمایش‌ها در محیط‌های واقعی توجه شود تا پتانسیل کامل SDN-FL در مدیریت ترافیک و منابع به دست آید.

مراجع

[1] M. Chiang and T. Zhang, "Fog and IoT: an over view of research opportunities," *IEEE Internet of Things J.*, vol. 3, no. 6, pp. 854-864, Dec. 2016.

[2] J. Wen, *et al.*, "A survey on federated learning: challenges and opportunities," *International Journal of Machine Learning and Cybernetics*, vol. 14, pp. 513-535, 2023.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication," *IEEE Communications Surveys Tutorials*, vol. 9, no. 4, pp. 2322-2358, Fourthquarter 2017.

[4] P. Macha and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628-1656, Thirdquarter 2017.

[5] A, Kumari and P. K. Jana, "Communication efficient federated learning with data offloading in fog-based IoT environment," *Future Generation Computer Systems*, vol. 158, pp. 158-166, Sept. 2024.

[6] J. Singh, P, Singh, M. Hedabou, and N. Kumar, "An efficient machine learning-based resource allocation scheme for SDN-enabled fog computing environment," *IEEE Trans. on on Vehicular Technology,* vol. 72, no. 6, pp. 8004-8017, Jun. 2023.

[7] A. Zaman, A. Jarray, and A. Karmouch, " Software-defined network-based edge cloud resource allocation framework," *IEEE Access*, vol. 7, 10672–10690, 2019.

[8] R. W. Cottle and T. N. Mukund, *Linear and Nonlinear Optimization*, 2nd Editon, vol. 253, New York, NY, USA: Springer, 2017.

[9] P. Muts, I. Nowak, and E. M. T. Hendrix, "The decomposition-based outer approximation algorithm for convex mixed-integer nonlinear programming," *J. of Global Optimization*, vol. 77, no. 1, pp. 75-96, May 2020.

[10] P. P. Liang, *et al.*, *Thinklocally, Actglobally: Federated Learning with Local and Global Representations*, 2020, arXiv: 2001.01523.
[11] H. H. Zhuo, W. Feng, Y. Lin, Q. Xu, and Q. Yang, *Federated Deeprein-Forcement Learning*, Feb. 2020, arXiv: 1901.08277.
[12] H. Yu, *et al.*, "A fairness-aware incentive scheme for federated learning," in *Proc. AAAI/ACM Conf. AI, Ethics, and Society*, pp. 393-399, New York, NY, USA, 7-9 Feb. 2020.
[13] S. Truex, *et al.*, "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM Workshop on Artificial Intelligence and Security*, pp. 1-11, London, UK, 15-15 Nov. 2019.
[14] A. A. Süzen and M. A. Simsek, "A novel approach to machine learning application to protection privacy data in health care: federated learning," *Namlk Kemal Tip Dergisi*, vol. 8, no. 1, pp. 22-30, 2020.
[15] S. Lin, G. Yang, and J. Zhang, Real-Time Edge Intelligence Inthemaking: A Collaborative Learning Framework via Federated Meta-Learning, 2020, arXiv: 2001.03229.
[16] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "Acrowd sourcing framework for on-device federated learning," *IEEE Trans. on Wireless Communications*, vol. 19, no. 5, pp. 3241-3256, May 2020.
[17] M. G. R. Alam, Y. K. Tun, and C. S. Hong, "Multi-agent and reinforcement learning based code offloading in mobile fog," in *Proc. Int. Conf. on Information Networking,* Kota Kinabalu, Malaysia, pp. 334-338, 13-15 Jan. 2016.
[18] T. N. Dang and C. S. Hong, "A distributed ADMM approach for data offloading in fog computing," in *Proc. of the Int. Conf. on Advanced Technologies for Communications*, Hanoi, Vietnam, pp. 286-291, Atlanta, GA, USA, 12-14 Oct. 2016.
[19] C. Fricker, F. Guillemin, P. Robert, and G. Thompson, "Analysis of an offloading scheme for data centers in the framework of fog computing," *ACM Trans. on Modeling and Performance Evaluation of Computing Systems*, vol. 1, no. 4, pp. 1-18, Jul. 2016.
[20] Q. Zhu, B. Si, F. Yang, and Y. Ma, "Task offloading decision in fog computing system," *China Communications*, vol. 14, no. 11, pp. 59-68, Nov. 2017.
[21] L. Liu, Z. Chang, and X. Guo, "Socially-aware dynamic computation offloading scheme for fog computing system with energy harvesting devices," *IEEE Internet of Things J.*, vol. 5, no. 3, pp. 1869-1879, Jun. 2018.
[22] L. Phan, D. Nguyen, M. Lee, D. Park, and T. Kim, "Dynamic fog-to-fog offloading in SDN-based fog computing systems," *Future Generation Computer Systems*, vol. 117, pp. 486-497, Apr. 2020.
[23] J. Baek and G. Kaddoum, "Heterogeneous task offoading and resource allocations via deep recurrent reinforcement learning in partial observable multi-fog networks," *IEEE Internet of Things J.*, vol. 8, no. 2, pp. 1041-1056, Jan. 2017.
[24] H. Kim, J. Park, M. Bennis, and S. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279-1283, Jun. 2019.
[25] L. Liu, J. Zhang, S. Song, and K. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. of IEEE Int. Conf. on Communications*, 6 pp., Dublin, Ireland, 7-11 Jun. 2020.
[26] M. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* pp. 8866-8870, Virtual Barcelona, 4-8 May 2020.
[27] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported internet of things," *IEEE Internet of Things J.,* vol. 7, no. 8, pp. 6914-6921, Aug. 2020.
[28] Y. Yunfan, L. Shen, L. Fang, T. Yonghao, and H. Wanting, "EdgeFed: optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209191-209198, 2020.
[29] R. Saha, S. Misra, and P. K. Deb, "FogFL: fog assisted federated learning for resource-constrained IoT devices," *IEEE Internet of Things J.*, vol. 8, no. 10, pp. 8456-8463, 15 May 2020.

