دوره ۱۳، شماره پاییز و زمستان ۱۳۹۹

مجله فناوری اطلاعات در طراحی مهندسی
Information Technology in Engineering Design
http://ited.sinaweb.net

زمانبندی وظایف در اینترنت اشیا با استفاده از بازنمایی تنسوری

شبنم شادرو *(۱) ریحانه معارف دوست (۲)

(۱) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران*

(۲) دانشکده مهارت و کارآفرینی، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

(تاریخ ارسال : ۱۳۹۹/۱۱/۱۰ تاریخ پذیرش : ۱٤۰۰/۰٦/۲۹)

چکیده

به‌کارگیری رایانش ابری در کنار رایانش مه در کاربردهای اینترنت اشیا می‌تواند، مشکلات موجود در رایانش ابری مانند عدم امنیت کافی، عدم آگاهی از مکان و تاخیر در اجرای وظایف را بهبود بخشید. در این مقاله جهت تصمیم گیری برروی مکان اجرای وظایف مستقل ار یکدیگر، وظایف برروی ساختار تنسور با مرتبه‌ی بالا بازنمایی می‌شوند و سپس با استفاده از SOM، خوشه‌بندی صورت می‌گیرد. وظایف در دو خوشه ابر و مه قرار می‌گیرند. برای حفظ محرمانگی وظایف، وظایف محرمانه موجود در خوشه‌ی ابری را به مه با روش آستانه گذاری انتقال می‌دهیم. وظایف با توجه به بستر اجرایی خود، با استفاده از زمانبندی EDF زمانبندی می‌شوند. روش پیشنهادی، با روش‌های خوشه‌بندی SOM و SOM سلسله مراتبی و روش DPTO و AE-SOM مقایسه شده است. نتایج نشان می‌دهد که بازنمایی تنسور توانسته در زمان‌بندی وظایف نیز تاثیر داشته و ازلحاظ زمان اتمام وظایف و هزینه و goodput و همچنین تعداد وظایف اجراشده در مهلت مقرر بهبود حاصل کند. روش پیشنهادی، در زمانیکه بارکاری سیستم بالا است، از لحاظ تعداد وظایف اجرا نشده در مهلت مقرر، نسبت به میانگین روش‌های دیگر، ۵۳ درصد و از لحاظ زمان اتمام وظایف، ۱٤/۰۸ درصد و از لحاظ هزینه ۳٤/٦ درصد بهبود داشته است.

کلمات کلیدی: اینترنت اشیا، تنسور، رایانش ابر، رایانش مه، زمانبندی وظایف

* **عهده‌دار مکاتبات:**

شبنم شادرو

نشانی: گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

پست الکترونیکی: shabnam.shadroo@srbiau.ac.ir **تلفن : ۰۹۱۵۵۲۱۳٤۵۹**

۱- مقدمه

امروزه توسعه‌ی چشمگیر بکارگیری اینترنت اشیا و مواجهه با حجم کلان داده‌ها و نیازهای پردازشی آنها و نقاط ضعف و محدودیت‌های دستگاه‌های اینترنت اشیا، لزوم بکارگیری روش‌های جدید محاسباتی باقدرت بیشتر را بیش از پیش عیان نموده است. در ابتدا رایانش ابری جهت رفع مشکلات مطرح شد ولی رایانش ابری هم دچار نقطه‌ضعف‌هایی از جمله حفظ امنیت و محرمانگی، برطرف نکردن نیاز برنامه‌های کاربردی حساس به تأخیر و همچنین عدم آگاهی از مکان و هزینه‌های ارتباطی بالا و پشتیبانی نکردن از حرکت پذیری می‌باشد [۱, ۲, ۳, ٤].سپس برای رفع مشکلات ذکرشده رایانش مه روی کار آمدند. در بیشتر معماری‌ها، مه در کنار ابر قرار گرفت تا مشکلات ذکر شده را بهبود دهند، اما استفاده از ابر و مه درکنار هم به تنهایی کافی نیست و حجم زیاد داده‌ی ارسالی به آن‌ها نیاز به کنترل و مدیریت دارد. در مقاله شادرو و همکارش [٥] بیان شده است که می‌توان چالش‌ها و روش‌های ارئه شده در اینترنت اشیا و کلان داده را در چهاربخش شامل ذخیره‌سازی، انتقال داده‌ها، پردازش و تحلیل داده‌ها، قرار داد. چالش‌های موجود در بخش ذخیره‌سازی، علاوه بر حجم داده و میزان سرعت تولید آن از سمت دستگاه‌های اینترنت اشیا را می‌توان به نیاز به محل مناسب ذخیره‌سازی داده‌ها و تشخیص اطلاعات مفید از داده‌ها و ذخیره‌سازی آن اطلاعات، ناهمگن بودن داده‌های ارسالی و امنیت، ناهمگنی داده و حساس به زمان بودن داده، هزینه‌های مالی آن بررسی نمود. چالش‌های مربوط به انتقال داده را می‌توان به پهنای باند کافی، تاخیر در ارسال داده‌ها، هزینه‌های مالی مربوط به پهنای باند تقسیم‌بندی کرد. چالش‌های موجود در پردازش و تحلیل داده‌های ارسالی را می‌توان به نیاز به محیط و توان پردازشی مناسب و زمانبندی درست وظایف ارسالی و به‌کارگیری ابزارهای تحلیل مناسب و استفاده از ظرفیت‌های منابع موجود بطور موثر و هزینه‌های مالی آن اشاره نمود. اینترنت اشیا در بخش‌های مختلف از جمله حوزه سلامت، خدمات رفاهی، ماشین‌های هوشمند، خرید و فروش و غیره، داده‌های بسیاری تولید می‌کند که حفظ امنیت و حریم آن‌ها ضروری است. طبق بررسی‌های صورت گرفته، تنها ۲۲ درصد از کاربران اعلام کرده‌اند که در این حوزه هیچ نگرانی در مورد حریم خصوصی خود ندارند. گرچه اینترنت اشیا، کیفیت زندگی مردم و روال کاری سازمان‌ها را بهبود می‌بخشد، اما بستری آسیب‌پذیر در برابر حمله احتمالی هکرهاست. مطالعات انجام‌شده در سال ۲۰۱٤ نشان داد که حدود ۷۰ درصد از دستگاه‌های اینترنت اشیا موجود در دنیا در معرض آسیب‌های امنیتی هستند. برخی از دستگاه‌های اینترنت اشیا به علت عدم رمزگذاری انتقال، رابط وب ناامن و عدم حفاظت کافی از نرم‌افزار، آسیب‌پذیرند [۷] [٦]. برخی از برنامه‌های اینترنت اشیا از زیرساخت‌های حساس و خدمات استراتژیک مانند شبکه‌های هوشمند و حفاظت از تأسیسات پشتیبانی می‌کنند و تعدادی از برنامه‌های اینترنت اشیا مقدار زیادی از اطلاعات شخصی در مورد خانواده، سلامت و وضعیت مالی فرد را جمع‌آوری می‌کنند. رعایت نشدن نکات امنیتی و بی‌توجهی به حریم خصوصی سبب ایجاد مقاومت در پذیرش اینترنت اشیا توسط افراد و سازمان‌ها خواهد بود.

همانطور که بیان شده، می‌توان برای ذخیره‌سازی و پردازش و تحلیل داده‌ها در اینترنت اشیا از رایانش ابری و یا رایانش مه و گاهی هردو و در برخی مواقع نیز برای ذخیره‌سازی موقت و تحلیل‌های مختصر از دستگاه‌های اینترنت اشیا نیز استفاده کرد. در بیشتر کاربردها، دستگاه‌های اینترنت اشیا وظایف خود را به مه یا ابر ارسال می‌کنند. وظایف ایجادشده در اینترنت اشیا وابسته به پروژه مطرح‌شده متفاوت می‌باشد. برخی از این وظایف با توجه به اهمیت آن‌ها اجباراً با داده مورد نیاز همراه می‌باشد. همراه بودن داده با وظیفه می‌تواند سرعت انجام آن وظیفه را افزایش دهد ولی محدودیت‌هایی دارد. اگر بخواهیم این وظیفه را برای تحلیل و اجرای وظیفه به ابر ارسال کنیم، به پهنای باند بالا و زمان تأخیر بیشتر و در نتیجه هزینه زیادتر نیازخواهیم داشت. همچنین این چالش زمانی پررنگ تر می‌شود که وظیفه دارای کلان داده بوده و امنیت آن نیز از اهمیت بالایی برخوردار است. استفاده از ابر و مه در اینترنت اشیا می‌تواند به ذخیره‌سازی حجم زیادی از داده‌ها و پردازش و تحلیل آن‌ها کمک کرده و همچنین روش‌هایی مانند کلاسترینگ، یادگیری ماشین، یادگیری عمیق نیز می‌تواند در تحلیل این حجم داده نقش بسزایی را ایفا

نمایند. ذخیره‌سازی و انتقال این حجم داده به مه یا ابر و پردازش و تحلیل آن‌ها خود نیاز به راه حل‌هایی برای بهبود زمان انتقال و پردازش و تحلیل با لحاظ کردن پارامترهایی مانند تعداد منابع مصرفی و پهنای باند مورد نظر و همچنین مباحثی چون بهبود زمان پاسخ، قابلیت اطمینان و حفظ حریم خصوصی و دسترس‌پذیری دارد. در بحث انتقال کلان داده می‌توان از روش‌هایی برای کاهش ابعاد داده و بازنمایی داده‌ها و فشردگی داده به همراه روش‌های یادگیری ماشین، یادگیری عمیق و روش‌های ابتکاری استفاده نمود.

