

رایانش مه : بررسی روند ها ، معماری ، الزامات و اهداف تحقیق

Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions

**RANESH KUMAR NAHA¹, SAURABH GARG¹, (Member, IEEE),
DIMITRIOS GEORGAKOPOULOS², (Member, IEEE),
PREM PRAKASH JAYARAMAN², (Member, IEEE), LONGXIANG GAO³, (Senior Member, IEEE),
YONG XIANG³, (Senior Member, IEEE), AND RAJIV RANJAN⁴, (Senior Member, IEEE)**

¹School of Technology, Environments and Design, University of Tasmania, Hobart, TAS 7001, Australia

²Faculty of Science, Engineering and Technology, Swinburne University of Technology, Melbourne, VIC 3122, Australia

³School of Information Technology, Deakin University, Burwood, VIC 3125, Australia

⁴School of Computing, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

Corresponding author: Longxiang Gao (longxiang.gao@deakin.edu.au)

The work of R. K. Naha was supported by the University of Tasmania for providing Tasmania Graduate Research Scholarship.

دانشگاه محلات

استاد راهنما: محمد خلیلی

دانشجو: نیما ستاری

چکیده: فناوری های نوظهور مانند اینترنت اشیا برای پردازش برنامه در زمان اجرا به محاسبه آگاهانه تاخیر احتیاج دارند. در محیط اینترنت اشیا ، وسایل متصل مقدار زیادی داده تولید می کنند که به طور کلی به آنها داده های بزرگ گفته می شود. داده های تولید شده از دستگاه های اینترنت اشیا کلاً به دلیل خدمات درخواستی و ویژگی های مقیاس پذیری پارادایم رایانش ابری ، معمولاً در زیرساخت ابر پردازش می شوند. با این حال ، پردازش درخواست های کاربردی اینترنت اشیا روی ابر به طور انحصاری برای برخی از برنامه های اینترنت اشیا ، به ویژه موارد حساس به زمان ، یک راه حل کارآمد نیست.

برای پرداختن به این مسئله ، رایانش مه ، که بین دستگاههای cloud و IoT قرار دارد ، پیشنهاد شد. به طور کلی ، در محیط رایانش مه ، دستگاه های اینترنت اشیا به دستگاه های مه متصل می شوند. این دستگاههای مه در نزدیکی کاربران قرار دارند و وظیفه واسطه محاسبه و ذخیره را بر عهده دارند. یکی از چالش های کلیدی در اجرای برنامه های اینترنت اشیا در یک محیط رایانش مه ، تخصیص منابع و برنامه ریزی کارها است.

تحقیقات رایانش مه هنوز در مراحل ابتدایی است و تحقیقات مبتنی بر طبقه بندی در مورد نیازهای زیرساخت ها ، سیستم عامل و برنامه های نقشه برداری شده با تحقیقات فعلی هنوز نیاز است. این بررسی به صنعت و جامعه تحقیقاتی کمک می کند تا الزامات رایانش مه را ترکیب و شناسایی کنند.

این مقاله با مروری بر رایانش مه آغاز می شود که در آن تعریف محاسبه مه ، روند تحقیق و تفاوت های فنی بین مه و ابر بررسی می شود. سپس ، ما چندین معماری پیشنهادی محاسباتی مه را بررسی کرده و اجزای سازنده این معماری را با جزئیات شرح می دهیم. از این رو ، نقش هر مولفه مشخص خواهد شد ، که به استقرار رایانش مه کمک خواهد کرد. در مرحله بعد ، طبقه بندی رایانش مه با در نظر گرفتن الزامات الگوی رایانش مه ارائه می شود.

ما همچنین در مورد کارهای تحقیقاتی موجود و خلا در تخصیص و برنامه ریزی منابع ، تحمل خطا ، ابزارهای شبیه سازی و ریز خدمات مبتنی بر مه بحث می کنیم. سرانجام ، با پرداختن به محدودیت های کارهای پژوهشی فعلی ، ما برخی از موارد باز را ارائه می دهیم ، که جهت تحقیق آینده برای الگوی رایانش مه را تعیین می کند.

۱- مقدمه: افراد و سازمان ها برای انجام کارهای روزمره به طور فزاینده ای به کامپیوتر و دستگاه های هوشمند وابسته می شوند. این دستگاه ها از طریق حسگرها و برنامه های مختلف در حال تولید اطلاعات هستند. در نتیجه ، سازمان ها به طور منظم در حال تولید و ذخیره مقدار زیادی داده هستند [1]. پس از همه گیر شدن اینترنت اشیا ، داده های تولید شده توسط حسگرها بسیار زیاد شده است. با این افزایش ناگهانی حجم داده های تولید شده و عدم توانایی پایگاه های داده معمولی در پردازش اشکال مختلف داده های ساختاریافته و غیر ساختاری ، تجزیه و تحلیل داده های بزرگ در سال های اخیر مورد توجه بسیاری قرار گرفته است. اکنون هر سازمانی تجزیه و تحلیل داده های جمع آوری شده را در اولویت قرار می دهد تا بینش مفید را به منظور تصمیم گیری مهم استخراج کند [2].

امروزه ، سازمان ها به دلیل تغییر در رایانش ابری به دلیل قابلیت دسترسی ، مقیاس پذیری و ویژگی های پرداخت به ازای استفاده ، به زیرساخت آی تی پویا نیاز دارند. رایج ترین سرویس های ارائه شده توسط ابر: نرم افزار به عنوان سرویس (SaaS) ، بستر به عنوان سرویس (PaaS) و زیرساخت به عنوان سرویس (IaaS) شناخته می شوند که همه به سمت ، هر چیزی به عنوان سرویس (XaaS) می روند [3] با این حال ، داده های تولید شده از میلیاردها حسگر ، که به آنها داده های بزرگ گفته می شود ، نمی توانند در ابر منتقل و پردازش شوند. علاوه بر این ، برخی از برنامه های اینترنت اشیا باید سریعتر از ابر پردازش شوند. این مشکل را می توان با استفاده از الگوی رایانش مه ، که از قدرت پردازش دستگاه های واقع در نزدیکی کاربران (قدرت محاسبه ساکن) برای پشتیبانی از ذخیره سازی ، پردازش و شبکه در لبه استفاده می کند ، حل کرد [4].

رایانش مه یک مفهوم محاسبات غیرمتمرکز است ، که منحصراً به هیچ یک از مولفه های مرکزی مانند رایانش ابری متکی نیست [5] ، [6]. با استفاده از منابع بیکار دستگاه های مختلف در نزدیکی کاربران ، می تواند بر مشکل تأخیر زیاد ابر فائق آید. با این حال ، رایانش مه برای پردازش های پیچیده به ابر متکی است. برخلاف رایانش ابری ، رایانش مه یک مفهوم محاسباتی غیرمتمرکز است ، جایی که بسیاری از دستگاه های اطراف ما ، که ظرفیت محاسبه دارند ، استفاده می کنند. در حال حاضر ، حتی یک تلفن هوشمند با مشخصات پایین نیز دارای ظرفیت پردازش است ، گاهی اوقات با چندین هسته. از این رو ، بسیاری از دستگاه ها مانند تلفن های هوشمند ، سوئیچ ها ، روترها ، ایستگاه های پایه و سایر دستگاه های مدیریت شبکه مجهز به قدرت پردازش و ظرفیت ذخیره سازی می توانند مانند دستگاه های مه بازی کنند. منابع این دستگاه ها خارج از ساعات اوج مصرف بیکار هستند.

بسیاری از موضوعات تحقیقاتی مربوط به رایانش مه به دلیل اتصال همه جا و سازمان ناهمگن آن در حال ظهور است. در الگوی رایانش مه ، موارد اصلی نیازها و استقرار محیط رایانش مه است. این به این دلیل است که دستگاه هایی که در محیط های مه وجود دارد ناهمگن هستند: بنابراین ، سوالی که پیش می آید این است که چگونه محاسبه مه با چالش های جدید مدیریت منابع و مدیریت نارسایی در چنین محیط ناهمگن برخورد می کند؟ از این رو ، لازم است نیازهای اساسی سایر جنبه های مرتبط مانند مسائل استقرار ، شبیه سازی ها ، مدیریت منابع ، تحمل خطا و خدمات مورد بررسی قرار گیرد. چندین بررسی [5] ، [7] - [12] در مورد رایانش مه انجام شده است. در اینجا ، ما به طور خلاصه حوزه های بررسی و تمرکز این بررسی ها را ارائه می دهیم.

مفاهیم مشابه رایانش مه ، تعاریف ، سناریوهای کاربردی و موضوعات متعدد توسط یک مطالعه توصیف شده است [7]. هو و همکاران [8] معماری سلسله مراتبی محاسبات و فناوری های مه مانند محاسبات ، ارتباطات و فناوری های ذخیره سازی ، یعنی مدیریت منابع ، امنیت و محافظت از حریم خصوصی را فراهم می کند که از استقرار و کاربرد مه پشتیبانی می کند. باکرلی و همکاران [9] با استفاده از یک محاسبه مه و اینترنت همه چیز (IoE) با یک دیدگاه یکپارچه از رایانش مه و IoE استفاده کرد. وارشنی و سیمهان [10] اندازه های مختلفی از مشخصات برنامه ، معماری سیستم و انتزاعات سکوی اکوسیستم های لبه ، مه و ابر را بررسی کردند. پرا و همکاران [11] دامنه رایانش مه را از دیدگاه سیستم عامل توسعه دهندگان و کاربران نهایی به منظور ایجاد زیرساخت سنجش پایدار برای برنامه های شهر هوشمند بررسی کرد. محمود و همکاران [5] طبقه بندی رایانش مه با توجه به چالش های شناسایی شده و ویژگی های کلیدی آن ارائه شده است. طبقه بندی پیشنهادی طبقه بندی آثار موجود در رایانش مه را ارائه می دهد. مورادیان و همکاران [12] معماری و الگوریتم های مه را براساس شش معیار ارزیابی مختلف ، یعنی ناهمگنی ، مدیریت QoS ، مقیاس پذیری ، تحرک ، فدراسیون و توانایی رابط کاربری بررسی کرد. با این حال ، هیچ یک از مطالعات بر اساس نیاز زیرساخت ها ، سیستم عامل و کاربردها در رایانش مه ، تاکسونومی را بررسی نکرده است. علاوه بر این ، هیچ یک از آنها به طور جامع تحمل خطا ، عملکرد منابع انسانی یا ریز خدمات در رایانش مه را بررسی نکردند. ما مسائل فعلی فوق را در نظر می گیریم و به طور گسترده در مورد این بحث می کنیم و همچنین نشان می دهیم که چگونه در برخی موارد از راه حل های مربوط به محاسبات ابری می توان در مه استفاده کرد. مشارکتهای این کار مروری را می توان به شرح زیر خلاصه کرد:

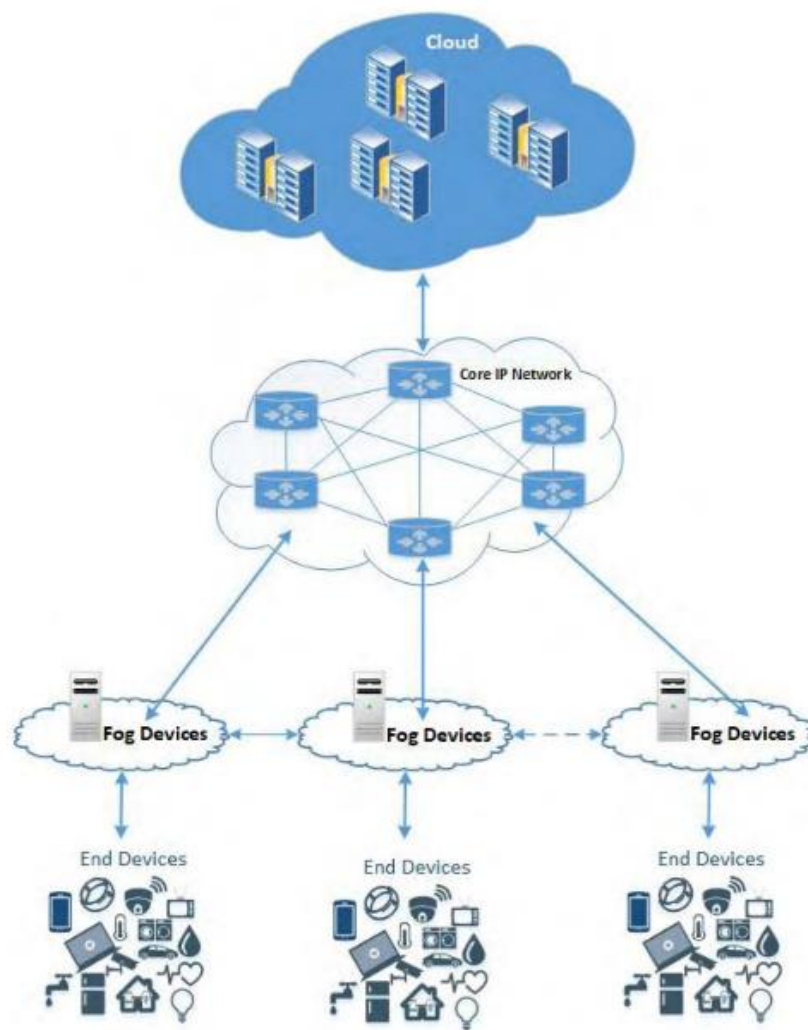
- A. با بررسی تعداد کارهای پژوهشی منتشر شده و وقایع جستجو در Google Scholar ، روند تحقیق در رایانش مه را ارائه دهید.
- B. مرور چندین معماری محاسباتی مه و ارائه معماری دقیق ، زیرا بیشتر محققان قبلی فقط معماری سطح بالایی را ارائه داده اند.
- C. با در نظر گرفتن الزامات زیرساخت ، سیستم عامل و کاربرد در الگوی رایانش مه ، طبقه بندی ارائه دهید.
- D. شکافهای تحقیقاتی رایانش مه در برنامه ریزی و تخصیص منابع ، تحمل خطا ، ابزارهای شبیه سازی و ریز خدمات مبتنی بر مه را شناسایی کنید.
- E. به محدودیت های کارهای تحقیقاتی فعلی و برخی از مسائل باز در زیرساخت ها ، سیستم عامل ها و برنامه ها بپردازید.
- از این نظرسنجی ، صنعت و جامعه تحقیقاتی قادر خواهند بود تا با درک بهتری از مدیریت منابع در مه ، نیازهای ایجاد محیط رایانش مه را درک کنند.
- ادامه این مقاله به شرح زیر است: بخش ۲ تعاریف بررسی و روند تحقیق در رایانش مه. مقایسه فنی بین پارادایمهای مه و ابر در بخش ۳ ارائه شده است. بخش ۴ پارادایم محاسبات مشابه رایانش مه را مورد بحث قرار می دهد. بخش ۵ آثار مربوط به معماری رایانش مه را ارائه می دهد و در مورد مولفه های معماری رایانش مه صحبت می کند. بخش ۶ طبقه بندی رایانش مه را با بررسی نیازهای آن نشان می دهد. بخش ۷ ابعاد مختلف کاربرد رایانش مه را ارائه می دهد. بخش ۸ در مورد فن آوری پیشرفته رایانش مه در مورد اجاره بحث می کند. بخش ۹ موارد باز و جهت گیری آینده را ارائه می دهد. بخش ۱۰ مقاله را به پایان می رساند.

۲- بررسی اجمالی رایانش مه : اصطلاح "رایانش مه" در سال 2012 توسط محققان سیسکو سیستمز [13] پیشنهاد شد. پردازش منطق کاربردی و داده ها در لبه یک مفهوم جدید نیست. مفهوم محاسبه لبه در حدود دهه 2000 [14]، [15] و یک مفهوم مشابه دیگر، Cloudlets، در سال 2009 مطرح شد [16]. Cloudlets و رایانش مه پیشرفت یک مفهوم مشابه هستند که حول پردازش در سطح لبه می چرخد. در حالی که Cloudlet ها در شبکه تلفن همراه اعمال می شوند، رایانش مه برای موارد متصل مانند IoT که با مفهوم IoT بازی می کنند، اعمال می شود [17].

مه یک الگوی محاسباتی مجازی و غیر مجازی است که خدمات شبکه، ذخیره سازی و محاسبات را در میان سرورهای ابری و دستگاه های اینترنت اشیا فراهم می کند [4]، [13] با این حال، این سرویس ها به طور کامل در لبه شبکه قرار ندارند. مه یک رویکرد محاسباتی توزیع شده است که عمدتاً بر روی تسهیل برنامه های کاربردی متمرکز است، که به خدمات تأخیر کم احتیاج دارند [18]، رایانش مه همچنین از خدمات آگاهانه غیر تأخیر پشتیبانی می کند. بدیهی است که اگر حجم پردازش آنقدر زیاد نباشد، استفاده از منابع بیکاری در کنار کاربران باعث بهبود عملکرد کلی سرویس می شود. تعداد زیادی از گره های ناهمگن به مه متصل می شوند. این گره ها از جمله حسگرها و محرک ها هستند [13]. محاسبه در دستگاه های مه در صورت لزوم انجام می شود و امکانات ذخیره سازی نیز برای مدت کوتاهی در دسترس است، حداقل در اکثر دستگاه های مه. محاسبه حساس به زمان در مه بدون درگیر شدن اشخاص ثالث انجام می شود و در بیشتر موارد توسط دستگاه های پردازش مه انجام می شود. به گفته یی و همکاران [7]، الگوی رایانش مه از اجرای سرویس های جدید یا توابع و برنامه های اساسی کار خالص در یک محیط جعبه ماسه شبیه به Cloudlets پشتیبانی می کند. با این حال، این موضوع هنوز یک چالش تحقیقاتی است زیرا این سوال که مه چگونه این خدمات را ارائه می دهد هنوز باقی است. علاوه بر این، آیا مه ارائه دهنده خدمات ابری خواهد داشت یا به طور کلی مانند یک موجود واحد خواهد بود؟ شکل (۱) یک مدل اساسی رایانش مه را نشان می دهد. دستگاه های مه، سرورهای مه، و دروازه ها اجزای اصلی محاسبه در محیط مه هستند. هر دستگاهی که قابلیت محاسبه، شبکه و ذخیره سازی داشته باشد می تواند به عنوان یک دستگاه مه عمل کند. این دستگاه ها شامل جعبه های تنظیم بالا، سوئیچ ها، روترها، ایستگاه های پایه، سرورهای پروکسی یا هر دستگاه محاسباتی دیگری هستند. سرورهای مه که چندین دستگاه Fog و Fog gateways را مدیریت می کنند، وظیفه خدمات ترجمه بین دستگاه های همگن در محیط رایانش مه را دارند. Fog gateways همچنین خدمات ترجمه بین

اینترنت اشیا، مه و لایه های ابری را ارائه می دهد. چالش های جدیدی در این الگوی محاسباتی
نوظهور در چند سال گذشته ظاهر شده است.

در این بخش، ما در مورد تعاریف مختلف رایانش مه بحث می کنیم و رایانش مه را از دیدگاه
خود تعریف می کنیم. علاوه بر این، ما روند تحقیق در رایانش مه را مورد بحث و تحلیل قرار می
دهیم. در آخر، ما تفاوت های فنی بین محاسبه مه و محاسبات ابری را مقایسه می کنیم.



شکل 1. مدلی از محاسبات مه.

تعریف رایانش مه : رایانش مه یک الگوی محاسباتی توزیع شده است که پردازش آن در لبه شبکه با یکپارچه سازی یکپارچه زیرساخت های ابر انجام می شود. این امکان محاسبه را برای محیط های اینترنت اشیا یا سایر برنامه های کاربردی حساس به تأخیر فراهم می کند. تخمین زده می شود که تا سال 2020 حدود 50 میلیارد "چیز" به اینترنت متصل شود [19]. انتقال تمام داده ها از تمام دستگاه های متصل برای پردازش روی ابر به پهنای باند و ذخیره سازی گسترده ای نیاز دارد. همه دستگاه ها از طریق IP به کنترل کننده متصل نیستند اما توسط برخی دیگر از پروتکل های صنعتی IoT متصل می شوند. به همین دلیل ، برای پردازش یا ذخیره سازی اطلاعات از دستگاه های اینترنت اشیا نیز به یک فرایند ترجمه نیاز است. محققان مختلف محاسبه مه را به روش های مختلف تعریف کرده اند. برخی از مثالها به شرح زیر است:

- “رایانش مه یک پلتفرم بسیار مجازی است که خدمات محاسبه ، ذخیره سازی و شبکه را بین دستگاه های اینترنت اشیا و مراکز داده سنتی رایانش ابری ارائه می دهد ، اما به طور انحصاری در لبه شبکه قرار ندارد. “ [13]
 - “رایانش مه یک سناریو است که در آن تعداد زیادی از دستگاه ها (بی سیم و گاهی اوقات خودمختار) با یکدیگر ارتباط برقرار می کنند و به طور بالقوه برای انجام کارهای ذخیره سازی و پردازش بدون دخالت اشخاص ثالث با شبکه همکاری می کنند. این وظایف می تواند برای پشتیبانی از عملکردهای اساسی شبکه یا سرویس ها و برنامه های جدیدی باشد که در یک محیط جعبه ماسه اجرا می شوند. کاربرانی که بخشی از دستگاههای خود را برای میزبانی این خدمات اجاره می کنند ، انگیزه هایی برای این کار دریافت می کنند. “ [20]
 - “اصطلاح Fog computing یا Edge Computing به این معنی است که سیستم های مه به جای میزبانی و کار از یک ابر متمرکز ، در انتهای شبکه کار می کنند. این اصطلاحی برای قرار دادن برخی فرآیندها و منابع در لبه ابر است ، به جای ایجاد کانال هایی برای ذخیره سازی و استفاده از ابر. “ [21]
- اولین تعریف از رایانش مه توسط بونومی و همکاران ارائه شد. [13] ، جایی که آنها الگوی محاسبات را به عنوان یک پلتفرم کاملاً مجازی مطرح کردند. با این حال ، برخی از دستگاه های اینترنت اشیا مانند تلفن های هوشمند مجازی نیستند ، اما همچنین می توانند بخشی از زیرساخت های مه باشند ، زیرا برخی از پردازش ها هنوز هم می توانند انجام شوند. به گفته سیکو [22] ،

الگوی رایانش مه مکانی ایده آل برای تجزیه و تحلیل بیشتر داده های نزدیک دستگاههایی است که آن داده ها را بلافاصله تولید می کنند و بر روی آنها عمل می کنند. مه در نزدیکی چیزهایی قرار دارد که قادر به پردازش و کار بر روی داده های تولید شده هستند. دستگاه هایی که در محیط مه قرار دارند به عنوان دستگاه های مه شناخته می شوند. این گره ها را می توان در هر مکان با اتصال به شبکه مستقر کرد: در تیر برق ، در کف کارخانه ، در کنار جاده ، در امتداد خط راه آهن ، در یک وسیله نقلیه ، داخل یک مرکز خرید ، روی یک دکل نفتی و غیره دستگاهی که قابلیت پردازش ، ذخیره سازی ، حافظه و شبکه را دارد می تواند مانند یک دستگاه مه عمل کند. اگرچه مه ابر را گسترش می دهد ، اما از نظر فنی در بین ابر و دستگاه های اینترنت اشیا قرار دارد و کارهای پردازش و ذخیره سازی را در نزدیکی کاربر انجام می دهد. یی و همکاران [7] اظهار داشت که تعریفی که توسط وکوئرو و رودرو-مرینو ارائه شده است [20] قابل بحث است و هنوز به تعریفی که بتواند بین رایانش مه و سایر الگوی محاسبات مرتبط تفاوت قائل شود ، لازم است. تعریف ارائه شده توسط IBM [21] نشان دهنده Edge و Fog محاسبات به عنوان الگوی محاسبات یکسان است. به گفته شی و همکاران [23] ، رایانش مه بیشتر در سمت زیرساخت متمرکز می شوند در حالی که محاسبات لبه بیشتر در قسمت چیزها تمرکز دارند. علاوه بر این ، محاسبات Edge به طور خودجوش با هیچ سرویس مبتنی بر ابر مانند SaaS ، IaaS و PaaS ارتباط ندارد [5]. به طور خلاصه ، جدول 1 خلاصه تعاریف مه ارائه شده توسط کارهای مختلف تحقیقاتی است. با توجه به تعاریف فوق ، ما محاسبه مه را به صورت زیر تعریف می کنیم:

■ رایانش مه یک سکوی محاسباتی توزیع شده است که بیشترین پردازش توسط دستگاه های انتهایی یا لبه مجازی و غیرمجازی انجام می شود. همچنین برای پردازش بدون تأخیر و ذخیره طولانی مدت داده های مفید با قرار گرفتن در بین کاربران و ابر ، با ابر همسان است.