محمدرضا شرفی هویدا تحصیلات خود را در مقطع کارشناسی در رشته مهندسی کامپیوتر (نرم‌افزار) در سال ۱۳۸۵ از دانشگاه آزاد اسلامی واحد تویسرکان و در مقطع کارشناسی ارشد در رشته مهندسی مکاترونیک در سال ۱۳۹۰ از دانشگاه آزاد اسلامی واحد شهر مجلسی به پایان رسانده است. وی از سال ۱۳۹۵ به عنوان دانشجوی دکتری در دانشگاه آزاد اسلامی واحد میبد مشغول به تحصیل می‌باشد. زمینه‌های تحقیقاتی مورد علاقه او عبارتند از: شبکه‌های کامپیوتری، هوش مصنوعی و یادگیری ماشین، اینترنت اشیاء و رایانش مه و فناوری آموزشی.

محمدرضا ملاحسینی اردکانی مدرک دکترای خود را در سال ۱۳۹۷ در رشته کامپیوتر- نرم‌افزار از دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران دریافت کرد. ایشان استادیار گروه مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد میبد می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: پایگاه داده‌ها، رایانش ابری، رایانش مه، رایانش لبه و معماری سرویس‌گرا.

وحید آیت‌اللهی تفتی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد رشته مهندسی کامپیوتر (نرم افزار) به‌ترتیب در سال‌های ۱۳۷۵ و ۱۳۸۵ از دانشگاه صنعتی اصفهان و دانشگاه علوم و تحقیقات تهران و دکتری علوم کامپیوتر (شبکه) در سال ۱۳۹۴ از دانشگاه یوتی‌ام مالزی به پایان رسانده است. از سال ۱۳۸۵ هیات علمی دانشگاه آزاد اسلامی واحد تفت شده و اکنون استادیار دانشکده مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد یزد می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه های کامپیوتری و شبکه های نرم افزار محور، اینترنت اشیاء و مجازی سازی.