وجود برنامه‌های کاربردی که نیاز به محاسبات زیادی دارند و همچنین محدودیت منابعی و چالش‌های مطرح شده، محققان را ملزم می‌کند که از روش‌هایی برای مدیریت وظایف قبل از ورود به محیط مه و ابر استفاده کنند. جهت مدیریت وظایف، وظایفی را که نیازهای پردازشی و منابعی مشابه‌ای دارند و یا ویژگی‌های مشترکی دارند را در یک خوشه قرار داده و هر خوشه را به ابر یا مه ارسال می‌کنند. سپس زمان‌بندی وظایف مناسب با محیط‌های رایانش ابری و مه اعمال می‌شود. زمانبندی وظایف در رایانش مه و ابر به تخصیص مناسب منابع به وظایف کاربردی گفته می‌شود. می‌توان زمان‌بندی را به دو مدل تبدیل نمود، زمان‌بندی وظایف مستقل و زمان‌بندی وظایف وابسته. در وظایف مستقل هیچ رابطه و ارتباطی بین وظایف وجود ندارد. در صورتی که در وظایف وابسته برخی از وظایف به یکدیگر وابسته بوده و ارتباط داده‌ای بین آن‌ها وجود دارد. اثبات شده است که زمان‌بندی وظایف در نودهای ابر و مه مساله ای NP-Complete می‌باشد[۸] . مساله‌ای که در این مقاله به آن می‌پردازیم، ارائه راه حل مناسب مدیریت خودکار وظایف قبل از ورود به مه و ابر با حفظ محرمانگی است. اهداف مورد نظر در این مقاله، پس از مدیریت و زمانبندی وظایف در مه و ابر، کاهش هزینه‌های استفاده از مه و ابر، افزایش تعداد اجرای وظایف در مهلت مقرر، اجرای وظایف با درجه‌ی محرمانگی بالا در محیط مه و همچین کاهش زمان کل اجرای وظایف می‌باشد. در این تحقیق فرض شده است که وظایف به همراه خود داده داشته و جهت پردازش نیاز به داده دارند، وظایف مستقل از یکدیگربوده، محیط مه مکان امنی برای پردازش داده‌های محرمانه است.

در این مقاله وظایف دریافتی از دستگاه‌های اینترنت اشیا با استفاده از بازنمایی تنسوری مدل شده و سپس وظایف توسط نگاشت خودسازمانده[۱] در دو خوشه (وظایف ارسالی به ابر – وظایف ارسالی به مه) قرار می‌گیرند. قبل زمانبندی خوشه‌ها، در صورتی که وظیفه‌ی دارای درجه‌ی محرمانگی بالا در خوشه‌ی ابر باشد، به خوشه‌ی مه منتقل می‌شود. هر دو خوشه متناسب با بستر اجرایی خود با استفاده از زمانبندی مبتنی بر مهلت زمانی، زمانبندی می‌شوند.

در بخش دوم ادبیات تحقیق و مروری بر روش‌های مدیریت و زمانبندی وظایف در اینترنت اشیا ارائه می‌شود. روش پیشنهادی در بخش سوم، بیان خواهد شد. در بخش چهارم به بررسی نتایج پرداخته‌ایم.

۱- مروری بر مقالات

در این بخش به مرور مقالات در سه حوزه (خوشه‌بندی وظایف– زمانبندی وظایف– بازنمایی وظایف) می‌پردازیم. هنگامی‌که در طراحی سیستم، همزمان از ابر و مه استفاده می‌شود، نیاز به مدیریت وظایف خواهیم داشت تا بتوانیم بهره‌وری سیستم را افزایش دهیم. جهت مدیریت وظایف می‌توان از دو دیدگاه آنها را مدیریت نموده و بستر اجرای آنها را به درستی انتخاب کرد. تحقیقات مختلفی جهت مدیریت وظایف صورت گرفته است. در برخی از مقالات ایده‌ی اصلی آن‌ها برروی خوشه‌بندی وظایف بوده و وظایف داخل خوشه‌ها را به بستر ابر یا مه ارسال می‌کنند. در تحقیقات دیگری جهت بهبود بهره‌وری بسترهای ابر یا مه، وظایف ارسال شده به ابر یا مه را با استفاده از روش‌ها ی مختلف

[۱] Self-Organizing Map

زمانبندی می‌نمایند. در الگوریتم‌هایی که خوشه‌بندی وظایف، انجام می‌دهند، وظایف را به خوشه‌هایی مطابق با قوانینی تخصیص می‌دهد. سپس وظایف داخل خوشه را به پردازنده‌ی مناسب اختصاص می‌دهد. این الگوریتم‌ها سربار ارتباطی را با نگاشت وظایف خوشه به پردازش‌گر یکسان کاهش می‌دهد. جهت نمایش ویژگی‌های وظایف، الگوریتم‌های مختلفی ارائه شده است، تنسور یکی از روش‌هایی است که در این مقاله نیز به کار گرفته شده است. بنابراین در بخش مرور برخی از مقالاتی که از تنسور جهت بازنمایی داده‌ها استفاده نموده‌اند می‌پردازیم.

۲-۱ مرور مقالات مربوط به زمانبندی وظایف در ابر یا مه

اوان و همکارانش [۹] در یک مقاله مروری، زمانبندی وظایف در محیط ابری را در دسته‌های زمانبندی براساس اولویت، کاهش زمان اتمام وظایف، بهبود انرژی و بهبود هزینه، QOS و بهبود Load Balancing،Consistency و Fairness قرار داده است. در این بخش، به بررسی برخی از مقالات در حوزه‌ی زمانبندی وظایف می‌پردازیم.

فانگ و همکارانش [۱۰] الگوریتم زمان‌بندی وظایف مبتنی بر توازن بار ارائه دادند. این الگوریتم در دو مرحله انجام می‌شود. در مرحله اول تخصیص ماشین مجازی برنامه‌های کاربردی صورت می‌گیرد و در مرحله دوم منابع مناسب را برای ماشین‌های مجازی فراهم می‌کند. فرضیه‌های این مقاله مستقل بودن وظایف و عدم وجود وظیفه تکراری می‌باشد و تنها زمان پاسخ و تقاضا برای منابع در این الگو در نظر گرفته شده است و پهنای باند و هزینه درنظر گرفته درنظر نگرفته است.

یین و همکارانش [۱۱] در سال ۲۰۱۸ مقاله‌ای منتشر کردند که در آن از مظروف‌ها (container) برای زمان‌بندی و تخصیص منابع استفاده می‌کنند. هدف آن‌ها غلبه بر معایب استفاده از ماشین مجازی بوده است. الگوریتم مطرح‌شده در این مقاله برای فعالیت‌های صنعتی بلادرنگ مورد استفاده قرار می‌گیرد. برای اینکه بتوانند زمان پاسخ بلادرنگی را برای وظایف فراهم کنند آن‌ها از مه استفاده نمودند. هر نود مه دارای سه سنجش درخواست، زمان‌بند وظیفه و مدیریت منابع می‌باشد. بخش زمان‌بندی وظایف مشخص می‌کند که وظایف درمه یا در ابر اجرا شوند. نتایج ارائه‌شده نشان می‌دهد که الگوریتم پیشنهادی تأخیر وظایف را کاهش داده و تعداد همزمانی وظایف در نود مه را افزایش می‌دهد.

تانگ و همکارانش [۱۳] [۱۲] مدل زمان‌بندی وظایف مبتنی بر الگوریتم ژنتیک ارائه دادند و هدف آن‌ها بهینه‌سازی مصرف انرژی بوده است و آن‌ها در مدل خود وابستگی وظایف، انتقال داده و مهلت زمان اجرای وظیفه و هزینه را لحاظ نمودند. در این تحقیق هر وظیفه می‌تواند هم در ابر متحرک و هم به صورت محلی در دستگاه متحرک آپلود شود و آن وظایفی که نیاز به تعامل با کاربر سیار دارد، بر روی دستگاه سیار اجرا می‌شود و وظیفه‌ای که نیاز به محاسبات و در نتیجه مصرف انرژی بیشتری دارد در ابر سیار ذخیره می‌شود. نتایج این زمانبندی نشان می‌دهد که در مصرف انرژی و زمان پاسخ بهینه عمل می‌کند.

هیانگ و همکارانش [۱٤] روش زمان‌بندی وظایف را برای تضمین دسترسی بهتر به شبکه ابر و سرعت پردازش رایانش ابری متحرک، با توجه به محدودیت‌های پهنای باند شبکه و هزینه استفاده از ابر، ارائه دادند. در واقع روش استفاده‌شده دارای دو بخش تعیین اولویت وظایف و انتخاب پردازشگر مناسب می‌باشد. با این حال جزئیات در مورد چگونگی به‌دست آوردن برخی معیارها مانند اولین زمان شروع اول، زمان پایان وظایف ارائه نشده و پیچیدگی الگوریتم ناشناخته است. در مقاله دیگری [۱۵] نیز زمانبندی وظایف در مه را براساس اولویت انجام داده اند. در این مدل، در صورتی که منابع لازم برای وظیفه‌ی مورد نظر در محیط مه فراهم باشد، وظیفه در مه اجرا می‌شود و در غیر این صورت به ابر ارسال

می‌شود. در مه وظایف براساس اولویت خود اجرا می‌شوند و در صورتی که نتواند وظیفه در زمان مقرر اجرا شود، نادیده گرفته می‌شود. روش ارائه شده زمان پاسخ و هزینه را نسبت به روش‌های مقایسه شده در مقاله، کاهش داده است.

چو و همکارانش [۱۶] به زمان‌بندی وظایف در رایانش تصویر مه پرداختند. آن‌ها با فرض به وابستگی وظایف و ناهمگونی دستگاه‌ها، در این مقاله مساله زمان‌بندی داده‌های‌تصویری در مه را به مساله زمان‌بندی خطی تبدیل نموده اند. الگوریتم جدیدی را ارائه نموده اند که مقیاس پذیر و feasible می‌باشد.