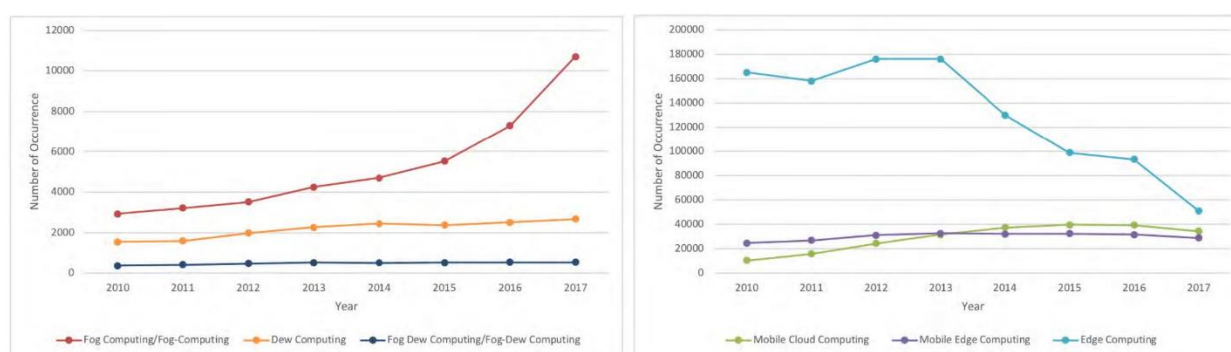
در تعریف ما ، ما همه دستگاه های دارای ظرفیت محاسبات و ذخیره سازی را به عنوان دستگاه های مه در نظر گرفتیم و همچنین دقیق تر نقش ابر را در محیط رایانش مه شناسایی کردیم.

Defined by	Characteristics
Bonomi et al. [13]	Highly virtualized
	Reside between IoT devices and cloud
	Not exclusively located at the edge
Cisco Systems [22]	Extends the Cloud
	Generally used for IoT
	Can be deployed anywhere
	Fog device consists of processing, storage, and network connectivity
Vaquero and Roderio-Merino [20]	heterogeneous, ubiquitous and decentralised devices communication
	Storage and processing done without third party invention
	Run in a sandboxed environment
	Leasing part of users devices and provide incentive
IBM [21]	Defined Fog and Edge computing as similar
	Not depends on centralized cloud
	Resides at network ends
	Place some resource and at the edge of the cloud
Proposed Definition	Virtualization and non-virtualization characteristics
	Association with the cloud for non-latency-aware processing and storage
	Any edge device with available processing power and storage capability can be act as a Fog device
	Always resides between end users and cloud

نمودار ۱- خلاصه ای از تعاریف رایانش مه

F. روند تحقیق در رایانش مه : در چند سال گذشته در صنایع و دانشگاهها توجه فزاینده ای نسبت به پردازش اطلاعات نزدیک به کاربران مشاهده شده است. مدیریت داده های تولید شده از اینترنت اشیا در سطح لبه به بهبود زمان پردازش کلی کمک می کند. در این بخش ، ما مه و سایر روندهای مرتبط فناوری را در چند سال گذشته در جامعه تحقیقاتی بررسی می کنیم. طبق چرخه هایپ گارتنر [24] ، اوج فناوری در حال ظهور خانه هوشمند است که با ترکیب محیط رایانش مه بهتر شکل می گیرد. چرخه هایپ [24] الگوهای متداول فناوری های جدید پرتعداد را نشان می دهد. رایانش مه همچنین می تواند خدمات خانه هوشمند آگاه از تأخیر را به روشی کارآمدتر و راحت تر ، به ویژه برای برنامه های کاربردی خانه هوشمند در موقعیت اضطراری ، فعال کند. طبق نمایش چرخه هایپ گارتنر ، برخی دیگر از فناوری های تأثیرگذار شامل دستیارهای مجازی ، وسایل نقلیه خودمختار ، سیستم عامل های اینترنت اشیا ، ربات های هوشمند ، محاسبات لبه ای و فضاهای کاری هوشمند

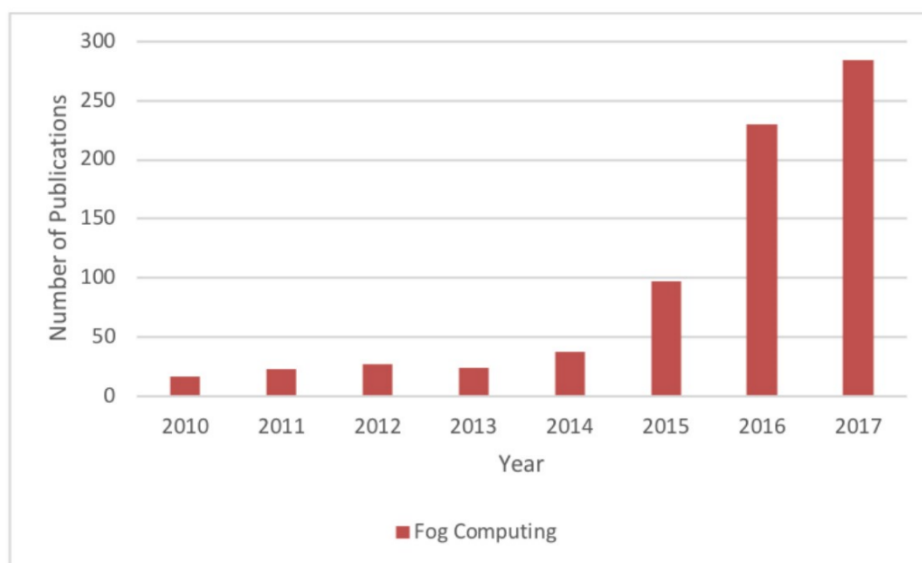
هستند که برای پشتیبانی از برنامه های آگاه از تأخیر مورد نیاز هستند. تمام این فناوری های ذکر شده به دلیل حساسیت تأخیر ، اتصال به ابر و قابلیت پردازش داده در سطح لبه ، می توانند از پشتیبانی الگوی رایانش مه بهره مند شوند. به جز فناوری وسایل نقلیه خودمختار ، سایر فناوری های فوق الذکر در 10 سال آینده به آستانه تصویب بازار خواهند رسید. علاوه بر تجزیه و تحلیل چرخه اعتیاد به مواد مخدره ، ما در جستجوی مه و سایر فن آوری های مرتبط در Google Scholar تجزیه و تحلیل کردیم. علاوه بر این ، تعداد مقالات موجود در کتابخانه های دیجیتال مختلف مربوط به مه نیز مورد تجزیه و تحلیل قرار گرفت.



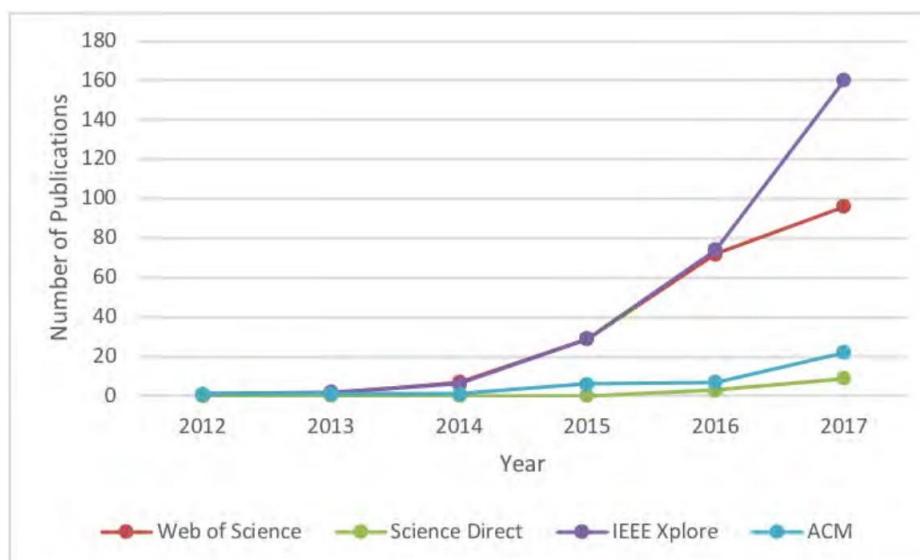
شکل ۲- جستجو در فناوری های مشابه مانند مه در Google Scholar

جستجوهای Google Scholar در انواع مختلف فناوری های مشابه مه در چند سال گذشته مورد بررسی قرار گرفت ، همانطور که در شکل 2 ارائه شده است. با توجه به داده ها ، رایانش لبه بالاترین مورد جستجو شده در Google Scholar است که با سایر فن آوری های مشابه مقایسه شده است. با این حال ، روند جستجو در هشت سال گذشته برای رایانش لبه ای بیش از سه برابر کاهش یافته است. رایانش ابری موبایل و رایانش لبه تلفن همراه دو الگوی رایانش دیگر با جستجوی برتر پس از رایانش لبه هستند. کمترین روند مشاهده شده برای رایانش شبهم (Dew Computing) و رایانش شبهم - مه است. در حالی که روند رایانش لبه در حال کاهش است ، رایانش مه مربوط به جستجوهای علمی سال به سال در حال افزایش است و از 2010 تا 2017 2.5 برابر افزایش یافته است. این نشان می دهد که رایانش مه سریعترین رشد منطقه تحقیقاتی در دانشگاه است و دارای یک سطح عالی خواهد بود. تأثیر بر صنعت نیز هست. جستجوی موضوع رایانش مه در Web of Science نشان می دهد که تعداد مقالات علمی بین سالهای 2015 و 2016 بیش از دو برابر شده است ، همانطور که در شکل 3 آمده است. اولین مقاله با عنوان "رایانش مه " در سال 2012 منتشر

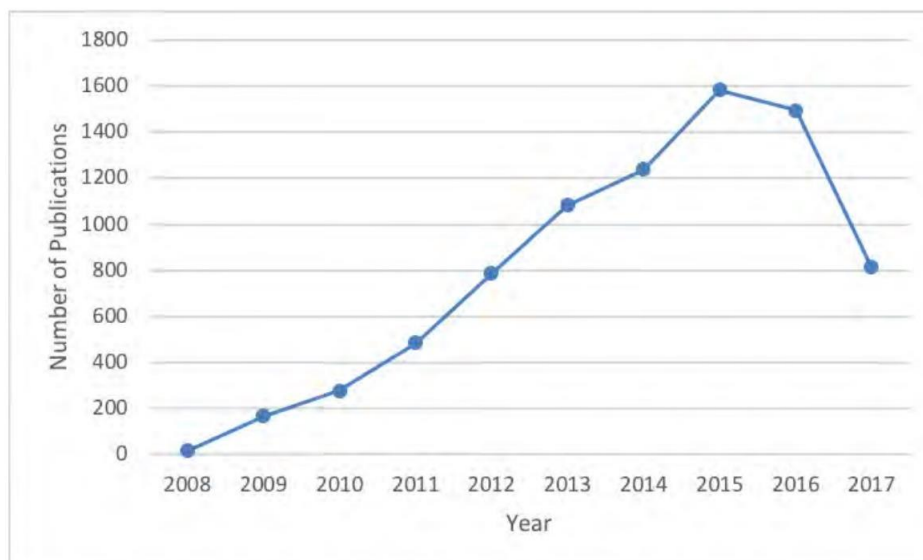
شد. از آن زمان ، حدود 564 مقاله از مجله ها و کنفرانس ها در مورد این موضوع در چهار کتابخانه اصلی دیجیتال (Web of Science , Science Direct , IEEE Xplore و ACM) منتشر شده است ، همانطور که در شکل 4 ارائه شده است. رایانش ابری برای اولین بار در سال 2008 پدیدار شد [25]. این نشان می دهد که انتشارات رایانش مه به طرز چشمگیری افزایش یافته است ، زیرا در طی دو سال پس از معرفی تحقیقات رایانش ابری در سال 2008 هیچ مطالعه ای در این زمینه مشاهده نشده است (شکل 5 را ببینید).



شکل ۳- تعداد مقالات مرتبط با رایانش مه در Web of Science (تا فوریه 2018)



شکل ۴- تعداد نشریات با عنوان "رایانش مه" در چهار کتابخانه اصلی دیجیتال



شکل ۵- مقالاتی با عنوان رایانش ابری در Web of Science منتشر شده است

از نظر ما، بدیهی است که علاقه به تحقیقات رایانش مه به سرعت در حال افزایش است. منابع ساکن به صورت دستگاه های نزدیک به کاربران می توانند در مفهوم رایانش مه استفاده شوند. بنابراین، جهت مشخصی در بازار برای پذیرش و توسعه فن آوری استقرار مه پدید آمده است.

۳- تفاوت بین الگو رایانش مه و ابر: معماری های رایانش مه بر اساس خوشه های مه ساخته شده اند که چندین دستگاه مه برای همکاری با بخش تولید شرکت می کنند. از طرف دیگر، مراکز داده از اجزای اصلی فیزیکی ابرها هستند. به همین دلیل، رایانش ابری هزینه های عملیاتی و مصرف بالایی دارند. طبق توافق، مصرف انرژی و هزینه های بهره برداری در الگوی رایانش مه کم است. مه نزدیکتر به کاربر است، بنابراین فاصله بین کاربران و دستگاههای مه می تواند یک یا چند هاپ باشد، که مورد توافق هو و همکاران است. [8] طبق گفته محمود و همکاران [26]، فاصله بین کاربران تا مه یک یا دو راز است. دوباره، لوان و همکاران [27] استدلال کرد که فاصله باید یک اتصال با اتصال بی سیم باشد. با این حال، همه با فاصله بین کاربران تا ابر، که یک فاصله چند هاپی است، موافقت کردند. به دلیل مسافت، تأخیر ارتباطی برای ابر همیشه در مقایسه با مه زیاد است. ابر یک رویکرد متمرکزتر است در حالی که مه یک رویکرد توزیع شده بیشتر بر اساس تنظیم جغرافیایی است [26].

تعامل در زمان واقعی برای ابر به دلیل تأخیر زیاد امکان پذیر نیست ، اما این مشکل با محاسبه مه به راحتی قابل حل است. از طرف دیگر ، میزان خرابی در مه به دلیل اتصال بی سیم ، عملکرد غیر متمرکز مرد و قطع برق زیاد است [26] ، [28] - [30]. بیشتر دستگاه های موجود در محیط های مه به صورت بی سیم متصل می شوند زیرا ابزارهای هوشمند و دستگاه های دستی در سیستم های مه شرکت خواهند کرد [31]. این دستگاه ها و سایر دستگاه های مدیریت شبکه عمدتاً غیرمتمرکز هستند. وقتی نرم افزار به درستی مدیریت نشود ، این دستگاه ها ممکن است خراب شوند. ممکن است کاربران از نرم افزار مخربی که ممکن است منجر به خرابی دستگاه شود آگاهی نداشته باشند. علاوه بر این ، پردازش مه در موارد دیگر نیز ممکن است خراب شود ، به عنوان مثال ، هر دستگاه مه وظیفه انجام پردازش برنامه خود را دارد. بنابراین ، توقف برنامه کاربردی اینترنت اشیا در دستگاه مه همیشه اولویت دوم را دارد. اگر دستگاه مه با استفاده از خود دستگاه کاملاً مورد استفاده قرار گیرد ، در انجام پردازش مه ناموفق خواهد بود. از این رو ، برنامه ریزی برنامه ها و منابع در مه پیچیده تر است. بعلاوه ، کنترل خرابی در مه به دلیل قطع برق رقابتی است که این فقط به این دلیل است که دستگاهها از باتری کار می کنند. در مجموع ، جدول 2 تفاوت های فنی بین ابر و مه را نشان می دهد.

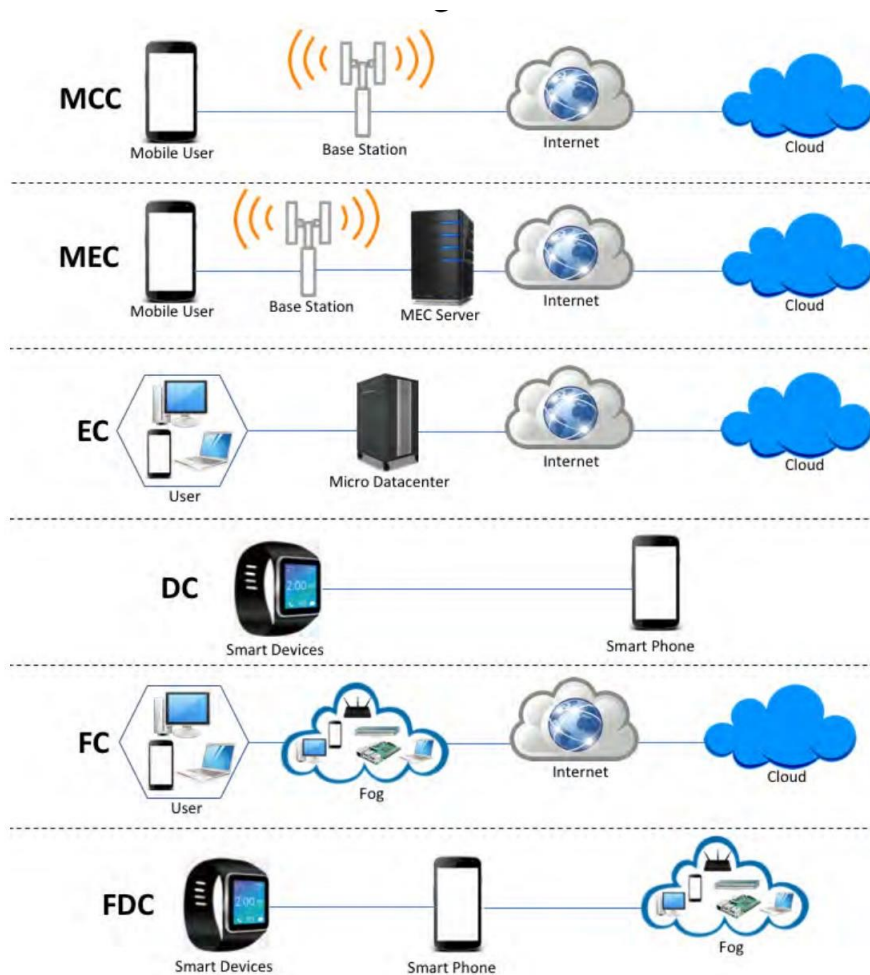
به طور قطع نمی توان گفت مه می تواند جایگزین ابر شود. ما حتی نمی توانیم نتیجه بگیریم که مه بهتر از ابر نیز هست ، هر دو از طریق تحقق دیدگاه ها و الزامات مختلف به طور متفاوتی کمک می کنند.

	Fog	Cloud
Participating Nodes	Constantly dynamic	Variable
Management	Distributed/Centralized	Centralized
Computation device	Any device with computation power	Powerful Server System
Nature of Failure	Highly diverse	Predictable
Connectivity from user	Mostly wireless	High speed (With the combination of wire and wireless)
Internal connectivity	Mostly wireless	Mostly Wired
Power source	Battery/Direct power/Green Energy, such as solar power	Direct power
Power consumption	Low	High
Computation capacity	Low	High
Storage capacity	Low	High
Network latency	Low	High
Node mobility	High	Very low
Number of intermediate hop	One/Few	Multi
Application type	latency-aware	Non latency-aware
Real time application handling	Achievable	Difficult
Computation cost	Low	High
Cooling cost	Very low	High
Space required for deployment	Very little, also possible to install at outdoor on existing infrastructure	Warehouse size building

نمودار ۲- تفاوت های بین مه و ابر

۴- پارادایم ها و فن آوری های مرتبط: رایانش مه از منابع محاسباتی در نزدیکی شبکه های زیرین ، واقع بین دستگاه های سستی ابر و لبه برای استفاده بهتر و سریعتر از پردازش و خدمات کاربردی استفاده می کند [13]. چندین الگوی محاسباتی مشابه علاوه بر رایانش مه مانند رایانش ابری موبایل (MCC) ، محاسبه تلفن همراه لبه (MEC) ، محاسبه لبه ، محاسبه شبنم و محاسبه شبنم مه وجود

دارد. در محاسبات ابری، تمام دستگاه‌های اینترنت اشیا مستقیماً به ابر متصل می‌شوند و محاسبات کاملاً به ابر بستگی دارد. با این حال، همه فن‌آوری‌های مشابه فوق صرفاً به ابر وابسته نیستند، بلکه برای محاسبه به برخی از دستگاه‌های میانی بستگی دارند. بعضی از آنها حتی نیازی به اتصال به ابر ندارند. شکل 6 معماری سطح بالای این فناوری‌ها را نشان می‌دهد.



شکل 6- معماری سطح بالا (MCC) Cloud Computing، Mobile Edge Computing (MEC)، Edge Computing (EC)، Dew Computing (DC)، Fog Computing (FC) و Fog Dew Computing (FDC)

(a) محاسبات ابری متحرک (MCC): اجرای از راه دور خدمات همراه بارگیری نشده با پشتیبانی MCC نزدیک به کاربران نهایی انجام می‌شود [32]، [33]. MCC بیش از این است که رایانش، انرژی و منابع ذخیره سازی محدودیت دستگاه‌های تلفن همراه هوشمند فراهم می‌شود. به طور کلی، یک

سرور ابر سبک (cloudlet) در لبه شبکه [34] قرار داده شده است تا بر این مشکلات غلبه کند. MCC یک فناوری فناوری رایانه همراه است که از طریق اتصال شبکه ناهمگن، قابلیت، تحرک و امکانات ذخیره سازی بدون محدودیت را فراهم می کند. این فناوری همچنین با پیروی از مدل پرداخت به ازای استفاده، منابع رایانش الاستیک یکپارچه را فراهم می کند. همچنین دسترسی به داده ها، برنامه ها و ابر از طریق اینترنت را برای کاربران تلفن همراه فراهم می کند. پیش بینی می شود که این فناوری در آینده در آموزش، توسعه شهری و روستایی، بهداشت و درمان و شبکه های اجتماعی واقع بینانه تر مورد استفاده قرار گیرد [32]. امروزه بسیاری از برنامه های کاربردی رایانش به طور گسترده در دسترس هستند، مانند واقعیت افزوده، دید و گرافیک رایانه، تشخیص گفتار، یادگیری ماشین، برنامه ریزی و تصمیم گیری و برنامه های پردازش زبان طبیعی. با این حال، به سادگی طراحی دستگاه های تلفن همراه قدرتمند، شرایط این برنامه ها را برآورده نمی کند [33] در عوض، برنامه ها به پردازش لبه و همچنین همکاری با ابر برای پردازش پیچیده نیاز دارند. بنابراین، رایانش موبایل تغییرات اساسی در رایانش ابری را می طلبد، به عنوان مثال، یک سطح متوسط با تأخیر کم، مدل های برنامه نویسی برای امکان اجرای از راه دور یکپارچه، خدمات اصلی ابر تلفن همراه مانند خدمات موجود، بهینه سازی زیرساخت ابر برای برنامه های تلفن همراه، خدمات memcache و غیره در [33]. سنسور همگانی رایانش ابری موبایل در یک شبکه قابل اعتماد، پایان به انتها و پهنای باند بالا پیش بینی شده است، که تضمین آن در محیط های سخت دشوار است. یکی از راه حل های این مشکل ریشه دار، ابرهای مبتنی بر ماشین مجازی است که در مکانی نزدیکتر از دستگاه همراه قرار دارند [34].

(b) محاسبات لبه متحرک (MEC): MEC پیشنهاد می دهد مکان مشترک محاسبه و ذخیره منابع در ایستگاه های پایه شبکه های تلفن همراه وجود داشته باشد [35]. MEC یا می تواند به مراکز داده ابری در مکانی از راه دور متصل شود یا قطع شود. از این رو، MEC از استقرار برنامه های سلسله مراتبی دو یا سه لایه همراه با دستگاه های تلفن همراه انتهایی پشتیبانی می کند [36]. در اکوسیستم MEC، دستگاه جدیدی به نام سرور MEC باید در نزدیکی برج های پایه مستقر شود تا قابلیت پردازش و ذخیره سازی در لبه را فراهم کند. چهار شرکت کننده در این الگوی محاسبات شرکت دارند که عبارتند از: کاربران نهایی تلفن همراه، اپراتورهای شبکه، ارائه دهنده زیرساخت اینترنت (InPs) و ارائه دهنده خدمات برنامه. کاربران نهایی موبایل مصرف کننده اصلی سیستم هستند و خدمات خود را از طریق تجهیزات کاربر (UE) درخواست می کنند. اپراتورهای شبکه مدیریت

ایستگاه های پایه ، شبکه هسته تلفن همراه و سرورهای MEC را بر عهده دارند. InP ها اتصال اینترنت و روترها را حفظ می کنند. ارائه دهندگان خدمات برنامه خدمات را در شبکه های تحویل محتوا (CDN) یا در مراکز داده میزبانی می کنند. پردازش درخواست های UE نزدیکترین MEC را جستجو می کند. سرور MEC قادر است به جای ارسال درخواست کاربر به سرویسهای اینترنتی از راه دور ، درخواست کاربر را پردازش کند. در مواردی که پردازش یا تکمیل درخواست در قطع کننده MEC امکان پذیر نباشد. درخواست به CDN یا مراکز داده از راه دور ارسال می شود [35]. طبق گفته کلاس [36] ، MEC تحول ایستگاه های پایه تلفن همراه است. این یک پیشرفت طبیعی است. این استقرار مشترک شبکه های مخابراتی و IT است. این الگوی محاسبات ، بخشهای تجاری و خدمات جدید عمودی را برای کاربران نهایی فردی و مشتریهای سازمانی امکان پذیر می کند. خدمات مختلفی می توانند از طریق این الگوی کامپوننت ارائه شوند از جمله اینترنت اشیا ، خدمات مکان ، واقعیت افزوده ، سرویس ذخیره سازی ، تجزیه و تحلیل ویدئو و محرومیت از محتوای محلی. این می تواند دسترسی کم زمان تأخیر به محتوای محلی یا با ذخیره محتوای موجود در سرور MEC را فراهم کند. با این حال ، محدودیت اصلی این سیستم نصب سرور MEC است که به طور خاص به خدمات MEC اختصاص یافته است. مقیاس گذاری مسئله مهم دیگری است که با افزایش تقاضای منابع در طول زمان افزایش می یابد.

(c) رایانش لبه: دستگاه های لبه یا سرورهای لبه روابط محاسباتی را در رایانش لبه فراهم می کنند. به طور کلی ، رایانش لبه به طور خود به خود با هیچ نوع سرویس مبتنی بر ابر ارتباط برقرار نمی کند و بیشتر در سمت دستگاه IoT متمرکز می شود [23]. یک مطالعه لبه را به عنوان هر شبکه یا منبع محاسباتی نزدیک مسیر بین مراکز داده ابر و منابع داده تعریف کرد [23]. هر دستگاه هوشمند یا حسگری می تواند منابع داده داشته باشد اما لبه آن متفاوت است. به عنوان مثال ، یک cloudlet و یک مرکز داده خرد لبه برنامه تلفن همراه و cloud است ، در حالی که دروازه IoT مرز بین حسگرهای اینترنت اشیا و ابر است. به همین ترتیب ، اگر برنامه Cloud روی تلفن هوشمند در حال اجرا است ، تلفن هوشمند لبه برنامه و Cloud است [37]. انگیزه اصلی محاسبه لبه این است که محاسبه باید در مکانی نزدیکتر به منابع داده انجام شود. در مفهوم محاسبه لبه ، همه چیز نه تنها جمع آوری داده است بلکه با شرکت در پردازش ، داده نیز تولید می کند. دستگاه های لبه علاوه بر درخواست خدمات و محتوا ، می توانند وظیفه محاسباتی را از طریق cloud نیز انجام دهند. ذخیره داده ، محاسبه بارگیری ، پردازش و ذخیره سازی اطلاعات توسط گره لبه ای انجام می شود. دستگاه

لبه همچنین قادر به تقسیم درخواست ها و ارائه خدمات از طرف ابر به کاربران است. در چنین حالاتی ، دستگاههای لبه باید به خوبی طراحی شوند تا شرایط حریم خصوصی ، اقدامات قابل اطمینان و نگرانی های امنیتی را برآورده کنند [23].

(d) محاسبات شبهم (DC): در سلسله مراتب رایانش فعلی ، رایانش شبهم [38] در سطح زمین محیط ابر و مه قرار گرفته است [39]. DC فراتر از مفهوم سرویس ، ذخیره سازی و شبکه ، به یک زیر پلتفرم تبدیل می شود ، که مبتنی بر یک مفهوم ریز سرویس است که سلسله مراتب محاسباتی آن به صورت عمودی توزیع شده است [39]. رویکرد DC منابعی مانند حسگرها ، تبلت ها و تلفن های هوشمند را که به طور یکپارچه به یک شبکه متصل هستند ، تسهیل می کند. به همین دلیل ، DC طیف گسترده ای از فن آوری های شبکه مبتنی بر ad-hoc را پوشش می دهد [39]. اسکالا و همکاران [39] استدلال کرد که DC در مقایسه با رایانش مه در زندگی روزمره بسیار مفیدتر است. برنامه های مبتنی بر اینترنت اشیا مه ، که به تأخیر کمتر و قابلیت زمان واقعی و پیکربندی شبکه پویا نیاز دارند در حالی که DC مفهوم ریز سرویس است و بنابراین به هیچ دستگاه ، سرور یا ابری متمرکز وابسته نیست. آنها مثالی را ارائه می دهند که در آن DC می تواند در یک سیستم کنترل هوشمند ترافیک ادغام شود ، جایی که واحدهای جمع آوری و پردازش داده ها در بین سیگنال های ترافیکی قرار خواهند گرفت. این واحدها یک نمای کلی از شرایط فعلی ترافیک ایجاد می کنند. به این ترتیب قبل از ورود به ازدحام سنگین ، به اتومبیل با سوخت کم اطلاع داده می شود یا قبل از نزدیک شدن به شلوغی ، از اتومبیل هیبریدی از تعویض سوخت متعارف مطلع می شود. در نتیجه ، اتومبیل های با سوخت کمتر از موقعیت های ناخواسته مسترد می شوند و اتومبیل های هیبریدی می توانند تراکم دود آگروز را به میزان قابل توجهی کاهش دهند. اگرچه این مفهوم مبتنی بر ریز سرویس است ، اما پردازش در دستگاه های Fog به پایان می رسد. در مفهوم رایانش مه ، مهم نیست که برنامه ها باید به ابر وابسته باشند یا نیاز به ذخیره نتایج در ابر داشته باشند. از طرف دیگر ، اگر چنین اطلاعات پردازش ترافیکی ذخیره می شد ، به تصمیم گیری استراتژیک برای بهبود مدیریت ترافیک کمک می کند. محاسبه شبهم یک منطقه تحقیقاتی نوظهور است و هدف آن استفاده از پتانسیل کامل ابر و منابع محلی است [40].

(e) رایانش مه - شبهم: در معماری رایانش مه - شبهم ، دستگاه های اینترنت اشیا هنگام اتصال به سرور جامع نیازی به اتصال اینترنت فعال ندارند. سرور اصلی با ابر ارتباط برقرار می کند و وظیفه ارائه خدمات به دستگاه های اینترنت اشیا را بر عهده دارد [41]. رایانش ابری همیشه نیاز به اتصال

به اینترنت دارد که اصلی ترین اشکال ابر است. در حالی که ابر بدون اتصال به اینترنت قادر به سرویس دهی به کاربران نیست ، رایانش مه – شبنم خدمات آفلاین را بدون اتصال به اینترنت تسهیل می کند. با این حال، چند استثنا وجود دارد. به عنوان مثال ، برنامه پیمایش ، Waze ، به کاربران امکان می دهد تا به صورت آفلاین پیمایش کنند. این ویژگی اخیراً به Google Maps نیز اضافه شده است. در این حالت ، یک فایل اطلاعات نقشه برای یک منطقه خاص در دستگاه کاربر بارگیری می شود و به کاربران امکان می دهد تا در حالت آفلاین پیمایش کنند. مثال دیگر Google Drive و Dropbox است که در آن کاربران می توانند پرونده ها و پوشه ها را در حالت آفلاین حذف ، ایجاد و به روز کنند و پس از اتصال دستگاه به اینترنت ، همگام سازی می شوند. با این حال ، این سرویس ها کاملاً آفلاین نیستند – ممکن است ما مستقیماً به اینترنت اتصال پیدا نکنیم اما نمی توانیم ارتباط اینترنت را کاملاً نادیده بگیریم. اوضاع وقتی پیچیده تر می شود که یک کاربر تنها از چندین دستگاه آفلاین در کنار پیچیدگی های ایجاد شده در یک محیط چند کاربر استفاده کند. چنین شرایطی را می توان با کمک رایانش مه – شبنم کاهش داد. به طور خلاصه ، در الگوی رایانش مه ، دستگاه های اینترنت اشیا از طریق دستگاه های مه به ابر متصل می شوند. دستگاه های مه از طریق شبکه هسته ای به ابر متصل می شوند. رایانش مه ترکیبی از MEC و MCC [7] است اما هدف اصلی همه الگوهای مربوط به مه انجام پردازش در لبه است. این الگوهای مرتبط بر اساس اتصال اینترنت و ابر با یکدیگر متفاوت هستند. همچنین ، میزان پردازشی که باید در لبه انجام شود بر اساس نیاز خدمات متفاوت است. بعلاوه ، نوع دستگاههایی که برای محاسبه و ذخیره سازی استفاده خواهند شد نیز مسئله دیگری است. به طور خلاصه ، جدول 3 ویژگی های الگوی محاسبات مربوط به بالا را به همراه الگو رایانش مه نشان می دهد.

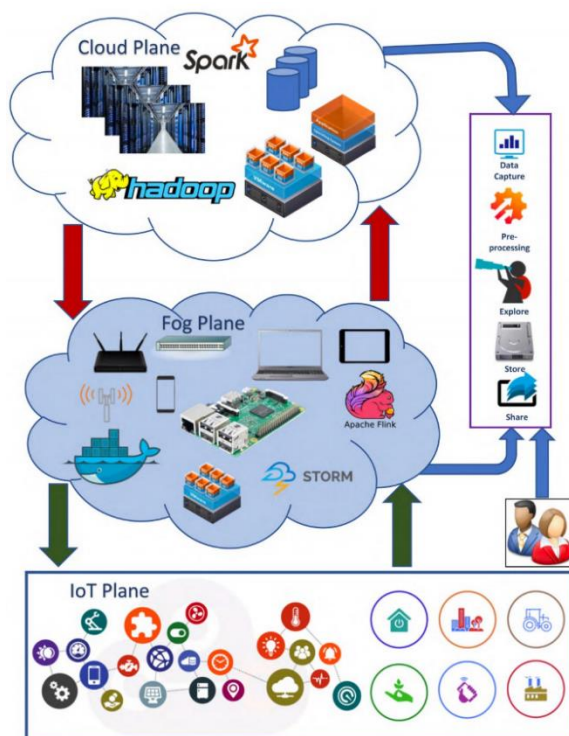
۵- معماری رایانش مه : برای پذیرش و گسترش در بازار ، رایانش مه باید معماری استاندارد داشته باشد. هیچ معماری استاندارد موجود تا به امروز وجود ندارد. با این حال ، بسیاری از کارهای پژوهشی از قبل معماری رایانش مه را ارسال کرده اند. در این بخش ، ما ابتدا به بحث در مورد معماری سطح بالا رایانش مه می پردازیم. در ضمن ، ما برخی از معماری های پیشنهادی را برای محاسبه مه به طور خلاصه بیان می کنیم. سرانجام ، ما یک معماری دقیق برای رایانش مه با توصیف جامع هر یک از مولفه های معماری ارائه می دهیم.

A. معماری سطح بالای رایانه های مه

همانطور که در شکل 7 نشان داده شده است ، در معماری سطح بالا ، الگوی رایانش مه دارای سه لایه مختلف است. مهمترین لایه ، لایه مه است. این لایه از کلیه دستگاههای محاسبه میانی تشکیل شده است. می توان از فناوری های سنتی مجازی سازی در این صفحه ، مشابه ابر استفاده کرد. با این حال ، با توجه به در دسترس بودن منابع ، استفاده از مجازی سازی مبتنی بر کانتینر مناسب تر است. این لایه داده های تولید شده توسط سنسور را از لایه IoT جمع می کند و پس از پردازش درخواست مربوط به عمل را ارسال می کند. اگرچه به نظر می رسد با پردازش داده های تولید شده در سطح لبه ، مشکل داده های بزرگ حل می شود ، اما میلیاردها دستگاه مشکل داده های بزرگ را ایجاد می کنند. در حقیقت ، می توان از پردازش داده های بزرگ در مقیاس کوچک و متوسط در این سطح استفاده کرد. کارهای تحقیقاتی زیادی برای پردازش داده های بزرگ در صفحه مه انجام شده است [47] - [53]. پایین ترین لایه هواپیمای اینترنت اشیا است که از همه دستگاه های متصل تشکیل شده است. دستگاه های موجود در این صفحه فرآیند سنجش و تحریک را انجام می دهند. برای کاربردهای حساس به زمان ، پردازش باید در صفحه مه به طور انحصاری انجام شود در حالی که ابر می تواند پردازش دیگری را انجام دهد که به زمان حساس نیست. با این حال ، لایه مه آنچه را که باید به ابر ارسال شود و آنچه را که نباید ، مدیریت خواهد کرد. کاربران می توانند بر اساس درخواست خود از هر دو مه و ابر خدمات دریافت کنند. با این حال ، هواپیمای ابری پردازش و ذخیره سازی پیچیده را مدیریت خواهد کرد.

Computing Paradigm & Applications	MCC	MEC	EC	DC	Fog-DC	Fog
	Offload mobile applications to the computation unit closer to the user					
	Mobile apps	Mobile apps	IoT related apps	Smart Devices (Fitbit, health monitoring)	Smart Devices (Fitbit, health monitoring)	IoT apps, Mobile apps, Video streaming, smart grid, smart transportation system, big data processing, stream processing
Connection to the Cloud	Yes	Yes or No	Yes or No	No	No	Yes
Types of Users	Mobile	Mobile	Mobile / Stationary	Smart sensor based devices	Smart sensor based devices	Mobile / Stationary
Virtualization Technology	Hypervisor / Container	Hypervisor / Container	Hypervisor / Container	Container	Container	Hypervisor / Container
Main Computation Element	Base station server	MEC server	Micro Data Center	Smartphone	Smartphone	Any device with the capabilities of computation, storage, memory and network adapter
Example of Commercial Prototype	Akamai [42]	-	Ec-IoT (Huawei) [43]	-	-	IOx (Cisco) [44]
Example of R&D Prototype	-	Hyrax [45], Saguna Open-RAN [46]	-	-	-	-

جدول ۳- خلاصه ای از فن آوری های مشابه مه



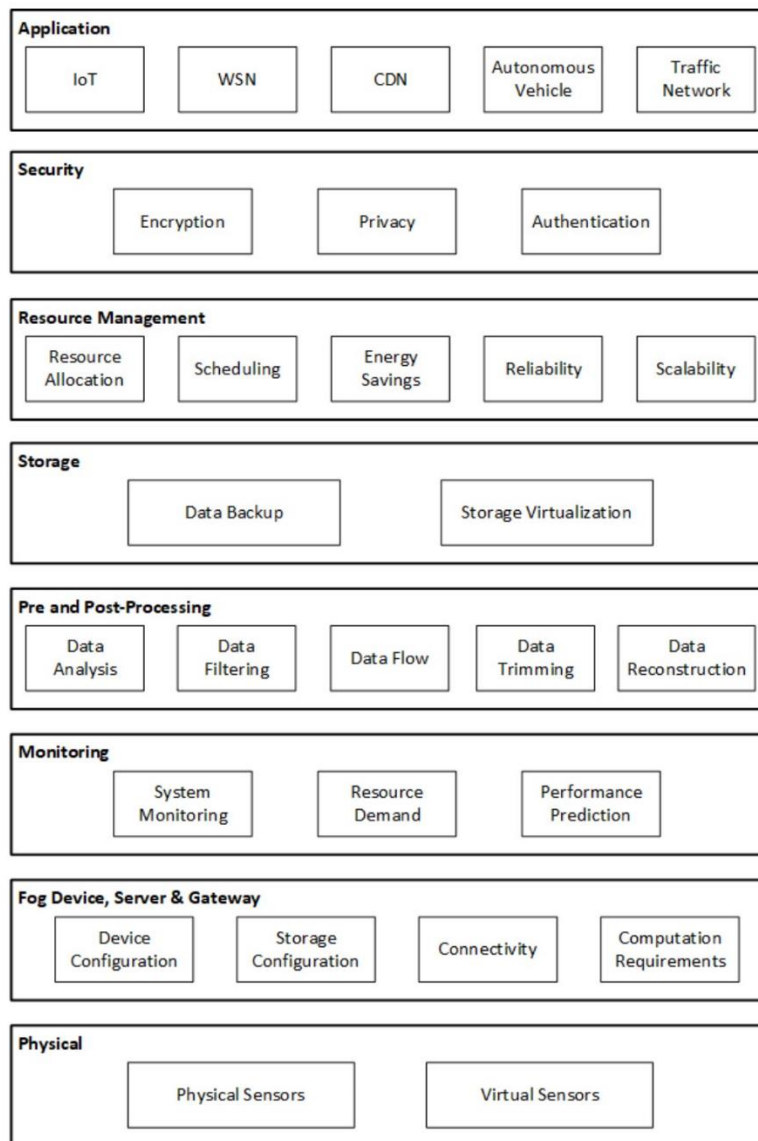
شکل ۷- رایانش مه سطح بالا

B. معماری های مختلف پیشنهادی برای رایانش مه

نمایش لایه ای بهترین راه برای نمایش معماری مه است. کارهای زیادی برای تعیین کمیت مفهوم لایه ای معماری مه انجام شده است [4]، [27]، [54] - [58]. از بررسی ما، دریافتیم که محققان سه [27]، [56] - [58]، چهار [55]، پنج [4] و شش [54] لایه را در معماری مه پیشنهاد کرده اند. هرکسی توجیهات خاص خود را برای ادعاهای خود دارد. اگر صفحه کاربر را نادیده بگیریم، بدیهی است که معماری مه می تواند به عنوان سه سطح متفاوت از سطح بالا تعریف شود. همانطور که به سطح نوع اجرای بیشتری می رویم، تعداد لایه ها در معماری متفاوت است و در سطح محاسباتی مه به پنج سطح [4] و شش [54] منجر می شود. Aazam و هو [54] شش لایه مختلف را براساس وظایف خاص ارائه دادند. از طرف دیگر، دستجردی و همکاران. [4] پنج لایه مختلف را براساس دیدگاه شبکه تعریف کرد. سایر معماری های سطح بالا در رایانش مه نیز توسط محققان مختلف از جمله معماری سلسله مراتبی مه [59]، [60]، معماری OpenFog [61]، معماری شبکه مه [62]، معماری مه برای اینترنت انرژی [9]، معماری رایانش مه مبتنی بر سیستم عصبی [63] و معماری IFClOT [64]. پس از مرور ادبیات بیان شده در بالا، مولفه های معماری رایانش مه را تعریف می کنیم که در زیر بخش ارائه شده است.