لیو و همکارانش [۸] زمانبندی وظایف در محیط مه (TSFC) را با استفاده از الگوریتم I-Apriori ارائه نمودند. ورودی TSFC، قوانین association استخراجی از I-Apriori می‌باشد، که ارتباط بین مجموعه نود مه و مجموعه وظایف را مشخص می‌کند. سپس آن‌هایی که اولویت بیشتری دارند، را جهت ورود به مه انتخاب می‌نماید. آن‌ها نشان دادند که این روش از نظر زمان اجرایی و میانگین زمان انتظار نسبت به روش‌های مشابه خود بهتر می‌باشد.

۲-۲ مرور مقالات مربوط به خوشه‌بندی وظایف در ابر یا مه

الگوریتم ترکیبی زمانبندی وظایف مبتنی بر خوشه‌بندی وظایف[۲] که توسط تیان و همکارانش بیان شده است [۱۷] . این الگوریتم ابتدا گراف وظایف DAG را خوشه‌بندی می‌کند و وظایفی را که سربار ارتباطی زیادی دارند، در یک خوشه قرار می‌دهد وسپس لایه بندی می‌کند. پس از محاسبه ی اولویت و پردازنده مناسب را به خوشه اختصاص می‌دهد. نویسندگان جهت ارزیابی روش پیشنهادی خود، دو پارامتر SLR(Schedule Length Ratio) و speed up را در نظر گرفتند و نشان دادند روش پیشنهادی نسبت به دو الگوریتم HEFT(Heterogeneous Earliest) Finish Time) [۱۸] و CPOP(Critical Path on a Processor) [۱۸]، کارایی بهتری دارد.

روشی ترکیبی با استفاده از فازی و گراف وزن‌دار دو قسمتی در [۱۹] ارائه شده است. نویسندگان این مقاله وظایف وابسته و مستقل را با استفاده از DAG در نظر گرفته‌اند و در مرحله ی اول از خوشه‌بندی فازی و گراف دو قسمتی استفاده کرده‌اند و سپس ماشین مجازی را به وظایف اختصاص می‌دهند. روش پیشنهادی در مقایسه با روش‌هایی که زمان اتمام وظایف، هزینه‌ی کل را در نظر گرفته‌اند، بهبود داشته است و تعادلی بین زمان اتمام وظیفه و هزینه برقرار می‌کند.

اولاه و همکاران [۲۰] وظایف را براساس منابع مورد نیازشان توسط k-MEANS خوشه‌بندی نمودند. سپس وظایف را مطابق با منابع دردسترس زمانبندی می‌نمایند. روش به‌کار گرفته شده در این مقاله، زمان بیکاری منابع را کاهش داده و بهره‌وری از منابع را افزایش داده است. همچنین با توازن بار در بین نودها، هزینه‌های را کاهش داده است. روش پیشنهادی نسبت به روش‌های [۲۱] FIFO و [۲۲] FIRE، میانگین زمان انتظار و مجموع زمان اجرای کمتری را دارد.

[۲] HTSTC (hybrid task scheduling algorithm based on task clustering)

وانگ و همکارانش [۲۳] پارامترهایی شامل معیارهای فضای حافظه، زمان مورد انتظار هر وظیفه و بهنای باند مورد انتظار هر وظیفه را جهت خوشه‌بندی در لایه مه در نظر گرفتند. در این روش در لایه مه علاوه بر خوشه‌بندی وظایف، مدلی جهت تجمیع منابع نیز در نظر گرفتند تا بتوانند محل اجرای وظیفه را با توجه به نیازش مشخص نمایند. در این مقاله به منظور زمانبندی وظایف به استفاده از خوشه‌بندی به بهبود

روش Firework [۲٤] که از الگوریتم‌های هوش جمعی می‌باشد، پرداختند. آنها نشان دادند که با به‌کارگیری الگوریتم پیشنهادی توانسته‌اند، زمان پردازش را کاهش داده و توازن بار را در دستگاه‌های مه برقرار سازند.

در مقاله [۲۵] ، زمانبندی مبتنی بر شناسایی الگو به صورت آنلاین برای هوشمندسازی کارخانه در محیط بلادرنگ ارائه شده است. نویسندگان از SOM برای شناسایی الگوی وظایف استفاده نمودند و با استفاده از اطلاعات پیشین وظایف SOM را آموزش داده اند. پس از تخصیص بهینه وظایف بهECE (Edge computing – enabled entity) مناسب، به صورت آنلاین زمانبندی صورت می‌گیرد.

موتوسامی و هکارانش [۲٦] چارچوبی را برای زمانبندی وظایف مبتنی بر خوشه بندی با استفاده از خوشه بند k-means ارائه دادند. آنها وظایف را براساس طول وظیفه خوشه بندی وVM ها را براساس ظرفیت VM گروه بندی می‌کنند. پس از خوشه بندی، هر وظیفه در داخل خوشه براساس VM اختصاص داده به آن در گروه VM ها زمانبندی می‌شود. روش پیشنهادی آنها روشی پویا برای توازن بار با هدف کاهش زمان اتمام وظایف و زمان اجرا بوده و نشان دادند که این روش به این هدف رسیده است.

جدول ۱ مقالات بیان شده در دو بخش زمانبندی و خوشه‌بندی، را براساس ایده، مزایا، معایب و نرم‌افزار استفاده شده و معیار ارزیابی مقایسه کرده‌ایم.

جدول ۱: مقایسه برخی از مقالات مربوط به زمان‌بندی و خوشه‌بندی وظایف

	مرجع	سال	ایده	مزایا	معایب	نرم‌افزار	معیار ارزیابی
زمانبندی وظایف	[۱۰]	۲۰۱۰	زمان‌بندی دو سطحی	-بکارگیری منابع بیشتر -بهبود توازن بار	در نظر نگرفتن پهنای باند و هزینه	Cloudsim	زمان اتمام وظایف
	[۱۱]	۲۰۱۸	استفاده از مظروف	-درنظرگرفتن حالت بلادرنگ - کاهش زمان پاسخ وظایف - افزایش تعداد همزمانی در وظایف در نود مه	-در نظر نگرفتن زمان اجرای وظایف و ترافیک شبکه -حذف زمان رایانش ابری - حذف معیار مناسب برای منابع ابری	Libvirt Docker QEMU	تعداد وظایف همزمان میانگین زمان
	[۱۲] [۱۳]	۲۰۱۸	استفاده از الگوریتم ژنتیک	لحاظ کردن موارد زیر -وابستگی‌های وظایف -تأخیر انتقال داده -مهلت زمان پاسخ	عدم مقایسه به روش‌های دیگر	C++	زمان اتمام وظایف حال رساندن مصرف انرژی
	[۱٤]	۲۰۱٤	استفاده از روش مبتنی بر آگاهی از رقابت	در بحث هزینه موثر خوب عمل کند کارایی را بهبود می بخشد	در برخی معیارها خوب عمل نمی‌کند.	جاوا	-هزیه -زمان پاسخ
	[۱٦]	۲۰۱۸	استفاده از برنامه ریزی خطی	رایانش تصویری مه تمرکز برروی دستگاه‌های ناهمگن مقیاس پذیر	لحاظ نکردن پیچیدگی محاسباتی	Matlab	زمان اجرا
	[۸]	۲۰۱۸	زمانبندی مبتنی بر روش-های داده کاوی	کاهش مجموع زمان اجرای وظایف کاهش میانگین زمان انتظار	فرض‌های مسئله خیلی ساده است	جاوا	زمان اجرا زمان انتظار
	[۱۵]	۲۰۱۸	زمانبندی مبتنی بر اولویت در رایانش مه	کاهش زمان پاسخ و هزینه	اولویت ها ثابت می‌باشند	cloudAnalyst	رمان پاسخ هزینه

خوشه بندی وظایف	[۱۷]	۲۰۲۰	زمانبندی وظایف مبتنی بر خوشه‌بندی	افزایش سرعت و کاهش کاهش تاثیر سربارارتباطی و زمان وظیفه		Matlab	SLR (Schedule Length Ratio) Speed up
	[۱۹]	۲۰۱۹	به‌کارگیری خوشه‌بندی فازی و گراف دو قسمتی وزن دار جهت تخصیص وظایف	استفاده از DAG ترکیبی تعادل بین هزینه و زمان کار	می‌توان بررسی را با تعداد بیشتری ماشین مجازی نیز انجام داد	CloudSim	زمان اتمام وظایف Cost
	[۲۰]	۲۰۲۰	به‌کارگیری خوشه‌بندی kMEANS براساس نیازمندی منابع با تئوری صف *M/M/c*	افزایش بهره وری منابع کاهش زمان انتظار کاهش مجموع زمان اجرا درهنگام افزایش وظایف وارد شده به سیستم	می‌توان با افزایش ملاک‌های خوشه-بندی، بهره وری را افزایش داده -تغییر سایز وظیفه ممکن است باعث تعییر در کارایی زمانبندی شود	CloudSim	Execution time Resource utilization رمان انتظار
	[۲۳]	۲۰۲۰	استفاده از مدل خوشه-بندی و تجمیع منابع و بهبود زمانبندی firework	کاهش زمان پردازش توازن بار درنودهای مه	زمان پردازش وظایف و مصارف انرژی در نظر گرفته نشده است.		زمان اجرا Load Ratio
	[۲۵]	۲۰۱۸	زمانبندی وظایف آنلاین مبتنی بر شناسایی الگو	کاهش پیچیدگی محاسباتی و کاهش تاخیر محاسبات و بررسی آنلاین			Task execution latency Execution error rate
	[۲۶]	۲۰۲۱	زمانبندی وظایف براساس خوشه بندی k-means	کاهش زمان اجرا و زمان اتمام وظایف	برای خوشه بندی فقط طول وظیفه را به عنوان ویژگی در نظر گرفته است	CloudSim	زمان اجرا زمان اتمام وظایف