C. اجزای معماری رایانش مه

معماری رایانش مه از چندین لایه تشکیل شده است. هر لایه و اجزای آن در شکل 8 نشان داده شده است. در این بخش، ما در مورد اجزای مختلف معماری رایانش مه بحث می کنیم. اجزا براساس عملکردشان به چند گروه تقسیم می شوند که به عنوان لایه تعریف می شوند. این ویژگی ها دستگاه های اینترنت اشیا را قادر می سازد تا با دستگاه های مختلف مه، سرورها، دروازه ها و ابر ارتباط برقرار کنند. در زیر توضیحات مفصلی از هر لایه آورده شده است، جایی که یک مورد استفاده از حمل و نقل هوشمند در توضیح در نظر گرفته شده است.



شکل ۸- مولفه های معماری رایانش مه

1) لایه فیزیکی:

منبع اصلی داده برای رایانش مه ، اشکال مختلف داده است که توسط سنسورها ساطع می شود [57]. این داده ها می توانند از دستگاه های هوشمند ، سنسورهای دما ، سنسورهای رطوبت ، خانه های هوشمند ، سیستم نظارت بر دوربین مداربسته ، سیستم نظارت بر خودرو ، وسایل نقلیه خودران و غیره تولید شوند. به عنوان مثال ، اگر ما می خواهیم یک سیستم هوشمند کنترل و نظارت بر ترافیک را اجرا کنیم ، باید شرایط مختلف ترافیک به روز شده را از حسگرهای مختلف ، دستگاه های کنار جاده و دوربین ها بدست آوریم ، که به مدیریت سیگنال های راهنمایی کمک می کند.

همچنین لازم است با جمع آوری داده ها از حسگرهای مختلف GPS ، تقاضای آینده برای ترافیک پیش بینی شود. علاوه بر حسگرهای فیزیکی ، نقش سنسورهای مجازی نیز مهم است [54] ، اگر یک حادثه جاده ای رخ دهد ، نمی توان فقط با استفاده از یک سنسور تصمیم گرفت که آیا جاده باید مسدود شود یا ترافیک ادامه یابد. جاده ممکن است دارای یک یا چند خط باشد یک خط ممکن است تحت تأثیر این اتفاق قرار گیرد در حالی که یک خط دیگر می تواند جریان ترافیک را ادامه دهد ، اما به دلیل این اتفاق از ظرفیت مدیریت ترافیک کاسته می شود. در این حالت ، یک حسگر مجازی ممکن است به تصمیم گیری فوری در مورد شرایط جاده ، مالتی پلکسینگ ترافیک و تغییر مسیر ترافیک کمک کند. از این رو ، لایه فیزیکی متشکل از حسگرهای فیزیکی و مجازی است ، که در آن هر دستگاه تولید داده می تواند در هر یک از این گروه ها قرار گیرد.

2) دستگاه مه، سرور، لایه دروازه:

Fog device ، Fog server یا Fog gateway می تواند یک دستگاه جدا یا یک دستگاه IoT باشد [57] ، [59] ، [62]. با این حال ، بدیهی است که سرور مه باید پیکربندی بالاتری نسبت به دستگاه مه و دروازه داشته باشد ، زیرا چندین دستگاه مه را مدیریت می کند. عوامل مختلفی در این امر دخیل هستند تا سرور مه بتواند اجرا شود. اینها شامل نقش آن ، پیکربندی سخت افزار ، اتصال ، تعداد دستگاههایی است که می تواند مدیریت کند و غیره. اینکه سرور مه مشخص باشد یا بخشی از یک دستگاه اینترنت اشیا باشد ، با توجه به نقش آن مشخص می شود. گروهی از حسگرهای فیزیکی و مجازی به دستگاه مه متصل می شوند. به همین ترتیب ، گروهی از دستگاه های مه به یک سرور مه متصل می شوند. در این زمینه ، سرور مه باید در مقایسه با دستگاه مه از ظرفیت پردازش و ذخیره سازی بالاتری برخوردار باشد. یک خوشه خاص از دستگاه های مه ، که به همان سرور متصل هستند ، می توانند در صورت لزوم با یکدیگر ارتباط برقرار کنند. در مورد استفاده از حمل و نقل هوشمند ، برخی از پردازش های برنامه ممکن است به خوشه های مه دیگر بستگی داشته باشد. به عنوان مثال ، اگر برنامه ای برای یافتن مسیری با مصرف سوخت مناسب نیاز داشته باشد ، ممکن است به اطلاعات مربوط به خوشه های حسگر دیگر یا خوشه های دستگاه مه نیاز داشته باشد. برای رسیدن به یک تصمیم مناسب ، پردازش باید در چندین دستگاه مه و سرورها انجام شود. سرور مه و لایه دستگاه وظیفه مدیریت و حفظ اطلاعات مربوط به پیکربندی سخت افزار ، پیکربندی ذخیره سازی و اتصال دستگاه و سرورها را بر عهده دارند. این لایه همچنین الزامات محاسبه درخواست شده توسط

برنامه های مختلف را مدیریت می کند. الزامات محاسبه به جریان داده و تعداد کل دستگاه های اینترنت اشیا متصل به دستگاه مه و همچنین تعداد کل دستگاه های مه متصل به سرور مه بستگی دارد. ارتباط بین چندین سرور مه توسط این لایه حفظ می شود. به عنوان مثال ، یک روتر سری 800 پشتیبانی شده از ICox Cisco می تواند به عنوان دستگاه مه و دستگاه های سرویس داده داده Cisco Fog می توانند به عنوان سرور مه [65] ، [66] استفاده شوند.

(3) لایه نظارت:

لایه مانیتورینگ همیشه عملکرد و منابع سیستم [54] ، خدمات و پاسخها را ردیابی می کند. مولفه های نظارت بر سیستم به انتخاب منابع مناسب در حین کار کمک می کنند. برنامه های مختلف در حالات سیستم حمل و نقل هوشمند اجرا می شوند. بنابراین ، بدیهی است که شرایطی ممکن است بوجود آید که در دسترس بودن منابع برای محاسبه یا ذخیره سازی در دستگاه مه منفی باشد. مورد مشابهی می تواند برای سرور مه رخ دهد. برای مقابله با این نوع شرایط ، دستگاه مه و سرورها از سایر همتایان کمک خواهند گرفت. بنابراین ، مولفه نظارت بر سیستم به تصمیم گیری موثر چنین مواردی کمک می کند. مولفه تقاضای منابع تقاضای فعلی منابع را کنترل می کند و می تواند تقاضای آینده برای منابع را بر اساس استفاده از منابع فعلی و فعالیت های کاربر پیش بینی کند. به این ترتیب ، سیستم قادر خواهد بود با هر موقعیت نامناسبی که ممکن است قطع منابع رخ دهد ، کنار بیاید. ناظران پیش بینی عملکرد می توانند عملکرد رایانش مه را بر اساس بارگذاری سیستم و در دسترس بودن منابع پیش بینی کنند. این مولفه برای حفظ الزامات مناسب کیفیت کیفیت در موافقت نامه های سطح خدمات مورد نیاز است. اگر نقض SLA مرتباً رخ دهد ، به دلیل مجازات شدن ، هزینه سیستم برای ارائه دهنده افزایش می یابد. اگرچه پیش بینی عملکرد نمی تواند این مسئله را به طور کامل از بین ببرد ، اما با پیش بینی عملکرد و استفاده از سیستم می تواند نقض کلی SLA را به حداقل برساند.

(4) لایه قبل و بعد از پردازش:

این لایه شامل چندین مولفه است که بطور خاص در تجزیه و تحلیل داده های اساسی و پیشرفته کار می کنند. در این سطح ، داده های بدست آمده تجزیه و تحلیل و فیلتر می شوند و در صورت لزوم اصلاح و بازسازی داده ها نیز انجام می شود. بعد از توقف اطلاعات ، مولفه جریان داده تصمیم می گیرد که آیا داده ها باید به صورت محلی ذخیره شوند یا برای ذخیره سازی طولانی مدت باید به ابر

ارسال شوند [59]. چالش اصلی در رایانش مه پردازش داده ها در لبه و به حداقل رساندن حجم داده هایی است که باید ذخیره شوند. از این پدیده به عنوان پردازش جریان یاد می شود. در حالت استفاده از سیستم حمل و نقل هوشمند ، داده ها از حسگرهای زیادی تولید می شوند. این داده های تولید شده در زمان واقعی مورد تجزیه و تحلیل و فیلتر قرار می گیرند تا از داده های تولید شده اطلاعات کسب کنند. همه داده های تولید شده ممکن است هیچ استفاده ای نداشته باشند. در برخی موارد ، ذخیره سازی تمام داده های تولید شده حتی ایده خوبی نخواهد بود. به عنوان مثال ، اگر داده ها از یک حسگر در هر ثانیه تولید شوند ، ممکن است بسته به نیاز برنامه ، مقدار متوسط داده در عرض یک دقیقه یا یک ساعت ذخیره شود. داده ها از این طریق می توانند اصلاح و مقدار زیادی فضای ذخیره سازی ذخیره شود. در حالت دیگر ، اگر تفاوت در مقادیر داده در برخی از دوره های زمانی زیاد نباشد ، اما ممکن است عملکرد را تحت تأثیر قرار دهد ، تعداد کمتر خواندن در یک دقیقه قابل انجام است. به این ترتیب فیلتر کردن مقدار زیادی از داده های تولید شده امکان پذیر خواهد بود. اگرچه دقت ممکن است 100٪ نباشد ، اما ممکن است الزامات برنامه تا حدودی برآورده شود. بازسازی داده ها یکی از اجزای تشکیل دهنده این لایه است. این مازول از داده های معیوب و ناقص تولید شده توسط حسگرها مراقبت می کند. به همین ترتیب ، اگر یک یا چند سنسور در حین کار خراب شوند ، این مولفه داده ها را براساس الگوی تولید داده بازسازی می کند تا از وقفه یا خرابی برنامه دیگر جلوگیری کند.

(5) لایه ذخیره سازی:

ماژول ذخیره سازی وظیفه ذخیره داده ها از طریق مجازی سازی ذخیره سازی را بر عهده دارد. مولفه پشتیبان گیری از داده ها ، در دسترس بودن داده ها را تضمین می کند و از بین رفتن داده ها را کاهش می دهد. در مفهوم مجازی سازی ذخیره سازی ، مجموعه ای از دستگاه های ذخیره سازی متصل به شبکه به عنوان یک دستگاه ذخیره سازی واحد عمل می کنند که مدیریت و نگهداری آن آسان تر است. یکی از مزایای کلیدی مجازی سازی ذخیره سازی ، ارائه کارایی عملکردی در کلاس سازمانی با استفاده از فضای ذخیره سازی ارزان تر یا سخت افزار کالا است. بنابراین ، لایه ذخیره سازی مجازی سازی ذخیره سازی را تسهیل می کند تا پیچیدگی سیستم ذخیره سازی را به حداقل برساند. در یک سیستم ، ذخیره سازی ممکن است در هر نقطه از عملکرد سیستم از کار بیفتد [67]. بنابراین ، تهیه پشتیبان از داده های مهم برای کاهش هرگونه موقعیت ناخواسته بسیار مهم است.

ماژول پشتیبان گیری از داده ها در این لایه از طرح های پشتیبان گیری دوره ای یا سفارشی داده مراقبت می کند.

(6) لایه مدیریت منابع:

مولفه قابلیت اطمینان قابلیت اطمینان برنامه ریزی برنامه و تخصیص منابع را حفظ می کند. مقیاس پذیری مقیاس پذیری منابع مه را در ساعت های اوج که تقاضای منابع زیاد است تضمین می کند. ابر با مقیاس پذیری افق سرو کار دارد در حالی که رایانش مه قصد دارد مقیاس پذیری هوری زونتال و عمودی را ارائه دهد [9]. منابع سیستم توزیع شده زیادی برای شبکه ، پردازش و ذخیره سازی وجود دارد. این یک مسئله حیاتی برای منابع توزیع شده است که از پردازش برنامه استفاده می کنند. بنابراین ، تخصیص ، تخصیص منابع و تخصیص مجدد منابع اتفاق می افتد که در آن مولفه تخصیص منابع مسائل مربوط به تخصیص منابع را مدیریت و نگهداری می کند. مسئله حیاتی دیگر این است که بسیاری از برنامه ها همزمان در محیط رایانش مه اجرا می شوند. از این رو ، برنامه ریزی مناسب این برنامه ها مورد نیاز است. مولفه برنامه ریزی برنامه بر اساس اهداف مختلفی از این برنامه ها مراقبت می کند. این لایه همچنین دارای مولفه های صرفه جویی در انرژی است که منابع را به روشی با مصرف انرژی مدیریت می کنند. بهره وری انرژی همچنین بر محیط تأثیر مثبت می گذارد و به حداقل رساندن هزینه عملیاتی کمک می کند. اجزای قابلیت اطمینان نیاز به قابلیت اطمینان یک سیستم را بر اساس معیارهای مختلف قابلیت اطمینان و معیارها برطرف می کنند. رایانش مه یک سیستم پیچیده است که باید از تمام دستگاه های اینترنت اشیا ، دستگاه های مه و ابر مراقبت کند. بنابراین ، یک دستگاه یا اتصال ممکن است در هر سطحی خراب شود ، بنابراین مدیریت قابلیت اطمینان مسئله مهمی است.

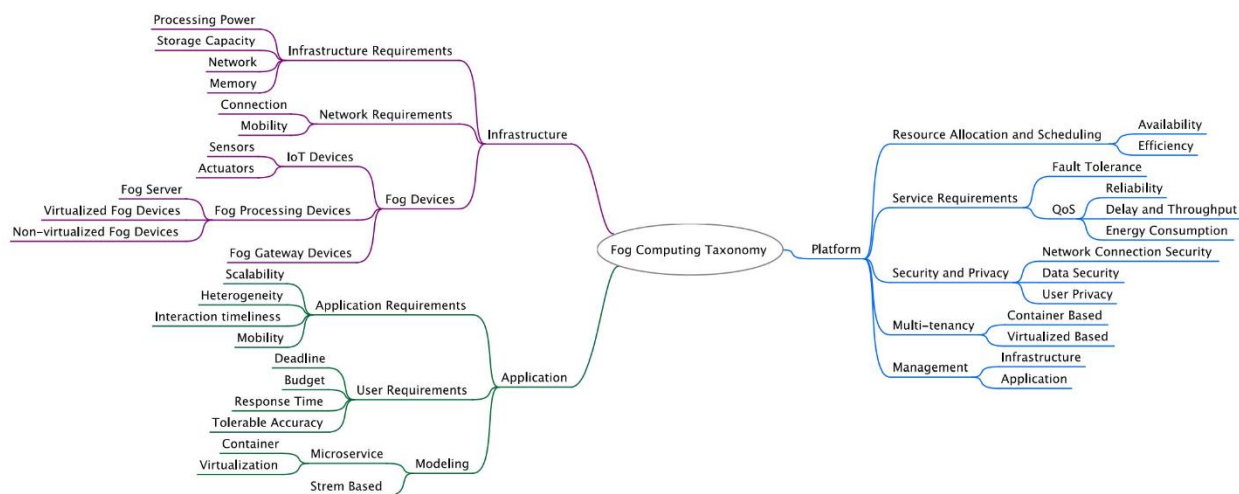
(7) لایه امنیتی

مسائل مربوط به همه امنیت مانند رمزگذاری ارتباطات و ذخیره سازی اطلاعات امن توسط مولفه های موجود در این لایه حفظ می شود که همچنین باعث حفظ حریم خصوصی کاربران مه می شود. رایانش مه به عنوان نوعی محاسبات سودمند مانند رایانش ابری در نظر گرفته شده است. با این حال ، در مفهوم رایانش ابری ، کاربر برای خدمات به ابر متصل می شود ، اما در مفهوم رایانش مه ، کاربر به زیرساخت مه برای سرویس ها متصل می شود در حالی که میان افزار مه ارتباطات را با ابر

مدیریت و حفظ می کند. از این رو ، کاربری که قصد اتصال به یک سرویس را دارد باید از طرف ارائه دهنده مجاز باشد. بنابراین ، مولفه تأیید اعتبار در لایه امنیتی درخواست های احراز هویت از کاربران را پردازش می کند ، بنابراین آنها می توانند به محیط سرویس رایانش مه متصل شوند [27]. برای حفظ امنیت ، حفظ رمزگذاری بین ارتباطات بسیار مهم است ، به طوری که نقض امنیت توسط افراد خارجی رخ ندهد. مولفه رمزگذاری تمام ارتباطات از و به دستگاه های اینترنت اشیا و به ابر را رمزگذاری می کند. اجزای رایانش مه بیشتر از طریق اتصال بی سیم متصل می شوند ، بنابراین نگرانی های امنیتی بسیار مهم است. برخی از خدمات در یک شهر هوشمند یا حریم خصوصی خانه های هوشمند نیز به دلیل درگیر شدن داده های مربوط به کاربر در این نوع سیستم ها مسئله ساز است. الگوی رایانش مه نباید اطلاعات کاربر را بدون رضایت آنها فاش کند. در عصر حاضر ، اکثر کاربران به طور معمول سیاست امنیتی ارائه دهنده را بدون خواندن آن می پذیرند. بنابراین ، برای چنین خدماتی که شامل اطلاعات مهم مربوط به کاربر است ، باید توجه ویژه ای به حریم خصوصی صورت گیرد.

(8) سطح برنامه

اگرچه مه برای سرویس دهی به برنامه های اینترنت اشیا ساخته شد [58] ، چندین برنامه دیگر مبتنی بر **Wire Sensor Network (WSN)** و **CDN** نیز از رایانش مه پشتیبانی می کنند. هر برنامه ای که دارای ویژگی های شخصیت تاخیر باشد ، می تواند از رایانش بهره برد. این شامل هر نوع سرویس مبتنی بر تسهیلات است که می تواند با ارائه کیفیت خدمات بهتر و مقرون به صرفه بودن در رایانش مه قرار گیرد. به عنوان مثال ، برنامه های کاربردی مبتنی بر واقعیت افزوده باید رایانش مه را به دلیل ماهیت آن استفاده کنند. روشن است که واقعیت افزوده دنیای مدرن را در آینده نزدیک متحول خواهد کرد. نیازهای پردازش در زمان واقعی برای برنامه های واقعیت افزوده را می توان با محاسبه مه ، که می تواند بهبود مستمر خدمات مرتبط با واقعیت افزوده را حفظ کند ، برطرف کند.



شکل ۹- طبقه بندی رایانش براساس نیازها ، زیرساخت ، سیستم عامل و برنامه ها

۶- طبقه بندی رایانش مه : طبقه بندی رایانش مه در شکل ۹ ارائه شده است. این طبقه بندی با در نظر گرفتن ادبیات موجود و دیدگاه کلی در مورد رایانش مه گرفته شده است. طبقه بندی پیشنهادی بر چشم انداز مورد نیاز برای زیرساخت ها ، سیستم عامل و برنامه ها متمرکز است. در مرحله اول ، با در نظر گرفتن زیرساخت ها ، زیرساخت ها و نیازهای شبکه و انواع دستگاه ها را در یک محیط رایانش مه شناسایی می کنیم. ثانیاً ، برای تخصیص و برنامه ریزی منابع پلتفرم ، امنیت و حریم خصوصی ، نیازهای خدمات ، مدیریت و تعدد قابلیت تعیین شد. سرانجام ، ما به برنامه های کاربردی ، نیازهای کاربر و مدل سازی برنامه های کاربردی برای رایانش مه نیاز داریم. این طبقه بندی به جامعه تحقیقاتی و بنگاه های اقتصادی کمک می کند تا درک و بینش بهتری در مورد استقرار در دنیا در مورد الزامات ، معماری و دستگاههای رایانش مه به دست آورند. شکل ۹ طبقه بندی رایانش مه را نشان می دهد. شرح مفصلی از هر شاخه از طبقه بندی در این بخش ارائه شده است.

الف - زیرساخت

نیازهای زیرساخت رایانش مه به شبکه ، دستگاه ها و نیازهای آنها بستگی دارد. تمام دستگاههای مه ، دستگاه های شبکه و دروازه های موجود در محیط مه که در محاسبات شرکت می کنند نیز بخشی از زیرساخت های مه هستند. زیرساخت نشان دهنده اجزای فیزیکی محیط مه است.