۲-۳ مرور مقالات در حوزه بازنمایی وظیفه با تنسور

تنسور [۲۶] آرایه‌ای چندبعدی است که به صورت ماتریس چند بعدی نمایش داده می‌شود. در این تحقیق فرض می‌شود، که تمامی ورودی‌های تنسور دارای مقادیر حقیقی می‌باشند و تنسور مرتبه‌ی n ام را با $\underline{X} \in R^{I_1 \times \ldots \times I_N}$ تشکیل شد، از آرایه‌ای باورودی‌ها $x_{i_1 \ldots i_N}$ $(1 \le n \le N$ مرتبه تنسور و $(1 \le n \le N)$بعد مرتبه n ام می‌باشد. به طور مثال تنسور $\underline{X} \in R^{3 \times 2 \times 20}$ دارای ابعاد ۳، ۲،۲۰ می‌باشد. بردار تنسور مرتبه ۱ ام و ماتریس تنسور مرتبه ی ۲ بوده و تنسور با مرتبه ۳ و بالاتر به عنوان تنسور مرتبه بالا[۳] شناخته می‌شوند. روش‌های مختلفی از جمله Principal Component Analysis (PCA) [۲۷] و Incremental Singular Value Decomposition (SVD) [۲۸]و Dynamic Tensor Analysis (DTA) [۲۹] جهت بازنمایی و کاهش ابعاد داده‌ها ارائه شده است. تنسور، برای بازنمایی داده مناسب بوده و تنسورها می‌توانند ساختار درونی دیتا را نگه دارند. مزایای قابل توجهی در بازنمایی و نمایش داده‌ها با ابعاد بالا و پیچیده دارد. در بسیار از مقالات از روش‌های تحلیلی مبتنی بر تنسور مانند tensor decomposition [۳۰]، tensor network در کاربردهای کاربردی مورد استفاده قرار می‌گیرند.

پنگ لی و همکارانش[۳۱] برای حل مشکل تنوع کلان داده در اینترنت اشیا، مدل محاسباتی عمیق پیچیده[۴] با گسترش دادن شبکه عصبی کانولوشن از فضای برداری به فضای تنسوری پیشنهاد دادند. در این مقاله برای آشکار ساختن ارتباطات پنهان میان جنبه‌های مختلف کلان داده، مدلی مبتنی بر تنسور ارائه داده تا نمایانگر هر شی ناهمگن باشد. یک تنسور قادر به بازنمایی ساختار داده‌ای ناهمگن و نگهداری ساختارهای پردازش نشده داده است که به ما اجازه می‌دهد اشتراک و مکمل بودن بین جهت‌های مختلف را در نظر بگیریم. همچنین تنسور می‌تواند از وقوع برخی مشکلات در فضای بردار مانند فاجعه‌ی بعد و مفرد بودن جلوگیری کند و در بسیاری از برنامه‌های کاربردی مانند استخراج/انتخاب مشخصه، تلفیق داده و تشخیص الگو به کار گرفته شود. به منظور استفاده کامل از مشخصه‌های محلی و توپولوژی‌های موجود در کلان داده، عملیات پیچیده تنسور تعریف شده است، تا از بیش برازش جلوگیری شود و کارایی آموزش بهبود بخشد. همچنین الگوریتم پس انتشار ابعاد بالا، برای آموزش پارامترهای مدل محاسباتی کانولوشن عمیق در فضای ابعاد بالا پیشنهاد شده است.

در استخراج دیتا، یادگیری ماشین، تشخیص چهره و زمینه‌های دیگر می‌توان از تنسور به عنوان آرایه‌ای با ابعاد زیاد استفاده نمود. در این مقاله [۳۲]، مدل یادگیری عمیق تنسور[۵]، که مدل محاسباتی عمق برای اجرای یادگیری ویژگی‌ها و بازنمایی فضای تنسوری داده‌های کلان پیشنهاد شده است. مدل بازنمایی دیتا بر پایه تنسور ارائه شده است که دیتای ناهمگن متنوع را به شیوه ای هماهنگ، مرتبط ساخته و نمایش می‌دهد. بر اساس چنین بازنمایی، یک خودرمزنگار تنسور[۶] به عنوان ماژول اصلی برای مدل یادگیری عمیق تنسور پیشنهاد شده است. الگوریتم پس انتشار مرتبه بالا، طراحی شده تا پارامترها درخودرمزنگار تنسور را آموزش دهد. در پایان، یک مدل یادگیری عمیق تنسوربا انباشتن خودرمزنگار تنسورهای چندگانه ساخته می‌شود تا سطوح چندگانه ویژگی‌های کلان داده را یاد بگیرد و دیتای ناهمگن مدالیته‌های چندگانه را ترکیب کند. نتایج تجربی، نشان می‌دهد که روش ارائه شده علاوه بر اینکه در بازنمایی داده‌های ناهمگن موفق بوده است در کنار روش‌های یادگیری عمیق بکار گرفته شده، دقت تشخیص، بالایی در دیتا ست‌های STL-۱۰ [۳۳] و CUAVE [۲۶] دارد.

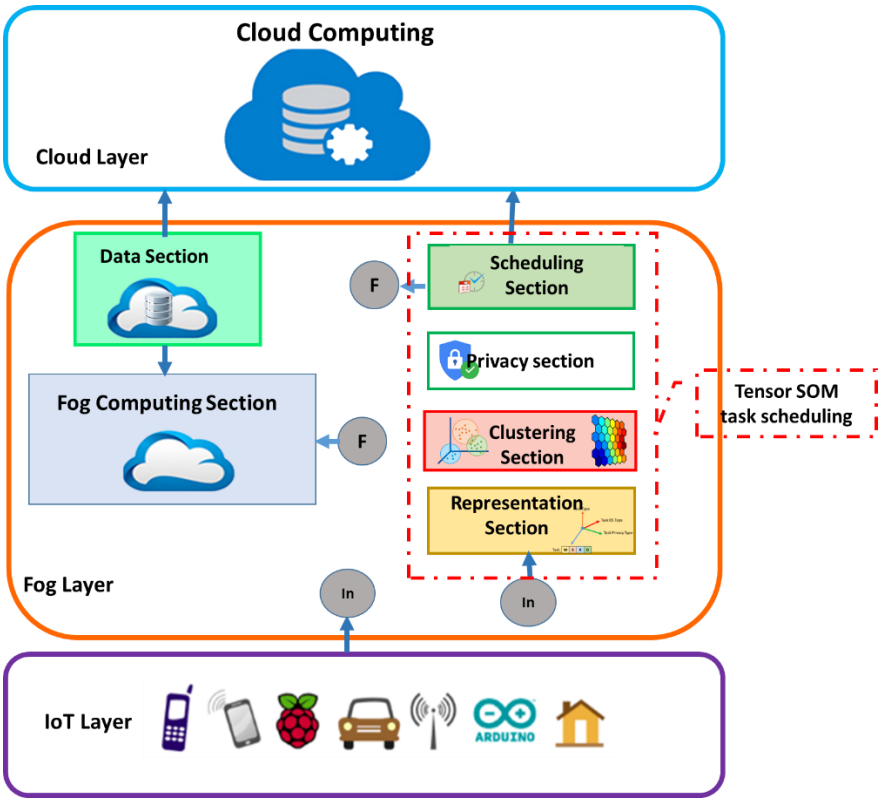
---

[۳] high order tensor
[۴] DCCM
[۵] TDL
[۶] TAE

در [۳٤] مدل تنسوری جهت بازنمایی و یکپاچه‌سازی داده‌های unstructured, semi structured, and structured بیان شده است. این مدل تنسوری علاوه بر یکپارچه‌سازی انواع داده‌ها، دارای مرتبه‌ی گسترده‌ای بوده و به صورت داینامیک مرتبه‌ی جدیدی با اعمال عملگر پیشنهاد شده می‌تواند، حاصل شود. نویسندگان این مقاله، تئوری Core Tensor Equivalence را جهت کمک به طراحی الگوریتم‌های افزایشی تجزیه، اثبات نمودند. همچنین روش Recursive Incremental High Order Singular Value Decomposition (RIHOSVD) را ارئه دادند که برای کاهش ابعاد داده‌های جریانی مناسب می‌باشد. روش ارائه شده از نظر پیچیدگی زمانی، استفاده از حافظه و دقت مورد ارزیابی قرار گرفته است. آن‌ها نشان دادند که روش پیشنهاد شده، جهت استفاده در کاربردهای کلان داده مفید می‌باشد.

۲- روش پیشنهادی

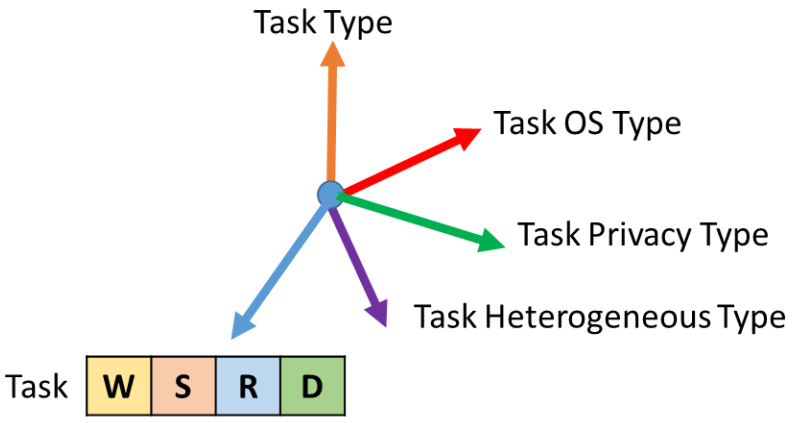
الگوریتم پیشنهادی ارائه شده بانام Tensor_SOM task scheduling دارای چهار بخش (بازنمایی وظایف، خوشه‌بندی و حریم خصوصی و زمان‌بندی) است. در بخش بازنمایی، ویژگی‌های وظایف با استفاده از تنسور با ابعاد بالا بازنمایی می‌شود. سپس به بخش خوشه‌بندی ارسال شده و با استفاده از روش SOM که برای ورودی تنسور، سازگار شده است، وظایف در دو خوشه ابر و مه قرار می‌گیرد. در بخش حریم خصوصی با استفاده از آستانه‌گذاری، وظایفی را که دارای میزان محرمانگی بالایی بوده‌اند، ولی با استفاده از خوشه‌بندی در خوشه‌ی ابر قرارگرفته‌اند، به خوشه‌ی مه منتقل می‌شوند. در نهایت وظایف وابسته به مکان اجرای خود (ابر یا مه)، زمان‌بندی می‌شوند. شکل ۱ روش Tensor_SOM task scheduling و جایگاه آن در لایه مه و تعداد بخش‌های آن را نشان می‌دهد.