1) الزامات زیرساخت

بسیاری از دستگاه های ریز متصل ، عناصر اصلی یک محیط رایانش مه هستند. این دستگاه ها در همه جا قرار دارند و به اتصال همه چیز در اطراف ما کمک می کنند. تخمین زده می شود که تا سال 2020 دنیا 50 میلیارد دستگاه دستی را ببیند. در کنار این دستگاه ها ، تعداد زیادی حسگر و محرک نیز نصب خواهند شد. بنابراین ، برای پشتیبانی از این محیط گسترده محاسباتی ، یک تسهیلات ساختاری زیربنایی مناسب لازم است [20]. نمونه ای از چگونگی افزایش تعداد حسگرها روز به روز در گزارش اکونومیست با عنوان "کسب و کار تقویت شده" آورده شده است ، که توصیف سنسورهای کاشت در گوش گاو است که می تواند به نظارت بر فعالیت ، سلامت و حرکات آنها کمک کند. این می تواند به افزایش قابلیت کلی سود کمک کند. کاشت سنسورهای چسبانده شده در یک گاو در سال حدود 200 مگابایت اطلاعات تولید می کند. در یک آزمایش دیگر ، با فناوری سنسور ، رولزرویس قادر به پیش بینی احتمال خرابی موتورها است. از چنین پیش بینی ای ، مشتریان می توانند تغییرات موتور را برنامه ریزی کنند. ماشین چاپ هایدلبرگر دروک دارای چاپخانه های عظیم مجهز به بیش از 1000 حسگر است. اینها نمونه هایی از کاربردهای مجزا سنسورها در حوزه های خاص است. با این حال ، هنگامی که قطعات مجزا برای تولید تصمیمات کارآمدتر و موثرتر به هم متصل شوند ، این پدیده کاملاً تغییر خواهد کرد. بنابراین ، زیرساخت مه باید توانایی تأمین منابع فیزیکی برای محاسبه ، شبکه سازی ، ذخیره سازی و حافظه را برای دستیابی به خدمات کارآمد رایانش مه را داشته باشد.

2) الزامات شبکه

این شبکه یکی از گلوگاه های اصلی در رایانش مه است ، جایی که میلیاردها دستگاه متصل در لبه شبکه داده می خورند و مصرف می کنند [68]. بیشتر دستگاه های حسگر و محرک به پهنای باند کم نیاز دارند اما تعداد بیشتری از دستگاه ها به طور همزمان متصل می شوند. بنابراین ، فن آوری های اتصال شبکه موجود مانند LAN ، MAN ، WAN یا PAN باید بیشتر مورد بررسی قرار بگیرند و اصلاحاتی برای کنار آمدن با محیط محاسبات Fog برای تسهیل تعداد بیشمار IoT لازم است. اپراتورهای شبکه به طور فزاینده ای در تحقیقات جدید فناوری دسترسی بی سیم سرمایه گذاری می کنند ، زیرا تعداد دستگاه های هر کاربر روز به روز در حال افزایش است. به عنوان مثال ، در شبکه تلفن همراه همراه ، ایستگاههای پایه تعداد محدودی پیوند دارند [20]. با افزایش تعداد موارد ، این

ایستگاه ها باید از تعداد بیشتری دستگاه پشتیبانی کنند. دستگاه های مه شکن باید به عنوان روتر برای دستگاه های اینترنت اشیا همسایه و به عنوان یک واحد پردازش اولیه برای استفاده از اینترنت اشیا در محیط مه عمل کنند. هر دستگاه مه باید اتصال انعطاف پذیری به دستگاه های لایه پایین و بالا داشته باشد. شبکه های موقت تلفن همراه به دلیل قابلیت تحرک و هزینه کمتری که دارند می توانند به عنوان مبنایی برای شبکه مه عمل کنند [20]. از این رو ، اتصال و تحرک از نیازهای اصلی شبکه مه است.

3) دستگاه های مه

رایانش مه در اصل برای پشتیبانی از فناوری های مرتبط با اینترنت اشیا برای انجام پردازش در سطح لبه در نظر گرفته شده است. پروژه های معدن [69] ، [70] در وسط دریا ، ناوگان هواپیمایی یا یک کشتی [70] می توانند به تعداد زیادی حسگر مجهز شوند ، بنابراین ارسال و ذخیره تمام داده های تولید شده در زمان واقعی غیرممکن است ابر. برخی از محاسبات ، پردازش و خدمات میانی توسط دستگاه های رایانش مه انجام می شود. بنابراین ، لایه مه باید دارای دستگاه های مدیریت حسگر ، پردازش مه و دستگاه های ذخیره سازی و Fog gateway باشد. همه این دستگاه ها به طور مشترک برای مدیریت و انجام وظایف در هواپیمای مه کار خواهند کرد. در اینجا ، ما در مورد دستگاههایی که برای استقرار رایانش مه مورد نیاز هستند نام میبریم:

۱- دستگاه های اینترنت اشیا

۲- دستگاه های پردازش مه

۳- دستگاه های دروازه برای مه

ب - پلتفرم

این سیستم عامل برنامه ها و زیرساخت ها را در محیط مه مدیریت می کند. این امر از تخصیص منابع ، برنامه ریزی ، تحمل خطا ، اجاره چندگانه ، امنیت و خصوصیات در رایانش مه مراقبت می کند. بر اساس طبقه بندی مه ، ما در این بخش درباره شرایط سیستم عامل برای محاسبه مه صحبت می کنیم.

1) تخصیص و برنامه ریزی منابع

دستگاههای ناهمگون چالشهای اصلی در ایجاد تخصیص و برنامه ریزی مناسب منابع در مه هستند. اگر می خواستیم از قدرت محاسبه دستگاه های بیکار استفاده کنیم ، باید کارها را روی این دستگاه ها به طور کارآمد زمان بندی کنیم. در غیر این صورت ، پردازش برنامه IoT در مه با موضوعات پیچیده ای روبرو خواهد شد ، که مانع تحقق هدف آگاهی از تأخیر خواهد شد. دو مورد از نیازهای اصلی برای تخصیص و برنامه ریزی منابع ، در دسترس بودن و کارایی است. منابع موجود در مه اختصاصی نیستند ، بنابراین باید از دسترس بودن آنها اطمینان حاصل شود. از طرف دیگر ، عدم تخصیص و برنامه ریزی کارآمد منابع ممکن است منجر به تأخیرهای ناخواسته در پردازش کلی شود.

2) الزامات خدمات

سرویس های مه را می توان به صورت درخواست های کاربر یک یا چندگانه تعریف کرد ، جایی که کاربر به طور مداوم از نتیجه سرویس به روز می شود تا زمانی که عضو اشتراک آن سرویس باشد. این به این معنی است که نتیجه خدمات ثابت نخواهد شد و تا پایان سرویس تغییر خواهد کرد. دستگاه مه و Fog server پردازش میانی را انجام می دهند که بین درخواست کاربر و خروجی سرویس رخ می دهد. سرور مه ممکن است در صورت لزوم برای پردازش و بازیابی اطلاعات با ابر ارتباط برقرار کند. به عنوان مثال ، اگر در سیستم حمل و نقل هوشمند بهترین مسیر را بر اساس ترافیک لحظه ای انتخاب کنیم ، سرویس مه تا پایان سفر در بهترین مسیر به روز می شود. در این حالت ، برای حفظ استاندارد خدمات ، باید کاهش خطا ، کیفیت خدمات ، تأخیر شبکه و مصرف برق را در نظر بگیریم.

a: مقاومت در برابر خطا تحمل خطا به سیستم امکان می دهد تا عملکرد خود را ادامه دهد حتی در صورت خرابی بخشی از سیستم. این خرابی ممکن است خرابی نرم افزار ، خرابی سخت افزار یا خرابی شبکه باشد. راه حل تحمل خطا منجر به ایجاد یک سیستم کاملاً عملیاتی می شود که در آن سیستم به جای خاموش شدن کامل ، با قابلیت کمتری به کار خود ادامه می دهد [78]. تحمل خطا بیشتر در ابر [79] - [89] بررسی می شود. با این حال ، لازم است که تحمل عیب در مه نیز بررسی شود. اگرچه بسیاری از کارهای تحقیقاتی به لزوم کشف مسائل مربوط به تحمل عیب پرداخته اند [5] ، [9] ، [64] در محاسبه مه ، اما هیچ یک در واقع موضوع را بررسی نکرده اند. ما در بخش VIII-B با جزئیات بیشتری موضوع تحمل خطا را مورد بحث قرار می دهیم.

b: کیفیت خدمات QoS (QoS) یک سرویس مهم برای رایانش مه است که براساس قابلیت اطمینان ، تاخیر شبکه ، توان مصرفی و انرژی است. علاوه بر اینها ، مدیریت منابع ، مدل مصرف برق ، سیاست زمانبندی و مدیریت قطع برق نیز برای اطمینان از QoS مهم است. اگر برخی حسگرها به هر دلیلی از کار بیفتند ، می توان بر دقت نتیجه یا عملکرد آن تأثیر گذاشت. مه برای کار با سیستم های حساس به تأخیر در نظر گرفته شده است. بنابراین ، باید اطمینان بالایی را با اطمینان QoS دقیق حفظ کند. در غیر این صورت ، معیارهای آگاهی از تأخیر محقق نخواهد شد. مادسن و همکاران [90] پیشنهاد کرد که در دسترس بودن روش ها و الگوریتم های مختلف با اطمینان از قابلیت اتصال به شبکه و اطلاعات ، برای اطمینان از صحت ، که برای ساخت پروژه های مبتنی بر رایانش مهمی بسیار مهم است ، کار می کنند.

3) امنیت و حریم خصوصی

در عصر فناوری ، مردم هنگام استفاده از برنامه های مختلف و خدمات وب ، به طور حتم در هر اطلاعات اصلی به اشتراک می گذارند. اطلاعات شخصی ما دیگر شخصی نیست. اکنون به بسیاری از غول های فناوری تعلق دارد زیرا ما بطور منظم از خدمات رایگان آنها استفاده می کنیم [91]. یک مثال ساده این است که اگر کسی از تلفن Android بدون تنظیمات امنیتی استفاده کند ، سیستم عامل Android داخلی به طور خودکار GPS و خدمات نقشه را اجرا می کند ، برای این کار می تواند تمام فعالیتهای مربوط به مکان در مورد کاربر را جمع آوری کند. بنابراین اطلاعات مربوط به زمان و کدام کشور از یک کاربر بازدید کرده است ، کاربر در کجا غذا صرف کرده است ، کاربر از کدام مسیر برای رفتن به دفتر ، خانه و غیره استفاده می کند در دسترس این شرکت ها قرار خواهد گرفت. با این حال ، این غول های فناوری ممکن است استدلال کنند که داده های ما را برای دیگران فاش نمی کنند ، زیرا آنها فقط داده های ما را فقط در جدول زمانی ما می توانند ببینند. با این حال ، یک حادثه اخیر در فیس بوک نتوانست ما را در مورد صداقت این غول های فناوری متقاعد کند [92]. الگوی رایانش مه کاملاً توزیع شده است و بیشتر اوقات مدیریت مرکزی آن در نظر گرفته نشده است. هنگامی که برنامه کنترل کامل دستگاه را ندارد ، داده های حساس ممکن است در دستگاه میانی پردازش شوند. از طرف دیگر ، کاربر کنترل کاملی روی برنامه های Fog نخواهد داشت. کاربران برای حفظ حریم خصوصی خود و محافظت از آن در برابر اشخاص بالقوه و بسیار مضر به

روشهای محافظت کننده و ابتکاری بیشتری احتیاج دارند [20]. به همین ترتیب ، ارائه دهندگان برنامه های Fog نیز برای محافظت از برنامه خود در برابر سرقت داده های ناخواسته ، نیاز به توسعه امنیت دارند. سه نوع امنیت مختلف باید تضمین شود: امنیت اتصال خالص کار ، امنیت داده و حریم خصوصی کاربر. امنیت اتصال شبکه و امنیت داده ها از دو نظر کاربر و ارائه دهنده قابل اجرا هستند. علاوه بر این ، حریم خصوصی کاربر نیز مهم است زیرا پردازش مه در بیشتر موارد بر روی داده های کاربر انجام می شود.

4) اجاره چندگانه

چندین مستأجر برای خدمات یکسان با مدت زمان جداگانه برای هر مستاجر به عنوان خدمات چند اجاره ای شناخته می شوند. اجاره چندگانه برای مه مهم است زیرا منابع محدودی در محیط مه وجود دارد. با فعال کردن اجاره چندگانه ، یک نمونه در دستگاه Fog اجرا می شود و به چندین مستاجر (کاربران) خدمت می کند. اجاره چندگانه می تواند مبتنی بر کانتینر باشد یا می تواند مبتنی بر مجازی سازی معمول باشد. مجازی سازی مبتنی بر کانتینر یک راه حل مجازی سبک وزن و قدرتمندتر است که Fog می تواند با اتخاذ آن ، سریع ترین راه حل پردازشی را ارائه دهد. مجازی سازی مبتنی بر کانتینر برای تسهیل مجازی سازی نیاز به تقلید از سیستم عامل ندارد. بنابراین ، مدیریت و مهاجرت راحت تر خواهد بود. اجاره چندگانه یک شرط لازم برای سیستم عامل است و باید قبل از استقرار تعریف شود. اجاره چندگانه ممکن است منجر به تخریب عملکرد و مسائل امنیتی شود [29]. بنابراین ، انزوای کافی و ایمن لازم است.

5) مدیریت

مدیریت مه را می توان متمرکز یا متمرکز کرد. از آنجا که دستگاههای موجود در محیط مه به دامنه های مختلفی تعلق دارند ، مدیریت متمرکز همیشه امکان پذیر نیست. روش دیگر ، پردازش برنامه های کاربردی اینترنت اشیا در خوشه های مختلف Fog انجام می شود ، بنابراین مدیریت در این حالت از یک ماهیت توزیع شده پیروی می کند. به طور خلاصه ، مدیریت سیستم عامل در یک مه باید تعریف شود. در مورد مدیریت سطح پایین ، فرآیندهای مشابهی باید برای دستگاه های مختلف Fog استفاده شود تا مسائل مدیریتی را برطرف کند.

ج. کاربرد

برنامه ها باید شرایط خاصی را برای اجرا در محیط مه شکن داشته باشند. در اینجا ، ما در مورد ویژگی های مورد نیاز برنامه ها برای اجرا بحث می کنیم.

1) الزامات درخواست

الف: مقیاس پذیری

تعداد دستگاه های اینترنت اشیا روز به روز در سراسر جهان به سرعت در حال افزایش است ، که مسئله جدیدی برای مقیاس پذیری را به وجود می آورد. بنابراین ، ما باید با مقیاس گذاری دستگاه ها و خدمات در محیط رایانش مه مقابله کنیم. بسیاری از کارهای تحقیقاتی ، که در آن تریلیون دستگاه اینترنت اشیا are درگیر هستند ، مانند لی و همکاران ، وابستگی به رایانش ابری برای پردازش کاربرد اینترنت اشیا مشاهده شده است. [93] با این حال ، اجرای کل برنامه در ابر در چنین محیطی که دستگاه های اینترنت اشیا are مقدار زیادی داده تولید می کنند ، نه عملی است و نه کارآمد. دستگاه های اینترنت اشیا نه تنها ثابت هستند بلکه در بیشتر موارد نیز متحرک هستند. از این رو ، حفظ وضعیت های دستگاه و قابلیت دسترسی در ابر که به طور مداوم تغییر می کند کار ساده ای نیست. همچنین ، با افزایش تعداد دستگاه های اینترنت اشیا ، درخواست و انتخاب دستگاه های اینترنت اشیا برای برنامه های اینترنت اشیا بسیار مهم است [59]. سیستم رایانش مه باید یک سیستم خودمختار باشد که در آن اجرای برنامه توسط دستگاه شرکت کننده بصورت خودکار از جمله مقیاس پذیری انجام شود.

ب: ناهمگنی

برای هر سیستم اینترنت اشیا ، دستگاه ناهمگن یک ویژگی ذهنی اساسی است که در آن ناهمگنی دستگاه در هر سطح در الگوی رایانش مه وجود دارد. برداشت از پیچیدگی دستگاه نیز تا حدی لازم است. ناهمگنی دستگاه فقط به تنوع خدمات و پروتکل ها اشاره نمی کند ، بلکه به مجموعه ای از سطح افقی و عمودی معماری مه نیز اشاره دارد [59]. برای پرداختن به این ناهمگنی ، Giang و همکاران [59] ، سه نوع دستگاه Fog را طبقه بندی کرد: محاسبه ، ورودی / خروجی (IO) و گره های لبه ای. گره های لبه سنسورها و محرک ها هستند ، گره های IO دستگاههایی با منابع محدود هستند که بیشتر مسئول ارتباطات دلالتی هستند و گره های محاسباتی امکانات محاسباتی را ارائه می

دهند. از بین سه نوع گره، گره های IO و عموماً پویا و قابل تنظیم و یا در صورت لزوم قابل برنامه ریزی هستند. بر اساس قابلیت و هدف طراحی آن می توان هر سه گره را در یک دستگاه واحد پیاده سازی کرد. دروازه هوشمند نمونه ای از چنین پیاده سازی است. برای استفاده از قابلیت انواع مختلف دستگاهها در محیط اینترنت اشیا، بدیهی است که برنامه باید به گونه ای طراحی شود که بتواند وظیفه خود را بر روی چندین دستگاه بدون در نظر گرفتن ظرفیت و موقعیت آن انجام دهد. دقیق تر، برنامه باید بتواند از حداکثر منابع محاسبه موجود از طریق میان افزار استفاده کند.

ج: به موقع تعامل

چرخه ادراک-عمل (PA) عملکرد اساسی یک سیستم عصبی است که جریان دایره ای بین ارگانسیم های حسی و عملکردهای آن را نسبت به عملکرد آن سنجش حفظ می کند. چرخه PA همچنین مشخصه برنامه های اینترنت اشیا است، جایی که زیرساخت ابر و مه نیازهای به موقع و منطق برنامه را برای ارتباط برقرار می کند. گیانگ و همکاران [59] چهار مدل تعامل را برای چرخه PA در یک محیط مه شناسایی کرد. نمونه هایی از این مدلها عبارتند از: اقدامات، (III) تعاملی که ابر با دستگاهی در شبکه محلی ایجاد می کند، که به اقدامات نیمه فوری نیاز دارد و (IV) ارتباط بین برنامه های مربوط به اینترنت اشیا در ابر. با این حال، کار آنها نقش Fog server را در نظر گرفته است، که چندین دستگاه مه را در یک خوشه خاص مدیریت و نگهداری می کند. از طرف دیگر، تعامل PA را می توان به اقدامات فوری، نیمه فوری و تأخیری تقسیم کرد تا با بهره وری بیشتر از نیازهای کاربردی اینترنت اشیا استفاده کند. تأخیر را می توان در نوع پردازش انجام داد که هیچ مشکلی در مورد به موقع بودن ندارد و می تواند توسط ساختار زیرساخت ابر پردازش شود.

د: تحرک

تحرک دستگاه یک احتمال طبیعی است و یکی از اصلی ترین الزامات پیاده سازی سیستم عامل اینترنت اشیا است [59]. از منظر مه، فقط دستگاه های لبه ای نیستند که متحرک خواهند بود بلکه دستگاه های محاسباتی و ذخیره سازی در لایه مه هستند. مدیریت دستگاه های تلفن همراه در دو صفحه مختلف و همگام سازی آنها با یکدیگر چالش برانگیز است. برای اطمینان از در دسترس بودن منابع و انجام موفقیت آمیز کار، تقسیم وظیفه، تکثیر و مهاجرت مورد نیاز است. این ناهنجاری مکانیکی قبلاً در ابر در نظر گرفته شده است اما نیاز به بررسی مجدد آنها با در نظر گرفتن تحرک است [94].

2) الزامات کاربر

نیازهای کاربر را می توان با محدودیت های مختلف تغییر داد. اول از همه ، یک کاربر ممکن است بخواهد کار ارسالی را در یک زمان لازم ، که به عنوان زمان دار ذکر می شود ، انجام دهد. ثانياً ، کاربر ممکن است محدودیت هایی را برای بودجه تعیین کند. ثالثاً ، در مورد برخی از کاربران ، آنها ممکن است به بودجه اهمیتی ندهند اما زمان پاسخ از اهمیت بالایی برخوردار است. چهارم ، برخی از کاربران ممکن است از دقت بالایی برخوردار باشند. این بدان معنی است که کاربر ممکن است به دنبال نتایج دقیق نباشد ، بلکه به دنبال نتایج سریع است که می تواند با برخی خطاهای منطقی ارائه شود. Aazam و Huh [54] اظهار داشتند که پیش از تخصیص و پیش بینی منابع به رفتار کاربر و احتمال استفاده آینده از منابع بستگی دارد. دستجردی و همکاران [4] ، [95] اظهار داشت که دستگاه های لبه با در نظر گرفتن رفتار کاربر و شرایط شبکه ، بهینه سازی را انجام می دهند.

3) مدل سازی برنامه

با در نظر گرفتن نیازهای برنامه های کاربردی در مه ، دو نوع مدل سازی برنامه امکان پذیر است. بیشتر دستگاه های اینترنت اشیا به طور دوره ای تاپل تولید می کنند که می تواند به عنوان یک جریان در نظر گرفته شود. برای دستیابی به نتایج دقیق ، این جریان ها باید در زمان واقعی پردازش شوند. متناوباً ، برنامه ای که پردازش را بر اساس مجموعه داده های ذخیره شده قبلی انجام می دهد ، می تواند شامل برنامه های مبتنی بر ریز سرویس باشد. مزیت ریز سرویس این است که می تواند تمام عملکردها و کتابخانه های مورد نیاز را در یک سرویس متصل کند ، سرویس هایی که می توانند بدون وابستگی بالاتر از معاون کنترل میکروسر کار کنند. از این رو ، مدل سازی برنامه در Fog می تواند مبتنی بر جریان یا ریز سرویس باشد.

۷- ابعاد برنامه های مبتنی بر رایانش مه :

چندین برنامه برای ارائه خدمات روان به زیرساخت رایانش مه نیاز دارند. اینها شامل سیستم های حمل و نقل هوشمند ، واقعیت افزوده و مجازی ، مراقبت های بهداشتی ، پخش ویدئو ، خانه های هوشمند و شهرهای هوشمند است. برای ارائه خدمات نیز به سیستم عامل و برنامه نیاز دارید. در این بخش ، ما در مورد برخی از کارهای تحقیقاتی بحث می کنیم ، که بطور خاص به کاربرد رایانش مه می پردازند. ما هر کار را بر اساس سهم آنها در زیرساخت ها ، سیستم عامل و برنامه های مه به

عنوان تعریف شده در طبقه بندی خود ارزیابی می کنیم. بدیهی است که هر سه نوع خدمات با هم ارتباط دارند. با این حال ، هر محقق فقط روی یک یا چند مورد از این جنبه ها تمرکز کرده است. نقشه برداری از کارهای مرتبط با طبقه بندی پیشنهادی ما در یافتن شکاف تحقیقاتی در برنامه های رایانش مه کمک خواهد کرد.

الف. سیستم حمل و نقل هوشمند

چندین کار تحقیقاتی در مورد سیستم های انتقال هوشمند که از رایانش مه استفاده می کنند ، انجام شده است. در این بخش ما به چند کار در سیستم حمل و نقل هوشمند مبتنی بر Fog پرداخته ایم و سپس موارد اصلی را که باید مورد توجه قرار گیرند را شناسایی می کنیم. تروونگ و همکاران [96] معماری یک کارگاه موقت شبکه (VANETs) Vehicle را به نام Fog Software Defined Networking (FSDN) پیشنهاد کرد ، که ترکیبی از SDN و مه را برای ارائه راه حل بهتر ارائه می دهد. از آنجا که SDN دارای قابلیت برنامه ریزی ، انعطاف پذیری ، دانش جهانی و قابلیت های مقیاس پذیری است و مه دارای آگاهی از مکان و حساسیت به زمان است ، ترکیب این دو از چالش های کلیدی VANET استفاده خواهد کرد. سیستم پیشنهادی علاوه بر کاهش تأخیر و بهینه سازی استفاده از منابع ، می تواند از طریق کنترل متمرکز ارتباطات را در میان وسایل نقلیه ، زیرساخت ها و ایستگاه های پایه افزایش دهد. با این حال ، کنترل کننده SDN مرکزی سیستم پیشنهادی جایی است که گلوگاه سیستم پیشنهادی رخ می دهد. این سیستم بر روی زیرساخت ها و نیازهای شبکه متمرکز شده است. کنترل کننده مه برای اجرای سرویس استفاده می شود. کار بر روی نیازهای سیستم عامل و برنامه متمرکز نبود. بررسی VANET ها در مه نیز در Giang و همکاران انجام شده است. [97] ، جایی که آنها چگونگی توسعه برنامه های انتقال هوشمند (VANET) را با استفاده از روش محاسبه مه جستجو کردند. رانندگی با وسایل نقلیه در یک منطقه شهری نیاز به تصمیم فوری در مورد فعالیتهای مختلف از جمله تغییر مسیر ، تغییر خط ، کاهش سرعت ، نگاه کردن به موانع و غیره دارد. از این رو ، برنامه ها باید کلیه جزئیات مرتبط را برای عملکرد این فعالیتهای جمع آوری کنند. نویسندگان در مورد برنامه حمل و نقل هوشمند مبتنی بر مه نیاز به مواردی مانند برنامه نویسی انتزاع و برنامه های کاربردی دارند. این کار مدل سازی برنامه را بررسی کرد اما سایر جنبه های کاربردی و نه زیرساخت ها یا سیستم عامل را بررسی نکرد.

ب - وسایل نقلیه به عنوان زیرساخت های مه

هو و همکاران [98] ایده محاسبه مه شکن (VFC) را پیشنهاد داد ، که از وسیله نقلیه به عنوان زیرساختی برای محاسبه و ارتباط استفاده خواهد کرد. معماری VFC با ارائه سرویس به دستگاه های لبه واقع در نزدیکی آنها ، از منابع محاسبه خودرو استفاده می کند. این منابع فراوان هر اتومبیل در حال حرکت را که می تواند کیفیت خدمات را افزایش دهد جمع می کند. آنها با استفاده از تجزیه و تحلیل کمی در سناریوهای مختلف ، رابطه جالبی را بین اتصال ، تحرک ، قابلیت ارتباطی و رفتار پارکینگ کشف کردند. این چهار ویژگی به ما کمک می کند تا استفاده از منابع وسایل نقلیه را درک کنیم ، که به دستیابی بهتر از منابع بلااستفاده کمک می کند.

ج- واقعیت افزوده و مجازی

برنامه های واقعیت افزوده بسیار حساس به زمان هستند. یک تاخیر اندک می تواند منجر به خطاهای جدی در تجربه کاربر شود. بنابراین ، راه حلهای مبتنی بر رایانش مه در این حوزه پتانسیل زیادی خواهند داشت [4]. این عبارات برای بازی های مبتنی بر واقعیت مجازی (VR) یا VR نیز قابل استفاده است. زائو و دیگران [99] یک بازی تعامل رایانه ای با هوش تقویت شده را پیشنهاد داد که از زیرساخت مه و ابر استفاده می کند. مه تجزیه و تحلیل زمان واقعی مانند پردازش سیگنال را که نیاز به طبقه بندی حالت مغز دارد و سایر تجزیه و تحلیل ها مانند طبقه بندی مدل به روز شده از ابر را انجام داد. با این حال ، کار آنها فقط در زیرساخت مه متمرکز بود اما از بسیاری از جنبه های مربوط به سیستم عامل و برنامه نادیده گرفته شد.

د- مراقبت های بهداشتی

رویکرد رایانش مه همچنین خدمات مراقبت های بهداشتی مبتنی بر سنسور را در زمان واقعی امکان پذیر می کند. رحمانی و همکاران [100] معماری سیستم به کمک مه را برای سیستم مراقبت های بهداشتی پیشنهاد کرد. دروازه هوشمند سلامت الکترونیکی مولفه اصلی این معماری است که داده های تولید شده از حسگرها را پردازش می کند و یک امتیاز هشدار زودرس (EWS) را برای اطلاع رسانی در مورد هرگونه فوریت های پزشکی ایجاد می کند. آنها بسیاری از جنبه های طبقه بندی ما را در نظر گرفتند. اما لازم است که هر جنبه به طور گسترده بررسی شود ، که این مطالعه انجام نداد. یکی دیگر از معماری های مراقبت های بهداشتی مبتنی بر مه توسط محمود و همکاران ارائه شد. [26] کارهای آنها عمدتاً معطوف به تأخیر شبکه ، میزان مصرف برق و بهینه سازی ارتباطات در

سرویس بهداشتی مبتنی بر مه بود. با این حال ، سیستم عامل ، برنامه و نیازهای کاربر مورد بررسی قرار نگرفت.

ذ- شهر هوشمند

برنامه های مرتبط با شهر هوشمند نیاز به پردازش داده های حسگر به صورت زمان واقعی دارند ، جایی که رایانش مه می تواند نقش اصلی را داشته باشد. جوردانو و همکاران [101] معماری رنگین کمانی را پیشنهاد داد ، که از برنامه های مختلف در یک شهر هوشمند پشتیبانی می کند. چارچوب مورد نظر رنگین کمانی سه برنامه کاربردی شهر هوشمند از جمله نقشه برداری از آلودگی صوتی ، شبکه های تخلیه شهری و خیابان هوشمند را ارزیابی کرد. این کار یک روش مبتنی بر عامل توزیع شده در لایه میانی را در بین زیرساخت های فیزیکی و ابر پیشنهاد کرده است. با این حال ، کار به جنبه های کاربردی و پلت فرم به جز مدل سازی برنامه متمرکز نبود. جدول 4 خلاصه ای از برنامه های مبتنی بر مه که در بالا بحث شد را نشان می دهد که در طبقه بندی پیشنهادی ما ترسیم شده است. به طور خلاصه ، می توان نتیجه گرفت که بیشتر آثار متمرکز بر زیرساخت ها و کاربرد بوده اند.

Author & Year	Application Type	Infrastructure			Platform					Application					
		Infrastructure requirements	Network requirements	Fog Devices	Resource Allocation and Scheduling	Service requirements	Security and privacy	Multi-tenancy	Management	Application requirements				User Requirements	Application Modeling
										Scalability	Heterogeneity	Interaction timeliness	Mobility		
Truong et al. [96]	Smart Transportation System	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Giang et al. [97]	Smart Transportation System	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Hou et al. [98]	Vehicles as infrastructure	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Zao et al. [99]	Augmented and virtual reality	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Rahmani et al. [100]	Healthcare	✓	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✓	✗	✓
Mahmud et al. [101]	Healthcare	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Giordano et al. [102]	Smart City	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

جدول ۴- ارزیابی کارهای موجود در برنامه های مه

۸- وضعیت تحقیقات رایانش مه

در این بخش ، ما بر برخی از کارهای تحقیقاتی موجود در زمینه رایانش مه تمرکز می کنیم. ما در مورد کارهای تحقیقاتی از چهار حوزه تحقیقاتی مختلف رایانش مه بحث می کنیم. این مناطق عبارتند از: تخصیص و برنامه ریزی منابع ، مدیریت خرابی ، ابزارهای شبیه سازی و ریز خدمات.

A. تخصیص و برنامه ریزی منابع در رایانش مه

رایانش مه به دلیل محاسبات سطح لبه و ماهیت ناهمگن سریع در حال تکامل و رشد سریع است. در این بخش ، ما چندین کار تحقیقاتی را ارائه می دهیم که در طی دو سال گذشته انجام شده است. ما همچنین کارهای تحقیقاتی ارائه شده را با یک بحث مقایسه ای خلاصه می کنیم تا شکاف های تحقیقاتی در این زمینه را برطرف کنیم. بیشتر کارهای بررسی شده مربوط به تخصیص منابع و زمان بندی در فضای ابری و مه است. با این وجود برخی کارها فقط در محیط مه انجام شده است.

1) تخصیص منابع و برنامه ریزی برای محیط ابر-مه

آلسار و همکاران [102] روشهای تخصیص منابع را برای یک بستر مشترک متشکل از پارادایم های مه و ابر پیشنهاد کرد. الگوریتم پیشنهادی آنها با در نظر گرفتن سه شرط مختلف برای مدیریت درخواست کاربر و متعادل کردن بار کار ، بر اساس قوانین درخت تصمیم گیری خطی استوار است. شرایط ظرفیت VM ، زمان اتمام و اندازه خدمات است. هر شرط دارای دو شاخه است: ظرفیت VM به اندازه کافی یا کافی منشعب می شود. زمان اتمام شامل اکنون یا بعداً است و اندازه خدمات به کوچک یا بزرگ تقسیم می شود. در بعضی موارد ، این شامل خدمات در صف است که با بله یا خیر نشان داده می شوند. آنها از $1 / m / 1$ ، با (1) / نمایندگی دلال ابر ، / (m) برای بسیاری از مسیرها ، / (m) برای بسیاری از کارگزاران مه و / (1) برای کاربران دستگاه اینترنت اشیا استفاده کردند. با استفاده از این روش ، سربار کل برای پردازش داده های بزرگ در سیستم کاهش یافت. در کار آنها ، در دسترس بودن سرورهای ابری و مه تضمین شده بود و زمان پاسخگویی سریع برای برآورده سازی QoS حاصل شد. SLA برای کاربران نیز متفاوت بود ، جایی که منابع مشترک و ذخیره شده ارائه می شد. با این حال ، در دسترس بودن و QoS به طور گسترده مورد مطالعه قرار نگرفت.

دنگ و همکاران [103] چارچوبی را برای تخصیص بار کاری در فضای ابری و مه به منظور بررسی مبادلات مصرف تاخیر مصرف برق ارائه داد. آنها مشکل تخصیص بار کار را به مشکلات اصلی و فرعی تعریف کردند ، که می تواند از طریق زیر سیستم های مرتبط حل شود. آنها برای حل مسئله از الگوریتم مجارستان و الگوریتم تجزیه متغیرهای عمومی (GBD) استفاده کردند. برای اثبات اینکه مه یک مکمل ابر است ، نتایج عددی و شبیه سازی ارائه شد. با این حال ، ماهیت پیچیده حجم کار و منابع در کار آنها مورد مطالعه قرار نگرفت.

بروگی و همکاران [104] نمونه اولیه ای از ابزاری موسوم به 'FogTorchQ' که قادر است قبل از استقرار یک برنامه ترکیبی در زیرساخت مه ، نیازهای سخت افزاری ، نرم افزاری و QoS را برآورده کند معرفی کردند. ابزار پیشنهادی شبیه سازی های مونت کارلو را دستکاری می کند و فقط QoS لینک ارتباطی را در نظر می گیرد. مصرف منابع و شرایط اطمینان QoS برای طبقه بندی صلاحیت استقرارها انجام شد. الگوریتم پیشنهادی بر اساس مرحله پیش پردازش و جستجوی عقب ردیابی بود. برای یافتن استقرار واجد شرایط ، پیش پردازش از ورودی حاصل از نتایج حاصل از الگوریتم

جستجوی ردیابی پستی استفاده می کند. با این وجود، در دسترس بودن و تأخیر در محیط مه در مقایسه با مصرف منابع و پیوندهای ارتباطی از اهمیت بیشتری برخوردار است.

به منظور اطمینان از استفاده بهینه از منابع و زیرساخت های شبکه در مه و ابر، Davy و Taneja [105] الگوریتم پینگ ماژول را پیشنهاد دادند که به طور کارآمد ماژول های کاربردی اینترنت اشیا را در زیرساخت مه و ابر مستقر می کند. آنها از جستجوی کران پایین استفاده کردند و الگوریتم های عملکرد را برای یافتن یک گره شبکه واجد شرایط در Fog و cloud مقایسه کردند. الگوریتم Module Mapping نقشه ای با گره ها، که برای تکمیل عملیات محاسبات مناسب هستند، برگرداند. اگر برنامه به پردازش سریعتر نیاز دارد، برنامه نزدیک به دستگاه منبع مستقر می شود. با این حال، کار CPU، RAM و پهنای باند را برای یافتن بهترین منابع در نظر گرفت. در چنین حالتی، منبع ابری همیشه بهترین منبع خواهد بود، بنابراین لازم است پارامترهای دیگری مانند زمان پاسخ و در دسترس بودن منابع مشخص شده در نظر گرفته شود.

یین و همکاران [52] یک سناریوی جریان داده بزرگ با کمک مه را مورد مطالعه قرار داد، جایی که دستگاه های مه وظیفه پیش پردازش داده های خام را برای برنامه های میزبانی شده در ابر با استفاده از منابع استفاده نشده دستگاه های مه دارند. در کار خود، نرم افزار کنترل کننده شبکه تعریف شده (SDN) به طور مداوم حجم داده را برای ارسال به دستگاه مه برای پیش پردازش تنظیم می کند. مسئله محاسبات مشارکتی به عنوان یک مسئله حداکثر رفاه اجتماعی تعریف شد و یک روش جهت متناوب ترکیبی الگوریتم ضرب (H-ADMM) برای به حداقل رساندن بار محاسبات از طریق توزیع دینامیکی دستگاه های مه، ابر و SDN با استفاده از تبادل پیام ارائه شد. فرمول بندی مسئله به حداکثر رساندن رفاه اجتماعی، اندازه داده هایی را که به دستگاه مه اختصاص داده می شود، تعیین کرد. در طی فرمول سازی، از دست دادن ارزش اطلاعات با پیش پردازش و هزینه بهره برداری از مه و ابر در نظر گرفته شد. کار برای پردازش پس از آن کاملاً به ابر بستگی داشت، اما می توان قبل و بعد از پردازش را در مه انجام داد تا از برنامه های زمان واقعی حساس به زمان پشتیبانی کند.

Aazam و همکاران [106] الگوریتم برآورد تخمین منابع پویا را با تلفیق رکورد تاریخی مشتری خدمات ابری (CSC) در یک محیط مه بر اساس احتمال انصراف پیشنهاد کرد. حداقل مقدار احتمال انصراف 0.1 است و این مقدار براساس سابقه کاربر افزایش می یابد. با این حال، برای تخمین منصفانه منابع، احتمال انصراف برای مشتریان جدید 0.3 خواهد بود. برای مشتریان موجود و مشتریان، مشخصات مشتری مشخص است، بنابراین مقدار احتمال را می توان به راحتی محاسبه

کرد. به این ترتیب می توان استفاده کم از منابع را به حداقل رساند و احتمال از دست دادن سود کم خواهد بود.

2) تخصیص منابع و برنامه ریزی برای یک محیط مه

یک استراتژی تخصیص منابع بر اساس شبکه های پتری با زمان بندی قیمت دار (PTPN) توسط Ni و همکاران پیشنهاد شد. [107] برای رایانش مه . ایده اصلی این کار این است که کاربر می تواند منابع رضایت بخش را به طور خودکار از یک گروه منابع از پیش تخصیص یافته انتخاب کند. با ارزیابی اعتبار هم برای کاربران و هم برای زیرساخت های مه ، استراتژی پیشنهادی آنها به طور جامع هزینه زمان و قیمت را برای انجام وظایف در نظر گرفته است. کاربری که محدودیت اعتباری بالایی دارد قادر خواهد بود منابع بسیار قابل اعتمادی را برای انجام وظایف خود اختصاص دهد. با توجه به ماهیت پویا اعتبار کاربران و منابع ، در محاسبه صحیح آنها انحراف وجود دارد. برای حفظ QoS ، منابع با توجه به ظرفیت پردازش آنها ترتیب داده می شوند و به چند گروه تقسیم می شوند. علاوه بر این ، کاربران با اعتبار مشابه به چندین گروه اختصاص می یابند.

Pooranian و همکاران [108] یک الگوریتم ساده برای یافتن راه حل بهینه برای تخصیص منابع پیشنهاد داد. آنها این مشکل را به عنوان یک مسئله آگاه از مجازات **bin packing** در نظر گرفتند که در آن سرورها **bins** و **VM** ها بسته هستند. بر اساس انرژی بیکار ، حداکثر فرکانس و حداکثر انرژی ، هر سرور از نوع کاملی شده و پاداش می گیرد. این روش محاسبه می کند که چند ماشین مجازی را می توان در شکاف **t** یک سرور اختصاص داد. سرویس های مجازی بر اساس محدودیت های زمانی و فرکانسی ارائه می شوند. در نتیجه مجازات ، یک سرور به صورت ممنوعیت استفاده برای چند تکرار مجازات خواهد شد. هنگامی که سرور یخ تکرار را رد کرد ، برای انجام محاسبات بیشتر به جریان برمی گردد. روش های مجازات و پاداش برای به حداقل رساندن افزایش انرژی مصرف انرژی به کار می رود.

سان و ژانگ [63] الگوریتمی برای تأمین بودجه جمعیت برای یک محیط مه ، با ادغام منابع بیکار در شبکه محلی ، پیشنهاد دادند. مکانیسم تشویقی برای تشویق صاحبان منابع سنی برای مشارکت در محاسبه و انجام مشتاقانه وظایف خود استفاده شد. از طریق سازوکار پاداش و مجازات ، اطمینان حاصل می شود که شرکت کننده وظایف را به طور مثبت انجام می دهد. این کار مشابه ادبیات فوق

الذکر ارائه شده توسط Pooranian و همکاران است. [108] با این حال ، در این حالت ، پاداش و مجازات به جای شرکت سرور فیزیکی به شرکت کننده تعلق می گیرد.

3) خلاصه تخصیص و برنامه ریزی منابع در مه

بر اساس تحقیقات مرتبط در زمینه تخصیص و برنامه ریزی منابع در مه ، خلاصه ای در جدول 5 ارائه شده است. از این جدول ، می بینیم که بیشتر محققان بر تخصیص منابع در مه تمرکز کرده اند. برای تحقیق در مورد به اشتراک گذاری منابع و تخصیص بار کار ، کارهای تحقیقاتی بیشتری لازم است. همچنین ، تحقیقات بیشتر برای رسیدگی به بهره وری انرژی ، تعادل بار ، SLA و QoS در مه مورد نیاز است. ما در تحقیق قرار دادن مه دو موضوع عمده را شناسایی کردیم. در مرحله اول ، محققان تمایل به استفاده از حجم کار مصنوعی برای تأیید روش ها و الگوریتم های خود دارند. ثانیاً ، بیشتر محققان از شبیه سازی های مبتنی بر ابر استفاده کرده اند ، که چندان قانع کننده نیست زیرا مه دارای ناهمگنی نفوس تر و ماهیت پویاتر است. بنابراین ، لازم است تحقیقات بیشتری در مورد تولید حجم کار و شبیه سازی های مه انجام شود.

B. تحمل اشتباه در رایانش مه

پارادایم رایانش مه یک پلتفرم بسیار ناهمگن توزیع شده است که در آن احتمال خرابی دستگاه در مقایسه با ابر بسیار زیاد است. از آنجا که مه در حال تکامل است ، هنوز هیچ مطالعه ای در مورد تحمل خطا در محاسبه مه انجام نشده است. با این حال ، تحمل خطا بیشتر در الگوی محاسبات ابری مورد مطالعه قرار گرفته است.

اغلب ، میزان تحمل خطا با در دسترس بودن اندازه گیری می شود. در ابر ، با استفاده از تکنیک های تحمل گسل فعال و تحمل گسل واکنش پذیر ، در خطای کار یا سطح کار ، گسل ها برطرف می شوند. تکنیک های تحمل خطای واکنشی برای کاهش تأثیر خرابی ها بر روی سیستم هنگامی که در واقع خرابی ها رخ داده است ، استفاده می شود. تکنیک های مبتنی بر این سیاست مهاجرت شغلی ، ایست بازرسی / راه اندازی مجدد ، تکثیر ، بازگرداندن و بازیابی ، ارسال مجدد وظایف ، استثنا تعریف شده توسط کاربر و نجات گردش کار است. تحمل خطای فعال گسل ها را پیش بینی فعال می کند و اجزای مشکوک را با سایر اجزای سازنده جایگزین می کند بنابراین ، جلوگیری از بازیابی

خطاها و اشتباهات. تحمل خطاهای فعال از خود ترمیم ، مهاجرت پیشگیرانه و جوان سازی نرم افزار استفاده می کند ، که چند روش تحمل خطای پیشگیرانه در ابر است.