شکل ۱: مراحل روش Tensor_SOM task scheduling

الف- بخش بازنمایی وظیفه مبتنی بر تنسور

وظیفه‌ی تولید شده از دستگاه‌های اینترنت اشیا به طور رسمی توسط چهارتایی <W,S,R,D> تعریف می‌شود؛ که در آن W حجم کار وظیفه، S بیت‌های داده همراه وظیفه، R تعداد منابعی که وظیفه برای اجرا نیاز دارد، D مهلت اجرای وظیفه است. پس از دریافت وظیفه از لایه اینترنت اشیا، وظیفه را بر روی تنسوری ۵ بعدی مدل می‌کنیم. با توجه به دستگاهی که وظیفه از آن تولید می‌شود، ویژگی‌هایی مانند نوع وظیفه (حافظه‌ای، رایانشی و توزیع محتوا)، نوع سیستم‌عامل (ویندوز، Obent، android،ios)، میزان حریم‌خصوصی (از درجه خیلی محرمانه تا بدون اهمیت)، نوع ناهمگونی داده مورد نیاز (صوتی، تصویری، متنی و multimedia) استخراج می‌شود. به‌منظور نگهداری کلیه ویژگی‌های برای هر وظیفه، از تنسور ۵ بعدی نشان داده شده در شکل ۲ استفاده می‌شود.



شکل۲: بازنمایی تنسور وظیفه

ب- بخش خوشه‌بندی مبتنی بر Tensor Self Organizing Map (Tensor_SOM)

به‌منظور خوشه‌بندی وظایف، روش SOM را برای ورودی‌های تنسوری، سازگار نموده‌ایم. SOM، دارای سه فاز، رقابتی، همکاری و تطبیق است؛ که برای طراحی SOM برای ورودی تنسوری، باید این سه فاز را سازگار نماییم. در فاز رقابتی، ورودی به‌صورت تنسور ۵ بعدی نشان داده شده در شکل ۲ درنظر گرفته شده‌است. جهت خوشه‌بندی وظایف به دو خوشه، از توپولوژی شامل دو نورون به‌صورت خطی استفاده می‌نماییم با این تفاوت که هر نورون با وزن‌هایی به‌صورت تنسور ۵ بعدی نشان داده می‌شوند. در فاز رقابتی ابتدا تنسور وزن‌ها به‌صورت تصادفی مقداردهی می‌شوند و سپس شباهت تنسور وظایف با تنسور وزنی دو نورون محاسبه می‌شود و بیشترین شباهت با استفاده از ضرب داخلی دو تنسور محاسبه می‌شود. نورون برنده در این فاز شناسایی می‌شود. در فاز همکاری باید نورون برنده، خودش و همسایگانش را تحریک کند تا ورودی جدید تصبیق پیدا کند. در این فاز از تابع گوسی استفاده شده، تا تاثیر نورون بر روی همسایگانش را مشخص کنیم. در فاز تطبیق از قانون ساده‌ی یادگیری هب جهت تغییر وزن نورون‌ها استفاده شده است. جهت ساخت مدل اولیه، در این تحقیق از ۷۰۰۰۰۰، وظیفه استفاده شده است.

ج- بخش حریم خصوصی

پس از خوشه‌بندی وظایف با استفاده از SOM، در این مرحله حریم خصوصی وظایف در خوشه‌ی ابر چک می‌شود و وظایفی که دارای حریم خصوصی خیلی محرمانه و محرمانه هستند از خوشه ابر به مه انتقال پیدا می‌کنند. تا بدون نیاز به روش‌های رمزگذاری بتوان امنیت مورد نیاز وظایف را حفظ نمود.

د) بخش زمان‌بندی

وظایف در هر خوشه با استفاده از الگوریتم Earliest-Deadline-First (EDF) زمان‌بندی می‌شوند و وظایف در خوشه‌ی مه مطابق با زمان‌بندی صورت گرفته، به بخش پردازش در لایه مه ارسال شده و وظایف داخل خوشه ابر به ابر ارسال می‌شود. زمان‌بندی EDF الگوریتم زمان‌بندی مبتنی بر اولویت بوده که در سیستم‌های بلادرنگ کاربرد دارد. هنگامی که وظایف به بخش زمان‌بندی می‌رسد، مطابق با نزدیک‌ترین مهلت زمانی اجرای خود، زمان‌بندی می‌شود [۳۵]. علت استفاده از EDF، بهینه بودن آن در میان الگوریتم‌های زمان‌بندی مختلف در پردازشگرهای واحد است. همانطور که در [۳٦] بیان شده است، اگر مجموعه‌ی وظایف توسط EDF نتواند در محیط بلادرنگ زمان‌بندی شود، آن مجموعه وظایف توسط الگوریتم‌های دیگر هم نمی‌تواند زمان‌بندی شود. برخی از معایب این الگوریتم قابلیت پیش‌بینی پایین و کنترل پایین بر عملکرد و سربار بالای آن و همچنین گرسنگی برای وظایف با مهلت زمانی زیاد است. در این تحقیق جهت رفع گرسنگی الگوریتم EDF از تکنیک سالخوردگی استفاده شده است.

۳- نتایج شبیه سازی

در این بخش به شبیه‌سازی الگوریتم پیشنهاد شده در این تحقیق پرداخته‌ایم. جزییات محیط شبیه‌سازی و پارامترهای مورد استفاده در این تحقیق را بیان می‌نماییم. در این شبیه‌سازی، سه نود ابر و شش نود مه که هر ابر دارای نرخ پردازنده، ۲۵۰۰۰ MIPS و هر مه دارای نرخ Cpu، ۱۲۰۰۰ MIPS، در نظر گرفته شده است. یک نود مه، به‌عنوان هماهنگ کننده در نظر گرفته شده است که مکان وظایف و زمان‌بندی وظایف را مشخص می‌کند. مشخصات محیط شبیه‌سازی شده در جدول ۵-۱ بیان شده است. برای شبیه‌سازی روش پیشنهادی از سیستمی با مشخصات Intel R Core*TM* i۵-٦۲۰۰U, CPU ۲,۳۰GHz و حافظه‌ی ٤GB و از زبان پایتون جهت تولید دیتاست و همچین نرم‌افزار متلب استفاده شده است. در این تحقیق، دیتاست‌ای شامل وظایف و ویژگی‌های بیان شده براساس شبیه‌سازی داده‌های واقعی ساخته شده است. این دیتا ست شامل صدهزار وظیفه با ده ویژگی است و هر وظیفه نیز به همراه خود داده‌ی موردنیاز خود را دارد که این داده‌ها دارای حجم بین ۲۰۰ بیت تا ۵۰۰۰ بیت بوده و هر وظیفه می‌تواند حداکثر ۱۰ داده با سایز مختلف و نوع مختلف با خود به همراه داشته باشد. این دیتا ست بگونه‌ای ساخته شده است تا بتواند داده‌های واقعی در محیط اینترنت اشیا را شبیه‌سازی نماید. جدول ۲ خلاصه ای از مشخصات شبیه‌سازی و دیتاست را نشان می‌دهد.

جدول ۱: مشخصات شبیه‌سازی و دیتاست

پارامتر	مقدار پارامتر
تعداد ابر	۳
نرخ سرعت پردازنده در ابر	MIPS۲۵۰۰۰
تعداد مه	٦
نرخ سرعت پردازنده در مه	MIPS ۱۲۰۰۰
تعداد وظایف	۱۰۰،۰۰۰
تعداد داده‌های همراه	۱۰،۰۰۰
حداکثر داده‌ی مورد نیاز برای هر وظیفه	۱۰
حداقل سایز داده برحسب بیت	۲۰۰ بیت
حدکثر سایز داده برحسب بیت	۵۰۰۰ بیت
میانگین فاصله زمانی رسیدن وظایف	۵ میلی ثانیه
تعداد ویژگی‌های هر وظیفه	۱۰
ویژگی‌های وظیفه	• نوع سیستم عامل • نوع وظیفه •اولویت وظیفه •میزان محرمانگی •ناهمگنی داده •مهلت زمان اجرا •زمان اجرا •منابع مورد نیاز •زمان رسیدن وظیفه •داده همراه وظیفه و حداکثر داده مورد نیاز

جدول ۳ محدوده‌ی مقادیر هر ویژگی و نوع توزیع آن را برای ساخت دیتاست نشان می‌دهد.