به گفته شارما و همکاران [110] ، دلایل خرابی در ابر متفاوت است و شامل خرابی نرم افزار و سخت افزار ، خرابی سرویس ، خرابی سرریز ، قطعی برق ، سیستم های منسوخ ، خرابی شبکه ، حملات سایبری و خطاهای انسانی است. مقابله با خطاهای موجود در رایانش مه که برای آنها گسل لازم است در هر مرحله بررسی شود ، بسیار مهم است ، نه تنها برای توقف ، بلکه همچنین برای فرایند انتقال و دریافت [111]. در این بخش ، ما در مورد برخی از کارهای تحقیقاتی موجود در مورد تحمل خطا در محاسبات ابری بحث می کنیم. ما به طور خاص بر روی مکانیسم های شکست منابع و وظایف تمرکز می کنیم. سپس ، ما کارهای موجود را خلاصه می کنیم و جهت تحقیق برای عدم موفقیت در رایانش مه را ارائه می دهیم.

جیانگ و هسو [112] یک طرح آماده به کار دارای دو سطح را برای مدیریت خرابی سرور در سیستم ابر ارائه دادند. در سیستم پیشنهادی آنها ، حالت آماده به کار سرد و گرم سیستم در دسترس است. هرگاه سرور خراب شود ، سیستم آماده به کار گرم جایگزین سرور خراب شده می شود و سرور خراب به تعمیرگاه ارسال می شود. پس از تعمیر ، سیستم در گروه آماده به کار قرار می گیرد. سیستم های گروه آماده به کار در حالت کاملاً خاموش هستند. کار مدلی را برای تعیین تعداد لازم سیستمهای آماده به کار سرد و گرم در ابر ارائه داد. با این حال ، این نوع مدیریت سخت افزار برای مه مناسب نیست زیرا بیشتر اوقات دستگاه های رایانش مه تحت مالکیت ارائه دهنده مه نخواهند بود. از این رو ، انتقال وظیفه بهترین راه حل برای خرابی سخت افزار است و این باید در اکثر موارد واکنش پذیر باشد ، مگر در مواردی که دستگاه مه متعلق به ارائه دهنده باشد. لاتیف و همکاران [113] یک طرح زمانبندی ابری را بر اساس الگوریتم قهرمانی لیگ چک چک پیشنهاد داد. آنها از یک روش انتقال وظیفه برای عدم اجرای مستقل کار استفاده کردند. در روش پیش بینی شده آنها ، حالت سیستم به صورت دوره ای با نشانگر ذخیره می شود ، بنابراین پس از انجام کار لازم نیست کار از ابتدا شروع شود. وقتی این وظیفه از کار افتاد ، به یک ماشین مجازی کم بار اختصاص داده می شود و از الگوریتم قهرمانی لیگ برای برنامه ریزی کارهای ناموفق استفاده می شود.

وو و همکاران [114] تکنیک تحمل خطا را با استفاده از مهاجرت به ابر پیشنهاد داده است. روش مدیریت خرابی عملی است ، که همیشه میزبان را کنترل می کند و به طور مداوم سعی می کند احتمال خرابی را پیش بینی کند. اگر پیش بینی درست شود ، سیستم سایر منابع موجود را جستجو

می کند و سپس انتقال انجام می شود. روش پیشنهادی دمای CPU، میزان استفاده از حافظه و سرعت فن CPU و غیره را کنترل خواهد کرد. برای استفاده از چنین روشی در محاسبه مه، بررسی بیشتر لازم است زیرا انواع دستگاه در مه متنوع است.

یک روش ترکیبی از کنترل نقطه اشاره و مهاجرت مبتنی بر عدم موفقیت پیشگیری توسط Egwutuoha و همکاران پیشنهاد شد. [115] برای HPC و cloud. در روش پیشنهادی، نویسندگان از یک ابزار نرم افزاری منبع باز با سنسورهای Lm برای نظارت بر سلامت رایانه استفاده کردند. از داده های نظارت، آنها نظارت بر قاعده را بسته به دما، سرعت فن، ولتاژ و استفاده از پردازنده برای پیش بینی خرابی تعریف کردند. این قوانین به ترتیب 1، 2 و 3 نشان داده می شوند که به ترتیب حالت طبیعی، هشدار و بحرانی را نشان می دهند. آنها از سه سیاست مختلف برای مهاجرت استفاده کردند. اولین مورد به ضرورت اجاره گره اضافی بستگی دارد. دوم گره را حذف می کند، که براساس وضعیت ناسالم نیست. در حالت سوم، وضعیت بحرانی در گره سر منتشر می شود. سرانجام، مدیر سیستم برای اقدامات بعدی مطلع می شود. این نوع روش ممکن است باعث افزایش سربار در مه شود. با این حال، اکتشاف بیشتر ضروری است. یک مطالعه اخیر نشان می دهد که تحمل خطای فعال بهترین راه حل برای ابر در مقایسه با محلولهای زائد است [88]. با این حال، دقت پیش بینی شکست عامل اصلی این نوع راه حل ها است. کارهای آنها نرم افزار، سخت افزار و رفتار ناپایدار را برای پیش بینی خرابی زیرساخت ها در نظر گرفت. به طور خاص، آنها شکست را بر اساس فرمول خطای $(err = (ActualTime - PredictTime) / ActualTime \times 100\%)$ تعریف کردند که از [118] و [119] مشتق شده است. توسط گائو و همکاران یک ملت ترکیبی از روش فعال و واکنشی استفاده شد. [118] برای مقابله با شکست وظیفه در محیط ابر. برای تشخیص خرابی ها، روش تشخیص تصادف و فاکتور تکثیر در این کار ارائه شده است. جدول 6 خلاصه ای از ادبیات بررسی شده در مورد تحمل عیب در ابر را نشان می دهد.

به دلیل ماهیت ناپایدار خرابی و ویژگی های ناهمگن، روش مدیریت خرابی ترکیبی برای محیط رایانش مه مناسب تر است.

Author & year	Proposed Algorithm	Infras-structure	Method applied	Evaluation Process	Workload	Method proposed for				Other Considerations			
						Resource Allocation	Resource Provisioning	Resource Sharing	Workload allocation	Energy Efficiency	Load balancing	SLA Parameters	QoS parameters
Alsaffar et al. [103]	Resource Allocation and Data Distribution	Fog / Cloud	Utilized 1/m/m/1 model where (1) refers to cloud broker, (m) refers to many paths, (m) refers to many Fog brokers in Fog environments, and (1) refers to IoT devices users.	Simulation (CloudSim)	Synthetic	Yes	Yes	Yes	No	No	No	Response time, target time	Server availability, fast response time
Deng et al. [104]	Generalized Benders Decomposition (GBD) algorithm and Hungarian Algorithm	Fog/ Cloud	Define optimal workload allocation among cloud and Fog which minimizes power consumption with a defined service delay. Proposed technique solved workload allocation problem in cloud and Fog.	Simulation (MATLAB) and Numerical	Synthetic	No	No	No	Yes	Yes	No	NA	NA
Ni et al. [109]	Dynamic resource allocation strategy	Fog	Ordering the resources into several groups based on their processing capabilities.	Testbed	Synthetic (Random)	Yes	No	No	No	No	No	NA	Price, completion time
Brogi et al. [105]	Backtracking search algorithm	Fog/ Cloud	Define the deployments of complex applications to Fog infrastructures, which fulfil hardware, software and service quality requirements.	Simulation	Yes	No	No	No	No	No	NA	Latency, bandwidth	
Taneja et al. [106]	ModuleMapping and LowerBound Algorithm	Fog/ Cloud	Efficient application deployment in Fog-Cloud environment for IoT based applications by employing module mapping algorithm for efficient utilization of network infrastructure and resources.	Simulation (iFogSim)	Synthetic	Yes	No	No	No	Yes	No	NA	Latency
Yin et al. [107]	Distributed Resource Sharing Scheme	Fog/ Cloud	Grounded on an algorithm known as Hybrid Alternating Direction Method of Multipliers (H-ADMM) algorithm	Numerical	NA	No	No	Yes	No	No	No	No	No
Pooranian et al. [110]	Energy- aware algorithm for Fog data center	Fog	Provide reward and penalty to each server depends on their energy characteristics such as maximum frequency, maximum energy and idle energy	Numerical and Simulation (iFogSim)	Worldcup98 workload [111]	Yes	No	No	No	Yes	Yes	No	Latency
Sun et al. [63]	Crowd-funding algorithm	Fog	A resource-sharing model grounded on the repeated game theory in Fog computing paradigm	Hadoop cluster	Synthetic (Jmeter)	No	No	Yes	No	No	No	SLA violation	No
Aazam et al. [108]	Dynamic resource estimation	Fog/ Cloud	Relinquish probabilities of the customers by considering resource utilization and the service quality experienced	Simulation (CloudSim)	Real IoT traces	Yes	No	No	No	No	No	NA	NA

جدول ۵- خلاصه تخصیص منابع و برنامه ریزی تحقیقات در مه

C. ابزارهای شبیه سازی برای محاسبات Fogg

شبیه سازی و مدل سازی در رایانش مه هنوز در مراحل ابتدایی است. با این حال ، چند کار تحقیقاتی در مورد شبیه سازی رایانش مه انجام شده است ، که بر برخی جنبه های خاص رایانش مه متمرکز شده است. Aazam و هو [54] بر پیش بینی منابع و قیمت گذاری آنها در رایانش مه تمرکز کردند. مدل مدیریت منبع مبتنی بر مه قادر به برآورد منابع مورد نیاز بر اساس احتمال رفتار کاربر در استفاده از منابع در آینده است. اعتبار سنجی و ارزیابی میزان عملکرد با استفاده از شبیه سازی انجام شد. با این حال ، آنها ناهمگنی سرویس ، QoS یا عوامل تحرک دستگاه را در نظر نگرفتند. کار دیگری که توسط دستجردی و همکاران پیشنهاد شده است. [4] بر روی درخواست جستجو برای کشف حادثه در یک مورد استفاده از شهرهای هوشمند متمرکز شده است. هر دوی این آثار از

CloudSim [119] برای اعتبار سنجی روش خود همراه با ارزیابی ارزشی استفاده شده اند. اولین جعبه ابزار برای شبیه سازی مه توسط Gupta و همکاران تهیه شد. [120]، معروف به iFogSim. این جعبه ابزار برای شبیه سازی و مدل سازی تکنیک های مدیریت منابع اینترنت اشیا در پارادایم های محاسبه مه و لبه استفاده می شود. چالش برانگیزترین مشکل، طراحی تکنیکهای کنترل منابع انسانی است که توزیع برنامه تحلیلی را بین دستگاههای لبه تعیین می کند، که باعث بهبود توان عملیاتی و کاهش تأخیر می شود. شبیه ساز پیشنهادی قادر به اندازه گیری تأثیر تکنیک های مدیریت منابع از نظر ازدحام شبکه، تأخیر، هزینه و مصرف انرژی است. شبیه ساز با استفاده از دو مورد استفاده اعتبار سنجی شد و نویسندگان نیز معماری محیط رایانش مه را ارائه دادند. چالش های موجود در استفاده از رایانش مه شامل ترکیب مه با فناوری های نوظهور مانند فناوری های G5 [121]، مجازی سازی عملکرد شبکه (NFV) و شبکه تعریف شده توسط نرم افزار (SDN) است. در این حالت، یک سیم کارت با پشتیبانی از ظرف، SDN و NFV بسیار مهم است. جدول 7 ویژگیهای اصلی این دو شبیه ساز را ارائه می دهد که بیشتر توسط محققان مختلف برای شبیه سازی رایانش مه استفاده می شود. این دو ابزار شبیه سازی بر روی پارامترهای شبکه مانند توزیع پهنای باند لینک و تأخیر رفت و برگشت از رسانه های مختلف تمرکز ندارند. این دو پارامتر به شدت بر نتایج شبیه سازی که در آن به حداقل رساندن تأخیر هدف اصلی در یک محیط رایانش مه است، تأثیر می گذارد. ثانیاً، هر دو ابزار مجازی سازی مبتنی بر کانتینر را در نظر نگرفتند. در یک محیط رایانش مه، دستگاه های زیادی وجود دارد که در محاسبات شرکت می کنند، جایی که مجازی سازی مبتنی بر hypervisor به دلیل کمبود حافظه و قدرت پردازش این دستگاه ها تقریباً غیرممکن است.

D. خدمات میکرو مبتنی بر مه

ریز سرویس فرایندی مستقل و معماری نرم افزار گرا (SOA) است که با ارسال پیام در تعامل است. SOA میکرو سرویس هیچ مانعی برای برنامه نویسی خاص ایجاد نمی کند. این دستورالعمل ها را برای برنامه های توزیع شده برای تقسیم بندی هر یک از اجزا به طور مستقل طراحی و اجرا می کند. هر یک از اجزای یک عملکرد خاص را نشان می دهد. عملکرد مولفه ها با ارسال پیام قابل دسترسی است و امکان پیاده سازی آن در هر زبان اصلی برنامه نویسی به صورت داخلی وجود دارد. به این ترتیب، این اصل به توسعه دهندگان و مدیران پروژه کمک می کند تا هر ماژول را به طور مستقل توسعه دهند و آن را با چند عملکرد مرتبط آزمایش کنند. بعضی از ریز سرویس ها که به آنها ریز

خدمات سطح بالا نیز گفته می شود ، عمدتاً مسئول هماهنگی با سایر ریز خدمات هستند [124]. رویکرد سازمانی میکرو سرویس ها ، چرخه پیشرفت را تسریع می کند ، مالکیت را تغذیه می کند ، نوآوری را تشویق می کند ، مقیاس پذیری را بهبود می بخشد و قابلیت نگهداری برنامه های نرم افزار را افزایش می دهد. با استفاده از این روش ، نرم افزار به یک سرویس مستقل کوچک تبدیل می شود و با API های بدون ابهام تعامل دارد. این خدمات از طریق تیم های کوچک خودگردان حفظ می شوند [125].

چابکی و ماهیت توزیع شده مستقل استقرار میکروسر ، آن را به یک راه حل خوب برای توسعه برنامه IoT مبتنی بر مه تبدیل کرده است. فرآیندهای مستقل و تعامل از طریق ویژگی های ارسال پیام باعث شده است که ریز سرویس ها برای کاربردهای اینترنت اشیا راحت تر باشند. در مه ، محدودیت منابع وجود دارد ، بنابراین توسعه ریز خدمات در مه با تقسیم آن به مجموعه ای از خدمات کوچک مستقل ، پیچیدگی رو به رشد سیستم بزرگ را به حداقل می رساند. میکرو سرویس با ترکیب انسجام بالا و اتصال شل سیستم های توزیع شده ، مدولار را به سطح بعدی می رساند.

1 جنبه های تحقیق فعلی میکرو سرویس

اخیراً ، برنامه های مبتنی بر ریز خدمات شروع به محبوبیت می کنند [124]. ریز سرویس های مبتنی بر مه به طور گسترده مورد بررسی قرار نگرفته اند. از این رو ، این یک منطقه تحقیقاتی باز است. با این حال ، برخی از کارهای تحقیقاتی در این زمینه تحقیقاتی نوظهور انجام شده است ، که بیشتر تلاش ها مربوط به اینترنت اشیا است. بوتزین و همکاران [126] استفاده از ریز سرویس ها را در اینترنت اشیا بررسی کرد و ادعا کرد که هدف معماری اینترنت اشیا و ریز خدمات مشابه است. با این حال ، آنها در واقع از نظر جنبه های مختلف ویژگی های مختلفی دارند. اول از همه ، ریز سرویس دارای یک ویژگی خود مهار است که در آن تمام وابستگی ها و کتابخانه ها با برنامه در یک تصویر واحد بسته بندی می شوند. از سوی دیگر ، برای اینترنت اشیا ، همه کتابخانه ها به طور کلی با برنامه پیچیده نمی شوند. با این وجود ، هر دو از انواع مشابه مجازی سازی و پروتکل های وب استفاده می کنند. میکرو سرویس همچنین دارای یکپارچه سازی و تحویل مداوم است در حالی که در اینترنت اشیا این موارد در دسترس نیستند یا فقط تا حدی وجود دارند.

Vresk و Čavrak [127] برای پشتیبانی از ناهمگنی دستگاه ، پروتکل های مختلف ارتباطی و خدمات ، یک میان افزار مبتنی بر ریز سرویس برای اینترنت اشیا ارائه دادند. آنها یک مدل داده و

مدل آدرس برای IoT مبتنی بر ریز سرویس ارائه دادند. بریتو و همکاران [128] معماری ارکستراسیون سرویس را برای مه با استفاده از سرویس های خرد ارائه داد. نویسندگان مدیر منابع را به عنوان یک سرویس خرد تعریف کردند. Khazaei و همکاران [129] یک سیستم عامل مبتنی بر ریز سرویس با برنامه ریزی عمومی قابل برنامه ریزی برای اینترنت اشیا ارائه داد. در سیستم عامل ، سرویس های خرد در همه لایه ها به روشی آبخار وجود دارد و یک سیستم مدیریت خودمختار ، ریز سرویس را مقیاس بندی می کند. یک نوع مشابه از چارچوب اینترنت اشیا توسط Sun و همکاران ارائه شده است. [130] در ساختار نه مولفه ای آنها ، همه سرویس های ریز هستند به جز سرویس اصلی. کار ثابت کرد که ریز سرویس ها از نظر مقیاس پذیری ، انعطاف پذیری و استقلال سیستم عامل بسیار بهتر از رویکرد یکپارچه هستند. با این حال ، هنوز ، معماری اینترنت اشیا mic مبتنی بر ریز سرویس از مشکلات مختلفی مانند خطاهای موجود در شبکه ، تاخیر شبکه ، سریال سازی پیام ها ، پردازش معاملات مشترک و سایر سناریوهای محاسبات توزیع شده رنج می برد. لی و همکاران [131] یک مدل مبتنی بر همکاری به طور خاص برای دستگاه های سنجش هوشمند ارائه داد. با انجام یک مفهوم مبتنی بر خدمات خرد ، می توان عملکرد چنین خدماتی را افزایش داد. کریویچ و همکاران [132] یک راه حل مدیریتی برای ارتباطات دستگاه ماشین به ماشین (M2M) در سیستم اینترنت اشیا با استفاده از ریز سرویس مشترک ارائه داد. آنها استدلال کردند که ریز سرویس ها می توانند به عنوان عامل در یک سیستم مبتنی بر عامل عمل کنند زیرا هر ریز سرویس وظیفه انجام یک کار خاص را بر عهده دارد و برای دستیابی به هدف سیستم همکاری می کند. مجازی سازی مبتنی بر کانتینر بهترین راه حل برای استقرار ریز سرویس ها است زیرا ظرف از مجازی سازی سیستم عامل پشتیبانی می کند و تمام وابستگی ها را در یک تصویر واحد بسته بندی می کند. یک کانتینر قادر است منابع سخت افزاری فیزیکی مورد نیاز یک برنامه را با برنامه هسته هسته سیستم عامل خود مدیریت کند [133].

2) سرویس های کوچک و برنامه های IOT

بسیاری از کارهای پژوهشی پردازش مبتنی بر مه را برای برنامه های اینترنت اشیا در سیستم های حمل و نقل هوشمند [96] - [98] ، واقعیت افزوده و مجازی [99] ، [134] ، [135] و بهداشت و درمان [26] ، [136] ، [137] رایش مه برای جریان ویدئو ، خانه های هوشمند ، شهرهای هوشمند و CDN نیز مناسب است. ویژگی های مشترک این برنامه ها حساسیت به زمان است که

باعث می شود رایانش مه یک الگوی محاسباتی نوظهور امیدوار کننده باشد. اشکال اصلی مه ، محدودیت منابع و خرابی آن است. بنابراین ، استفاده از سرویس های خرد برای برنامه های اینترنت اشیا-مبتنی بر مه این اشکال را به حداقل می رساند. ریز خدمات مستقل ، سبک وزن و به راحتی قابل تحویل هستند. برای کاهش محدودیت منابع ، ظرف مبتنی بر ریز سرویس بهترین راه حل تاکنون است. به همین ترتیب ، با استقرار سریع برنامه ، هزینه خرابی را نیز به حداقل می رساند. با اجرای خدمات خرد مبتنی بر مه می توان به بسیاری از مسائل تحقیقاتی باز پرداخت. اینها شامل مدیریت خدمات ، برنامه ریزی ، نظارت ، تحمل خطا ، امنیت و حریم خصوصی است.

Author & Year	Types of failure	Failure management	Mechanism employed	Infrastructure
Jiang and Hsu [114]	Hardware failure	Proactive	Two-level standby design	Cloud
Lati et al. [115]	Task failure	Reactive	Checkpointing and reallocation	Cloud
Wu et al. [116]	Host failure	Proactive	Migration	Cloud
Egwutuoha et al. [117]	Hardware /VM/ application failure	Proactive	Checkpointing and migration	Cloud/HPC
Sampaio and Barbosa [88]	Hardware, software and unstable behavior	Proactive	Prediction	Cloud
Gao et al. [120]	Task failure	Proactive and reactive	Crash detection, replication and migration	Cloud

جدول ۶- خلاصه تحقیقات تحمل خطا در ابر

Name of the simulation tool	Proposed by	Key features	Usage example
CloudSim	Calheiros et al. [121]	A broad simulation toolkit that enables simulation and modeling of application provisioning in the cloud computing environments. The CloudSim toolkit supports system modeling of cloud system components such as virtual machines (VMs), data centers and resource allocation and provisioning policies as well as support system behavior modeling.	Aazam and Huh [54]; Dastjerdi et al. [4]
iFogSim	Gupta et al. [122]	Modelled IoT and Fog environments and measure the impact of resource management techniques in terms of network congestion, latency, cost and energy consumption. Developed on top of CloudSim	Baccarelli et al. [9]; Bittencourt et al. [124]; Markakis et al. [125]; Taneja and Davy [106]

جدول ۷- ابزارهای شبیه سازی که برای شبیه سازی مه و ویژگی های اصلی آنها استفاده می شود.