جدول ۳: جزییات ویژگی‌های مجموعه داده

نام ویژگی	مشخصه	محدود مقدار	نوع توزیع
Task_Os_Type	نوع سیستم عامل	٤ نوع سیستم عامل Windows، Obent، ios، android	یکنواخت
Task_Type	نوع وظیفه	{۱-۲-۳} حافظه‌ای پردازشی توزیع محتوا	یکنواخت
Task_privacy	میزان محرمانگی	چهار نوع (۱) خیلی محرمانه (۲) عادی	یکنواخت
Task_deadline	مهلت زمانی اجرا	[۱ ... ۲۰] میلی ثانیه	یکنواخت
Task data heterogeneity	ناهمگنی داده	{۱-۲-۳-٤-٥} داده ویدیویی، داده صوتی، داده متنی، داده مولتی مدیا،اطلاعات سنسور	یکنواخت
Task Execuation	زمان اجرا وظیفه	[۱ ... ٦۰] میلی ثانیه	یکنواخت
Task Resource	تعداد منابع مورد نیاز	{۱ ... ۸ }	یکنواخت
Task arrival	زمان رسیدن وظیفه		توزیع پواسون با میانگین زمانی رسیدن پنج میلی ثانیه
Task data	داده مورد نیاز	{ ۱ ... ۱۰۰۰۰} بیت	زیف
Task needed data	حداکثر داده مورد نیاز	{۱ ... ۱۰}	-

در این بخش، روش پیشنهادی رابا چهار روش دیگر مقایسه می‌نماییم. این روش‌ها عبارت‌اند از:

- **روش SOM** : در این روش فقط از ویژگی‌های اصلی وظایف جهت خوشه‌بندی توسط SOM استفاده می‌شود.
- **روش SOM سلسه‌مراتبی**[۷]: از ساختار سلسله‌مراتبی SOM به جای SOM ساده استفاده می‌شود.

[۷] SOM Hierarchical

- **روش DPTO**[۸] [۳۷]: این روش توسط آدهیکاری و همکارانش بیان شده است، در این روش وظایف براساس مهلت‌زمانی شان اولویت‌بندی می‌شوند و به صف چند سطحی دارای بازخورد ارسال می‌شوند. در این روش از سه صف استفاده شده است، وظایف با مهلت زمانی سخت[۹] دارای اولویت بالا بوده و در صف ۱ قرار می‌گیرند و وظایفی که مهلت زمانی آن‌ها نرم[۱۰] بوده در صف دوم و وظایف بدون اولویت در صف سوم قرار می‌گیرند. در صورتی که وظیفه‌ای بعد از مدت زمانی در صف بوده و نوبت آن نرسیده باشد، اولویت آن یک واحد افزایش می‌یابد و به صف با اولویت بالاتر می‌رود. وظایفی که در صف یک و دو قرار دارند، به مه و وظایف صف سه به ابر جهت اجرا ارسال می‌شوند.
- **روش AE-SOM**[۱۱]: این روش توسط شادرو و همکارانش [۳۸] پیشنهاد شده است. در این روش پارامترهای وظایف به شبکه عمیق خودرمزگذار ارسال می‌شود. شبکه‌ی خودرمزگذار ابتدا داده‌های موردنظر را رمزگذاری و سپس رمزگشایی می‌کند. در این شبیه‌سازی از بخش رمزگذاری شبکه به‌عنوان ویژگی‌های جدید داده، استفاده‌شده است. سپس برروی این ویژگی‌های خوشه‌بندی SOM انجام می‌گیرد.

٤-١ بررسی تعداد وظایف اجرا نشده در زمان مقرر

هلت زمانی در اجرا وظایف دارای اهمیت است، لذا در این تحقیق، وظایفی که در زمان مقرر اجرا نشود، به‌عنوان «وظایف تکمیل نشده» و در غیر این صورت «وظایف تکمیل شده» در نظر گرفته می‌شود. جدول ٤، تعداد وظایف تکمیل نشده را به در صد نشان می‌دهد. مطابق با جدول، در روش Tensor_SOM تعداد وظایف تکمیل نشده در کل سیستم برابر ١٤/٠٩ در صد می‌باشد که نسبت به میانگین چهار روش دیگر، ۵۳ درصد بهبود داشته است؛ بنابراین می‌توان نتیجه گرفت که روش پیشنهادی تعداد وظایف تکمیل شده در کل سیستم را بهبود داده است. علت این بهبود نسبت به روش‌های دیگر، در نحوه ی توزیع وظایف در خوشه‌های ابر و مه است. بگونه ای که وظایفی که نیاز پردازشی بیشتری دارند، در خوشه‌ی ابر قرارگرفته اند.

جدول٤: مقایسه روش‌های پیشنهادی از لحاظ درصد وظایف تکمیل نشد

روش	نرخ وظایف تکمیل نشده		
	ابر	مه	ابر و مه
SOM	٦/٩٣	٢٩/٤٨	١٨/٢٥
SOM Hierarchical	٠/٠٠	٤٠/٢٤	٢٥/٩١
DPTO	١٩/٨٥	٧٠/٤٤	٥٧/١٥
AE_SOM	٣٨/٢١	٠/٠٢	١٧/٧٥
Tensor_SOM	٢٣/٠٢	١/٨٧	١٤/٠٩

[۸] Deadline and Priority-aware Task Offloading
[۹] hard
[۱۰] soft
[۱۱] Auto Encoder SOM

٤-٢ بررسی نتایج از لحاظ زمان

در این تحقیق، سه ابر که در هریک تعداد دستورات اجرایی در هر ثانیه ۲۵۰۰۰ دستور و شش مه که هرکدام در هر ثانیه ۱۲۰۰۰ دستور اجرا می‌کنند، درنظر گرفته شده است. با توجه به خوشه‌بندی و زمان‌بندی انجام شده، بعد از ارسال وظایف به مه و ابر، مدت زمان اجرای هر وظیفه محاسبه شده است. مدت زمانی که طول می‌کشد تمامی وظایف ارسالی به مه و ابر اجرا شوند (Makespan)، در جدول ۵ بیان شده است.

جدول ۵: مقایسه روش‌های پیشنهادی از لحاظ زمان اتمام وظایف در ابر مه و کل سیستم

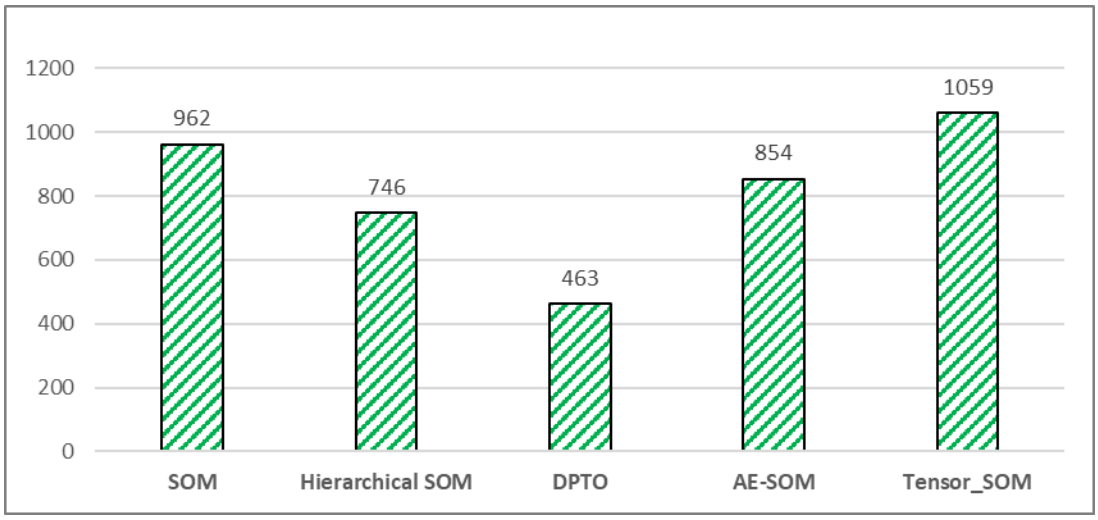
روش	Cloud زمان اتمام وظایف (s)	Fog زمان اتمام وظایف (s)	System زمان اتمام وظایف (s)
SOM	۳۸/۲۲	۸٤/۹۷	۸٤/۹۷
SOM Hierarchical	۲۱/۰٤	۹۹/۲۸	۹۹/۲۸
DPTO	۵۱/٤۱	۹۲/٤٦	۹۲/٤٦
AE_SOM	۹٦/۳۵	۳٦/۵۲	۹٦/۳۵
Tensor_SOM	۸۱/۱۳	٦۱/۵۰	۸۱/۱۳

صدهزار تسک وارد شده به سیستم، در روش SOM، ۸٤/۹۷ ثانیه و در روش Hierarchical SOM ۹۹/۲۸ ثانیه و در روش DPTO ۹۲/٤٦ ثانیه و در روش AE_SOM ۹٦/۳۵ ثانیه و روش Tensor_SOM ۸۱/۱۳ ثانیه نیاز به اجرای تمامی وظایف دارند. به‌منظور بررسی علت اختلاف زمان اتمام وظایف در روش پیشنهادی، میزان درصد توزیع وظایف در خوشه‌های در جدول ٦ آورده شده است. سه روش پیشنهادی، دارای توزیع یکسانی نسبت به میانگین سه روش دیگر است؛ اما با این وجود، وظایفی را در خوشه‌ی ابری قرار داده است که نیاز پردازشی زیاد دارند. این روش از ظرفیت پردازشی ابر به خوبی استفاده نموده و توزیع وظایف را بگونه ای انجام داده است که منجر به کاهش زمان اتمام وظایف و بهبود عملکرد سیستم شده است.

جدول٦: درصد توزیع وظایف در خوشه‌ها

روش	ابر (درصد)	مه (درصد)
SOM	٤۹/۸۰	۵۰/۲۰
SOM Hierarchical	۳۵/٦۲	٦٤/۳۸
DPTO	۲٦/۲۷	۷۳/۷۳
AE_SOM	٤٦/٤۲	۵۳/۵۸
Tensor_SOM	۵۸/۸۰	٤۲/۲۰

به‌منظور بررسی بیشتر، Goodput سیستم را نیز مورد بررسی قرار داده‌ایم. برای محاسبه‌ی Goodput، تعداد وظایفی که تکمیل شده‌اند را بر زمان کل اجرای آن‌ها تقسیم نموده‌ایم. شکل ۳، Goodput در سیستم را برای روش‌های مقایسه شده، نشان می‌دهد. روش Tensor SOM در واحد زمان تعداد بیشتری وظیفه را نسبت به روش‌های دیگر اجرا می‌نماید. این نشان می‌دهد که خوشه‌بندی درروش Tensor SOM بر روی Goodput نیز تاثیرگذار بوده است.



شکل۳: مقایسه goodput برای کل سیستم

٤-۳ بررسی از لحاظ هزینه

برای محاسبه هزینه باید هزینه‌هایی که برای استفاده از سرویس‌های ابر و مه پرداخت می‌شوند را در نظر گرفته و هزینه‌ی کل سیستم را محاسبه نماییم. هزینه‌های در نظر گرفته شده در این تحقیق مطابق با دلارهای گرید ($G) می‌باشد. G$ واحد ارزی بوده که در این شبیه‌سازی برای جایگزینی پول واقعی استفاده شده است. ابر و مه دارای خصوصیات یکسان مانند هزینه‌ی استفاده از پردازنده و هزینه‌ی مصرف حافظه و هزینه‌ی مصرف پهنای باند می‌باشند. با این تفاوت که هزینه‌های ابری نسبت به مه بیشتر بوده و ابر قدرت پردازش بسیار بیشتر نسبت به مه دارد. جهت محاسبه‌ی هزینه‌های ارتباطی، فرض شده است که ارتباط بین حسگرها و app ها با دستگاه ها و ارتباط بین دستگاه‌ها با گره مه از طریق ارتباطات ارزان قیمت بوده و در این شبیه‌سازی در نظر گرفته نشده است. برای ارتباطات داخلی گره‌های مه نیز به دلیل آن‌که در یک شبکه محلی و دردسترس بوده، هزینه اندکی برای آن‌ها در این شبیه‌سازی نسبت به ابر در نظر گرفته شده است. همچنین هزینه‌ی بالایی برای ارتباطات بین گره‌های مه و سرویس‌دهندگان لحاظ شده است. جدول ۷ هزینه‌های در نظر گرفته شده در این شبیه‌سازی را نشان می‌دهد.

جدول ۷: جزییات هزینه

Parameter	Fog	Cloud	Unit
CPU rate	۱۲۰۰۰	۲۵۰۰۰	MIPS
CPU usage cost	۰/۱	۰/۷	G$/s
Memory usage cost	۰/۰۱	۰/۰۲	G$/MB
Bandwidth usage cost	۰/۰۰۱	۰/۰۱	G$/MB

مجموع هزینه‌ها توسط فرمول ۱ در این شبیه‌سازی محاسبه می‌شود.

$$Total\ Cost = \sum_{T_i \epsilon T} Cost(T_i) \qquad (۱)$$

بخش هزینه در این فرمول شامل مجموع هزینه پردازشی C_p و حافظه ای C_m و پهنای باند C_b برای هر وظیفه بوده که با استفاده از فرمول‌های ۲ تا ۵ محاسبه می‌شود.

$$Cost(T_i) = C_p(T_i) + C_m(T_i) + C_b(T_i) \qquad (۲)$$

$$C_p(T_i) = c_۱ * ExeTime(T_i) \qquad (۳)$$

$$C_m(T_i) = c_۲ * Mem(T_i) \qquad (٤)$$

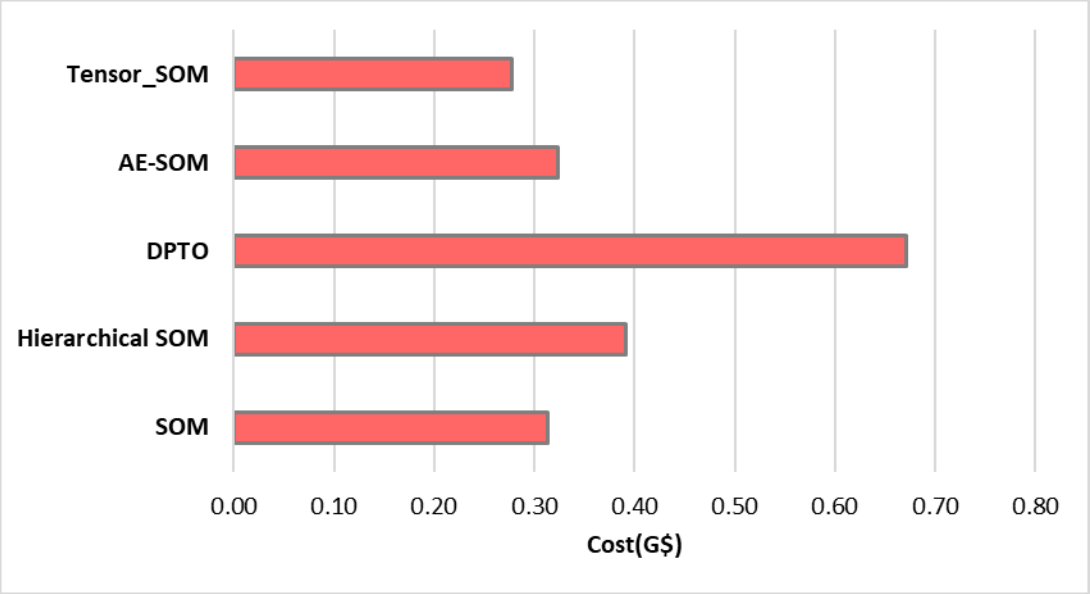
$$C_b(T_i) = c_۳ * BW(T_i) \qquad (٥)$$

$c_۱$هزینه‌ی استفاده از پردازنده در واحد زمان در ابر یا مه است. $c_۲$ هزینه ی استفاده از حافظه برای هر مگابایت داده است.$c_۳$ هزینه ی استفاده ازپهنای باند برای هر مگابایت داده است. جدول ۸ مجموع هزینه برای ابر و مه و کل سیستم را مطابق با فرمول ۱ نشان می‌دهد.

جدول ۸ : مجموع هزینه ابر و مه و کل سیستم

روش	هزینه ابر (G$)	هزینه مه (G$)	هزینه کل سیستم (G$)
SOM	۱۸۸٤۷/۲٥	٦۷۹۹/۹۱	۲٥٦٤۷/۱٦
SOM Hierarchical	۲٤۰۷٦/٦۱	٤۸٦٤/٤٤	۲۸۹٤۱/۰٤
DPTO	۲۳۷۲۲/٦۲	٥۰۲۷/۲٤	۲۸۷٤۹/۸٥
AE_SOM	۲۰۲۲۹/۲۸	٦۳٥٤/۲۰	۲٦٥۸۳/٤۸
Tensor_SOM	۱٥۹٥۸/۷	۷۹۰٤/۸٤	۲۳۸٦۳/٥٤

باتوجه به جدول ۸، روش Tensor SOM، در هزینه نیز نسبت به سایر روش‌ها بهتر می‌باشد. برای مقایسه بهتر، هزینه‌ای که برای هر وظیفه پرداخت می‌شود را محاسبه نموده‌ایم. برای این منظور، نسبت هزینه کل سیستم بر تعداد وظایف تکمیل شده به دست می‌آوریم. شکل ٤ هزینه‌ی زمانبندی یک وظیفه تکمیل شده در کل سیستم را نشان می‌دهد. روش پیشنهادی، هزینه را نسبت به هزینه‌ی یک وظیفه درچهار روش دیگر، بهبود داده اند. روش‌هایTensor_SOM نسبت به میانگین چهار روش مقایسه شده، ٣٤/٦ درصد بهبود یافته است. در نتیجه استفاده از بازنمایی تنسوری که درروش خوشه‌بندی، مورد استفاده قرارگرفته است، منجر شده تا خوشه‌بندی جهت کاهش زمان اتمام وظایف و افزایش goodput و کاهش هزینه یک وظیفه در سیستم، مؤثر باشد.



شکل ٤: هزینه ی زمانبندی یک وظیفه در کل سیستم

٤- نتیجه گیری

در این مقاله روش Tensor_SOM با استفاده از بازنمایی تنسوری و خوشه‌بندی SOM و روش EDF جهت زمانبندی وظایف در اینترنت اشیا معرفی شده است. روش ارائه شده، با روش‌های خوشه‌بندی SOM و SOM سلسله مراتبی، بدون بخش بازنمایی تنسوری و روش DPTO و روش AE-SOM مقایسه شده است. نتایج نشان می‌دهد که بازنمایی تنسوری به‌کار گرفته شده در روشTensor_SOM بر خوشه‌بندی وظایف نسبت به حالت غیر تنسوری، موثر بوده است. تنسور می‌تواند ویژگی‌های استخراجی از وظایف را به شکل یکپارچه در نظر گرفته، منجر به خوشه‌بندی بهتر شود. شبیه سازی روش Tensor_SOM باعث کاهش زمان اتمام وظایف، افزایش goodput و کاهش هزینه در سیستم مبتنی بر ابر و مه می‌شود. بخش حریم خصوصی که در روش TSOM قرار دارد، می‌تواند امنیت وظایف محرمانه را با اجرا در مه نیز تضمین نماید. در این تحقیق، وظایف به صورت مستقل در نظر گرفته شده‌اند، می‌توان در کارهای آینده، روش پیشنهادی را برروی وظایفی وابسته بررسی نموده و همچنین می‌توان خوشه‌بندی را به صورت توزیع شده برروی نودهای مه انجام داد.

۵- مراجع

[۱] Malik A., Om H., "Cloud Computing and Internet of Things Integration: Architecture, Applications, Issues, and Challenges.," در *Rivera W. (eds) Sustainable Cloud and Energy Services.* Springer, Cham, ۲۰۱۸, , pp. ۱-۲٤.

[۲] Vinothiyalakshmi, P., Anitha, R. , "Workload Mining in Cloud Computing using Extended Cloud Dempster–Shafer Theory (ECDST)," *Wireless Personal Communications* , ۲۰۲۰.

[۳] R. K. Behera, S. Gupta and A. Gautam, "Big-data empowered cloud centric Internet of Things," در *International Conference on Man and Machine Interfacing (MAMI)*, ۲۰۱۵.

[٤] Suciu, G., Suciu, V., Martian, A.,et al., "Big Data, Internet of Things and Cloud Convergence – An Architecture for Secure E-Health Applications," *Journal of Medical Systems*, ۲۰۱۵.

[٥] Shabnam Shadroo, Amir Masoud Rahmani, "Systematic survey of big data and data mining in internet of things," *Computer Networks*, جلد ۱۳۹, ۲۰۱۸. pp. ۱۹-٤۷.

[٦] Gubbi, A., Buyya, T., Marusic, S. and Palaniswami M., "Internet of Things (IoT):A vision, architectural elements, and future directions," *Future Generation Computer Systems*, جلد ۲۹, ۲۰۱۳. p. ۱٦٤٥–۱٦٦۰,

[۷] Atzori, L., Lera, A. and Morabito, G., "The Internet of Things: A Survey," *Computer Networks*, جلد ٥٤, شماره ۱٥, p. ۲۷۸۷–۲۸۰٥, ۲۰۱۰.

[۸] Lindong Liu, Deyu Qi, Naqin Zhou, and Yilin Wu, "A Task Scheduling Algorithm Based on Classification Mining in Fog Computing Environment," *Wireless Communications and Mobile Computing*, جلد ۲۰۱۸, ۲۰۱۸, ۲۰۱۸.

[۹] Mehwish Awan, Munam Ali Shah, "A Survey on Task Scheduling Algorithms in Cloud," *International Journal of Computer and Information Technology*, جلد ٤, شماره ۲, ۲۰۱٥.

[۱۰] Fang Y., Wang F., Ge J., "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing," در *In: Wang F.L., Gong Z., Luo X., Lei J. (eds) Web Information Systems and Mining*, Lecture Notes in Computer Science, vol ٦۳۱۸. Springer, , Berlin, Heidelberg, ۲۰۱۰.

[۱۱] L. Yin, J. Luo and H. Luo, "Tasks Scheduling and Resource Allocation in Fog Computing Based on Containers for Smart Manufacturing," *IEEE Transactions on Industrial Informatics*, جلد ۱٤, شماره ۱۰, ۲۰۱۸. pp. ٤۷۱۲-٤۷۲۱,

[۱۲] Tang, C., Hao, M., Wei, X. et al, "Energy-aware task scheduling in mobile cloud computing," *Distributed and Parallel Databases*, جلد ۳٦, شماره ۳, ۲۰۱۸. p. ٥۲۹–٥٥۳,

[۱۳] Tang, C., Xiao, S., Wei, X., Hao, M., Chen, W., "Energy-Efficient and Deadline-satisfied Task Scheduling in Mobile Cloud Computing," در *IEEE International Conference on Big Data and Smart Computing*, Shanghai , ۲۰۱۸.

[۱٤] Hung P.P., Bui TA., Huh EN., "A New Approach for Task Scheduling Optimization in Mobile Cloud Computing," در *In: Park J., Zomaya A., Jeong HY., Obaidat M. (eds) Frontier and Innovation in Future Computing and Communications*, Lecture Notes in Electrical Engineering, vol ۳۰۱. Springer, ۲۰۱٤. Dordrecht,

[۱۵] Tejaswini Choudhari, Melody Moh, and Teng-Sheng Moh, "Prioritized task scheduling in fog computing," در *Proceedings of the ACMSE ۲۰۱۸ Conference (ACMSE '۱۸). Association for Computing Machinery*, New York, NY, USA, ۲۰۱۸.

[۱٦] Hong-Min Chu, Shao-Wen Yang, Padmanabhan Pillai, Yen-Kuang Chen, "Scheduling in Visual Fog Computing: NP-Completeness and Practical Efficient Solutions," در *AAAI Publications, Thirty-Second AAAI Conference on Artificial Intelligence*, ۲۰۱۸.

[۱۷] *Mobile Netw Appl*, Tian, Q., Li, J., Xue, D. et al., "A Hybrid Task Scheduling Algorithm Based on Task Clustering," جلد ۲۵, ۲۰۲۰. p. pages۱۵۱۸–۱۵۲۷,

[۱۸] Topcuoglu H, Hariri S, Wu M-Y, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Trans Parallel Distrib Syst* , جلد ۱۳, شماره ۳, ۲۰۰۲. pp. ۲٦۰-۲۷٤,

[۱۹] Ahmed. A.A. Gad-Elrab, Amin. Y. Noaman,, "A two-tier bipartite graph task allocation approach based on fuzzy clustering in cloud–fog environment," *Future Generation Computer Systems*, جلد ۱۰۳, ۲۰۲۰. pp. ۷۹-۹۰,

[۲۰] Ullah, I., Youn, H.Y. , "Task Classifcation and Scheduling Based on K-Means Clustering for Edge Computing," *Wireless Personal Communications*, جلد ۱۱۳, ۲۰۲۰. p. ۲٦۱۱–۲٦۲٤,

[۲۱] Zhang, G., "Research on job scheduling algorithm in Hadoop," *Journal of Computational* & Xia, Y., Wang, L., Zhao, Q., *Information Systems*, جلد ۷, شماره ۱٦, ۲۰۱۱. p. ۵۷٦۹–۵۷۷۵,

[۲۲] "Fair Scheduler," ۱ Oct ۲۰۱۵. [درون خطی]. Available: http://hadoop.apache.org/docs/r۱,۲,۱/fair_scheduler.html.

[۲۳] S. Wang, T. Zhao and S. Pang, "Task Scheduling Algorithm Based on Improved Firework Algorithm in Fog Computing," *IEEE Access*, جلد ۸, ۲۰۲۰. pp. ۳۲۳۸۵-۳۲۳۹٤,

[۲٤] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," در *International Conference in Swarm Intelligence*, Berlin, Heidelberg, ۲۰۱۰.

[۲۵] Kim, Hyunbum, "Pattern- & Cho, Chihyun & Cho, Sungrae & Lee, Yunseong & Vu, Duc-Nghia & Dao, Nhu-Ngoc *Mobile Information Systems*, Identified Online Task Scheduling in Multitier Edge Computing for Industrial IoT Services," جلد ۲۰۱۸, ۲۰۱۸. pp. ۱-۹,

[۲٦] Geetha Muthusamy, Suganthe Ravi Chandran, "Cluster-based Task Scheduling Using K-Means Clustering for Load Balancing in Cloud Datacenters," *Journal of Internet Technology*, جلد ۲۲, شماره ۱, ۲۰۲۱. pp. ۱۲۱-۱۳۰,

[۲۷] Nazrul Islam, Tensors and Their Applications, New Age International Publisher, ۲۰۰٦.

[۲۸] pp. ,٤ شماره ,۲ جلد *Wiley Interdiscipl. Rev., Comput. Statist.,* H. Abdi and L. J. Williams, "Principal component analysis," ٤۳۳-٤٥۹, ۲۰۱۰.

[۲۹] *th Eur. Conf. Comput. ۷*در M. Brand, "Incremental singular value decomposition of uncertain data with missing values," ۲۰۰۲ ,*Vis. (ECCV)*.

[۳۰] *th ACM SIGKDD Int. ۱۲*در J. Sun, D. Tao, and C. Faloutsos, " Beyond streams and graphs: Dynamic tensor analysis," ۲۰۰٦ ,*Conf. Knowl. Discovery Data Mining (KDD)*.

[۳۱] pp. ٤٥٥-٥۰۰, ,٥۱ جلد *SIAM Review,* Tamara G. Kolda and Brett W. Bader, "Tensor Decompositions and Applications," ۲۰۰۹.

[۳۲] Peng Li ; Zhikui Chen ; Laurence Tianruo Yang ; Qingchen Zhang ; M. Jamal Deen , "Deep Convolutional Computation جلد ۱٤, شماره *IEEE Transactions on Industrial Informatics* , Model for Feature Learning on Big Data in Internet of Things," pp. ۷۹۰ - ۷۹۸ , ۲۰۱۸. ,۲

[۳۳] *IEEE* W. Wang and M. Zhang, "Tensor Deep Learning Model for Heterogeneous Data Fusion in Internet of Things," pp. ۳۲-٤۱, ۲۰۲۰. جلد ٤, شماره ۱, *Transactions on Emerging Topics in Computational Intelligence,*

[۳٤] Available: http://people.csail.mit.edu/siracusa/avdata/. [درون خطی]. ۱۹۹۰.

[۳٥] L. Kuang, F. Hao, L. T. Yang, M. Lin, C. Luo and G. Min, "A Tensor-Based Approach for Big Data Representation and pp. ۲۸۰-۲۹۱, ۲۰۱٤. جلد ۲, شماره ۳, *IEEE Transactions on Emerging Topics in Computing,* Dimensionality Reduction,"

[۳٦] D., et al., "A simulation of priority based earliest deadline first P. R., Singh, Gupta, G., Kumawat, V. K., Laxmi, *Soft Computing & First International Conference on Networks* ۲۰۱٤در scheduling for cloud computing system," Guntur, ۲۰۱٤. ,*(ICNSC ۲۰۱٤)*

[۳۷] Ali, S. K. F., Hamad, M. B. , "Implementation of an EDF Algorithm in a Cloud Computing Environment using the *International Conference on Computing, Control, Networking, Electronics and Embedded Systems* در CloudSim Tool," Khartoum, ۲۰۱٥. ,*Engineering*

[۳۸] M. Adhikari, M. Mukherjee and S. N. Srirama, "DPTO: A Deadline and Priority-Aware Task Offloading in Fog pp. ,۷ شماره ,۷ جلد *IEEE Internet of Things Journal,* Computing Framework Leveraging Multilevel Feedback Queueing," ٥۷۷۳-٥۷۸۲, ۲۰۲۰.

[۳۹] The two-phase scheduling based on deep learning in the " ,Shabnam Shadroo, Amir Masoud Rahmani, Ali Rezaee ۲۰۲۰ ,*Computer Networks* ",Internet of Things.