E. محاسبات موبایل مبتنی بر مه

تعداد کاربران تلفن های هوشمند و دستگاه های تلفن همراه در مناطق شهری و همچنین در مناطق روستایی روز به روز در حال افزایش است. در نتیجه ، کاربران موبایل اکنون به طور مشترک مطالب با حجم بالا را درخواست می کنند. ارائه خدمات برای کلیه مطالب درخواستی در منطقه ای که کاربران تلفن همراه در آن بسیار پرجمعیت است برای ارائه دهندگان خدمات کاری واقعاً چالش برانگیز است [138]. تعداد زیاد درخواست های محتوای همزمان فقط وضعیت را بدتر می کند. یکی از بهترین راه حل ها برای کنار آمدن با این مشکل تخلیه محتوای نزدیک به کاربران است ، تا کاربران بتوانند خدمات بهتری دریافت کنند. این فرایند بارگیری محتوا را می توان با استفاده از محاسبات موبایل مه پشتیبانی کرد. در رایانش مه همراه ، محتوا به دستگاه مه که در نزدیکی کاربران قرار دارد بارگیری می شود. با این حال ، مدیریت محتوا در گره های مه یک مسئله تحقیقاتی فعلی است. بسته به تقاضای مطالب ، تخلیه باید در گره های مه توزیع شود. نظارت مداوم و عملکرد کارآمد حافظه پنهان برای مقابله با گره های مه محدود به منابع بسیار مهم است. چند کار تحقیقاتی در مورد رایانش موبایل مه انجام شده است. این بخش به طور تقارنی در مورد کارهای تحقیقاتی که در چند سال گذشته در این زمینه تحقیقاتی خاص انجام شده است ، بحث می کند.

هونگ و همکاران [94] یک مدل برنامه نویسی سطح بالا ارائه می دهد که از برنامه های حساس به زمان توزیع شده از نظر جغرافیایی پشتیبانی می کند. مدل پیشنهادی برنامه نویسی موبایل مه دارای دو هدف طراحی است. هدف اولیه این است که توسعه یک برنامه ساده برای تعداد زیادی از دستگاه های ناهمگن ، که در یک منطقه گسترده توزیع شده است. هدف بعدی مقیاس پویای منابع بر اساس تقاضای منابع است. آنها یک API برای مدل برنامه نویسی خود ایجاد کردند و آن را با استفاده از دو مدل کاربردی ارزیابی کردند: ردیابی وسیله نقلیه با استفاده از دوربین و نظارت بر ترافیک با استفاده از یک سیستم پردازش رویداد پیچیده (MCEP) مبتنی بر تحرک. با این حال ، آنها بر جایگذاری فرآیند یا مهاجرت فرآیند تمرکز نکردند.

شی و همکاران [139] یک مدل ارتباطی الهام گرفته از P2P بین ابر دستگاه تلفن همراه و گره های تلفن همراه را پیشنهاد داد تا منابع و وظیفه محاسبات را در میان دستگاه های تلفن همراه به اشتراک بگذارد. آنها در کار خود از یک پروتکل برنامه محدود (CoAP) برای پیاده سازی ریز خدمات استفاده کردند. این کار رویکرد M2M را در رایانش مه معرفی می کند در حالی که مه کلاسیک

مبتنی بر سلسله مراتب است. تخلیه محتوا در موبایل مه توسط خان و همکاران بررسی شد. [138]

آنها مه موبایل را به عنوان گره های سیار خود سازمان دهنده مستقر تعریف کردند که منابع توزیع شده را در لبه ارائه می دهد. هدف از این کار همکاری گره ها برای ذخیره محتوا بود ، که در دسترس بودن محتوا را به حداکثر می رساند و هزینه عملیاتی را به حداقل می رساند. بازی ائتلاف پیشنهادی به شما کمک می کند تا بهترین نامزدهای مشترک را در نزدیکی کاربران برای به اشتراک گذاری ذخیره سازی و سازماندهی خود قرار دهید.

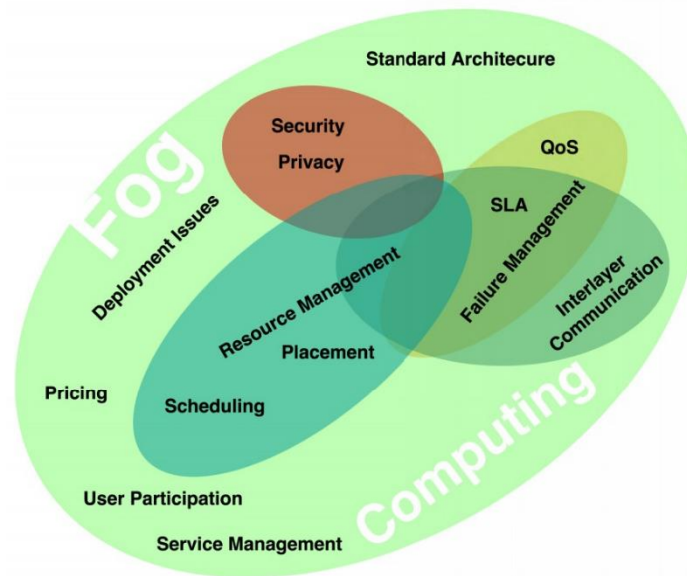
وانگ و همکاران [140] یک چارچوب سلسله مراتبی سه لایه را با استفاده از یک ساختار مه برای ایجاد ارتباط بین شبکه های WSN و ابر پیشنهاد داد. آنها یک الگوریتم مسیریابی را برای پل زدن ارتباطات با در نظر گرفتن تعداد هاپ و مصرف انرژی طراحی کردند. آنها گره مه را به عنوان یک سینک تعریف کردند ، که داده ها را از سنسور به ابر منتقل می کند. چارچوب پیشنهادی شامل لایه مسیریابی ، لایه مه و لایه غرق می شود. در لایه مه ، یک سینک مانند گره های مه نیز عمل می کند تا تأخیر انتقال را به حداقل برساند. با این وجود ، در محاسبات موبایل مه ، نگرانی امنیتی و حریم خصوصی وجود ندارد. رومن و همکاران [141] امنیت و حریم خصوصی همه محاسبات سطح لبه را مورد خطاب قرار می دهد. این شامل یک موضوع معمول در سیستم تلفن همراه مه است که به اقدامات احراز هویت ، اعتماد ، کنترل دسترسی ، پروتکل و امنیت شبکه نیاز دارد.

۹- مشکلات و سمت و سوی تحقیقات آینده

A. مسائل مرتبط با زیرساخت ها

مه یک فناوری در حال تکامل است ، به گونه ای گسترش می یابد که برای پشتیبانی از انواع برنامه های حساس به زمان نیاز به دستیابی به بازار دارد. مه به تصویب تلاش های تحقیقاتی آکادمی ها و صنایع مختلف تبدیل شده است. یکی از اقدامات اساسی کنسرسیوم (OpenFog) است که توسط ARM، Cisco، Dell، Intel، Microsoft و دانشگاه پرینستون در نوامبر 2015 تأسیس شد [31]. فاکسکان ، جنرال الکتریک ، هیتاچی ، اینترنت ساکورا ، دانشگاه شانگهای تک و ZTE اعضای مof در این کنسرسیوم غیرانتفاعی هستند. آنها با تلفیق فناوری بی سیم G5 ، اینترنت اشیا و هوش مصنوعی جاسازی شده با ارائه معماری باز و قابل اجرا ، در حال نوآوری دیجیتال هستند. با این حال ، بسیاری از چالش های باز برای این الگوی محاسبات در سطح جوانه وجود دارد. در این بخش ، ما

در مورد چالش های تحقیق بحث می کنیم و مسیرهای آینده تحقیقات رایانش مه را بررسی می کنیم. شکل 10 برخی از مسائل مهم تحقیقاتی در رایانش مه را نشان می دهد.



شکل ۱۰- مشکلات رایانش مه

1) مسائل مربوط به توسعه

از نظر توسعه ، OpenFog به عنوان یک محیط N-tier تعریف می شود. با این حال ، افزایش بیش از حد در تعداد سطوح در لایه مه ممکن است باعث ایجاد مشکلات تاخیر در الگوی رایانش مه تازه ظهور شده شود. بنابراین ، تعداد طبقات بر اساس مورد استفاده باید تعیین شود. تصمیمات استقرار بر اساس الزاماتی از قبیل نوع و میزان وظیفه ای که توسط هر ردیف ، تعداد کل سنسورها ، قابلیت دستگاه Fog ، بین زمان تاخیر و قابلیت اطمینان دستگاه Fog انجام می شود ، اتخاذ می شود. هنوز هم لازم است که چگونگی تحقق این الزامات بررسی شود. کاربرد و مقیاس گذاری منابع نیز مسئله مهمی در هنگام استقرار است. بر اساس نیاز برنامه و منبع ، مقیاس گذاری و کوچک سازی بدون قطع خدمات فعلی می تواند انجام شود. در این راستا ، ممکن است استقرار در استقرار مه تأثیر بگذارد.

2) معماری استاندارد برای رایانش مه

تاکنون هیچ معماری استاندارد مشخصی برای رایانش مه وجود نداشته است. کنسرسیوم OpenFog در فوریه 2016 [61] و فوریه 2017 [142] دو نسخه از معماری مه را منتشر کرد. اولین پیش نویس آنها مروری اولیه بر معماری مه بود. در پیش نویس دوم آنها، سخنرانی معماری مه با جزئیات بیشتری مورد بحث قرار گرفت. در معماری پیشنهادی خود، آنها بسیاری از جنبه های اصلی معماری مه از جمله عملکرد، مدیریت، امنیت، تحلیل داده ها و کنترل را در نظر گرفتند. با این حال، برای بررسی و کسب بینش عمیق تر در هر لایه با اعتبار سنجی مناسب، باید تحقیقات بیشتری انجام شود.

3) قابلیت همکاری و فدراسیون برای مه

به دلیل مه، کاربران می توانند درخواست خود را در نزدیکی خود پردازش کنند، که این امر تأخیر را به حداقل می رساند. با این حال، اگر تعداد فزاینده ای از چندین درخواست برنامه آگاه از تأخیر در یک عکس به دستگاه مه ارسال شود و دستگاه مه قادر به رسیدگی به این تعداد درخواست نباشد، چه اتفاقی می افتد؟ آیا برای پردازش به ابر منتقل می شود؟ اگر به ابر منتقل شود، شرایط تأخیر تأمین نمی شود. بنابراین، قابلیت همکاری و فدراسیون در میان خوشه های مه و سرورهای مه ضروری است. از این رو، اگر یک دستگاه مه کاملاً مورد استفاده قرار بگیرد، به جای ارسال آنها به ابر، درخواست ها را برای پردازش دستگاه های مه یا سرورهای مه ارسال می کند.

B. مسائل مرتبط با پلتفرم

1) مدیریت منابع

منابع در محیط مه به دلیل تنوع دستگاه ها و منابع موجود، بسیار پویا و ناهمگن هستند. همه دستگاه هایی که به عنوان دستگاه های مه شناخته می شوند وظیفه انجام محاسبه برنامه خود را دارند. به عنوان مثال، رایانه ای که برای انجام برخی کارهای معمولی ایمیل و مستندات به کارمندان دفتر کار متکی است ممکن است بخشی از مه باشد و همچنین ممکن است به عنوان یک دستگاه مه بازی کند. در چنین شرایطی، مقدار منابع موجود برای محاسبه مه پویا است اما از طریق تجزیه و تحلیل فعالیت طولانی مدت منابع آن قابل پیش بینی است. این پیش بینی ضروری است زیرا به محض شروع اجرای کار مه و در طی یک دوره زمانی، ممکن است وضعیت منابع به دلیل درخواست برنامه ای که دستگاه مسئول آن است، تغییر کند. اگر این مورد را با ابر مقایسه کنیم، می توان دانست که چه مقدار منابع در حال حاضر در دسترس است و آیا آنها منحصراً برای درخواست های برنامه مبتنی بر

ابر استفاده می شوند یا خیر. با این حال ، هدف مه استفاده از منابع غیرفعال موجود در هر دستگاه مه با رایانش مه است که همیشه اولویت دوم را دارد. از این رو ، تخصیص و برنامه ریزی منابع در مه نسبت به تخصیص و برنامه ریزی منابع سنتی در ابر چالش برانگیزتر است.

(2) مدیریت عدم موفقیت

احتمال خرابی دستگاه مه همیشه زیاد است زیرا دستگاه ها توزیع می شوند و مدیریت دستگاه های مه مرکزی نیست. از این رو ، دستگاه ها ممکن است برای بسیاری از پسران خراب شوند. این می تواند به دلیل خرابی سخت افزار ، خرابی نرم افزار یا به دلیل فعالیت کاربر باشد. علاوه بر این مشکلات ، برخی از دلایل دیگر شامل اتصال ، تحرک و منبع تغذیه است که آنها نیز نقش زیادی دارند. اکثر دستگاه ها در یک محیط مه ممکن است از طریق اتصالات بی سیم متصل شوند. بدیهی است که اتصالات بی سیم همیشه قابل اعتماد نیستند. اکثر دستگاه هایی که از طریق بی سیم متصل هستند تلفن همراه هستند ، بنابراین این دستگاه ها می توانند مکان را به طور منظم به خوشه های مختلف تغییر دهند. یکی دیگر از ویژگی های دیگر این دستگاه ها این است که از باتری استفاده می کنند و ممکن است در هر زمان خراب شوند. از این رو ، مقابله با ماهیت پیچیده شکست بسیار دشوار است. همچنین ، اطمینان از SLA با تعریف پارامترهای QoS ضروری است. بنابراین ، SIs این است: SLA ها چیست و چگونه باید تعریف شوند؟ همچنین: چه پارامترهای QoS باید در نظر گرفته شود ، تا مصرف کننده و ارائه دهندگان بتوانند وضعیت برنده را حفظ کنند؟

(3) ارتباط بین لایه های مختلف

برای اطمینان از برآورده شدن نیازهای برنامه در برنامه های حساس به زمان ، مه باید از اتصال بی وقفه با دستگاه ها اطمینان حاصل کند. اگر این برنامه برای کنترل اتومبیل یا هواپیمای بدون سرنشین مستقل باشد و اگر مسئولیت نظارت اضطراری را بر عهده داشته باشد ، شکست در اتصال ممکن است آسیب جدی ببیند. حتی اگر اتصال به ابر از کار بیفتد ، هنوز هم مه باید اتصال پیوسته را تضمین کند. بنابراین ، اتصال لایه ای بین دستگاه های اینترنت اشیا ، مه و ابر از اهمیت بالایی برخوردار است. نوع اتصال و پروتکل های استفاده شده توسط دستگاه های اینترنت اشیا و دستگاه های مه ممکن است متفاوت باشد. بنابراین ، چگونگی رسیدگی به این مسائل یک مسئله مهم تحقیقاتی است.

4) مدیریت مشارکت کاربر

مدیریت خدمات کارآمد مه به مشارکت کاربران در محاسبه مه بستگی دارد. با این حال ، چگونه می توان مشارکت کاربر را مدیریت کرد؟ چگونه در مواردی که هیچ کس نمی خواهد در آن شرکت کند حداقل منابع را مستقر کنیم؟ ما باید این مشکلات را به روشنی با یک راه حل عملی مرتفع کنیم. یکی از روش های افزایش مشارکت کاربران ، سیاست های تشویقی و پاداش محور است. با چنین سیاست هایی ، هر کاربری که در محاسبه مه شرکت کند ، سود خواهد برد. حتی یک کاربر که برای تکمیل درخواست خود شرکت می کند ، بر اساس مشارکت خود با دریافت برخی تخفیف ها پاداش می گیرد. با این حال ، این منطقه باید مورد توجه قرار گیرد زیرا موفقیت کلی در محاسبه مه به مشارکت دستگاه های مه ، که متعلق به افراد و سازمان های مختلف است ، بستگی دارد.

5) امنیت و حریم خصوصی

دستگاه های مه بر اساس موقعیت و مالکیت آنها توسط اپراتورهای مختلف مدیریت می شوند. اگر کنترل دستگاه به خطر بیفتد ، هیچ کس نمی خواهد در محاسبه مه کمک کند. بنابراین ، اگر دستگاه بخواهد در رایانش مه شرکت کند ، امنیت یک دستگاه شرکت کننده چگونه حفظ خواهد شد ، یک سوال بزرگ است. یکی دیگر از مشکلات اساسی امنیتی این سناریو ، امنیت داده های کاربر کاربر است. دستگاه شرکت کننده ممکن است دارای اطلاعات حیاتی باشد. چگونه ایمنی در چنین شرایطی تضمین می شود؟ از طرف دیگر ، داده های مهم ممکن است در دستگاهی پردازش شوند که متعلق به یک هکر کلاه سیاه باشد. چگونه امنیت و حریم خصوصی چگونه تضمین می شود؟ مسائل امنیتی نیز در طی ارتباطات لایه ای وجود دارد. مشابه ماهیت توزیع شده مه ، مدیریت امنیت نیز باید توزیع شود که به هیچ یک از اجزای اصلی وابسته نخواهد بود.

C. مسائل مربوط به برنامه

1) مدیریت خدمات برنامه

با استفاده از پارادایم مه ، میلیارد ها دستگاه اینترنت اشیا اداره خواهند شد ، این برنامه ها از برنامه های حساس به زمان و حساس به زمان استفاده خواهند کرد. درجه سرویس ، در دسترس بودن و کیفیت متنوع ترین در مه است. از این رو ، مدیریت خدمات یک مسئله معمول برای کل قلمرو مه

است. خدمات باید مبتنی بر ریز سرویس باشند ، به طوری که چابکی و مسائل مدیریتی به درستی قابل رسیدگی هستند. تحقیقات بیشتر برای کشف امکان راه حل های مبتنی بر مه لازم است.

2) مدل سازی برنامه

مدل سازی برنامه های مه پیچیده است زیرا این برنامه باید داده ها را از دستگاه های مختلف اینترنت اشیا، که از پروتکل ها و مجموعه ای از کد های مختلف استفاده می کنند ، جمع آوری کند. بنابراین ، مدل سازی برنامه های کاربردی عمومی ، که می توانند با حداقل تلاش استفاده شوند ، چالش برانگیز است. برای حل این مسئله ، یک فرم استاندارد از پروتکل ارتباطی لازم است ، بنابراین برنامه مدل شده می تواند با انواع مختلف دستگاه های اینترنت اشیا ارتباط برقرار کند و کار کند.

۱۰- نتیجه گیری

الگوی رایانش مه در حال حاضر در مراحل ابتدایی است ، بنابراین تحقیقات گسترده ای برای این فناوری در حال ظهور مورد نیاز است. در این بررسی ، ما نمای کلی ، معماری ، پیشرفته ترین و سایر فن آوری های مشابه را در رایانش مه ارائه و بحث کردیم. بر اساس ادبیات ، با تجزیه و تحلیل نیاز زیرساخت ها ، سیستم عامل و برنامه های مه ، طبقه بندی رایانش مه را بدست آوردیم. ما همچنین تخصیص و برنامه ریزی منابع ، تحمل خطا ، ابزارهای شبیه سازی و ریز خدمات را در رایانش مه تحت پوشش قرار دادیم. سرانجام ، ما برخی از موضوعات تحقیقاتی چالش برانگیز و باز را ارائه دادیم. ما قویاً معتقدیم که این بررسی جامع ، اجرای برنامه اینترنت اشیا را برای یک محیط رایانش مه و همچنین جهت گیری تحقیقات فعلی و آینده در این منطقه تحقیقاتی که به سرعت در حال رشد است ، نشان می دهد. به این ترتیب ، این الگوی محاسباتی که هنوز نابالغ است ، در آینده نزدیک به سمت دستیابی به بازار پیش خواهد رفت.

سپاسگذاری

نویسندگان صمیمانه از پروفیسور راجومار بوییا برای اظهار نظرها و بحث های بصیرت آمیز در مورد بهبود کیفیت کلی مقاله تشکر میکنند.

- [1] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. S. Netto, and R. Buyya, "Big data computing and clouds: Trends and future directions," *J. Parallel Distrib. Comput.*, vols. 79–80, pp. 3–15, May 2015.
- [2] J. Chen et al., "Big data challenge: A data management perspective," *Frontiers Comput. Sci.*, vol. 7, no. 2, pp. 157–164, 2013.
- [3] F. Alhaddadin, W. Liu, and J. A. Gutiérrez, "A user profile-aware policy-based management framework for greening the cloud," in *Proc. IEEE 4th Int. Conf. Big Data Cloud Comput. (BdCloud)*, Dec. 2014, pp. 682–687.
- [4] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principles, architectures, and applications," in *Internet of Things: Principle & Paradigms*. San Mateo, CA, USA: Morgan Kaufmann, 2016.
- [5] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything*. Singapore: Springer, 2018, pp. 103–130.
- [6] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "FogRoute: DTN-based data dissemination model in fog computing," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 225–235, Feb. 2017.
- [7] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.
- [8] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: Architecture, key technologies, applications and open issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, Nov. 2017.
- [9] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, "Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study," *IEEE Access*, vol. 5, pp. 9882–9910, 2017.
- [10] P. Varshney and Y. Simmhan, "Demystifying fog computing: Characterizing architectures, applications and abstractions," in *Proc. IEEE 1st Int. Conf. Fog Edge Comput. (ICFEC)*, Feb. 2017, pp. 115–124.
- [11] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Surv.*, vol. 50, no. 3, p. 32, Oct. 2017.
- [12] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art

- and research challenges,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.
- [13] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of Things,” in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.
- [14] X. Xie, H.-J. Zeng, and W.-Y. Ma, “Enabling personalization services on the edge,” in *Proc. 10th ACM Int. Conf. Multimedia*, 2002, pp. 263–266.
- [15] P. P. Gelsinger, “Microprocessors for the new millennium: Challenges, opportunities, and new frontiers,” in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2001, pp. 22–25.
- [16] S. Ibrahim, H. Jin, B. Cheng, H. Cao, S. Wu, and L. Qi, “CLOUDLET: Towards mapreduce implementation on virtual machines,” in *Proc. 18th ACM Int. Symp. High Perform. Distrib. Comput.*, 2009, pp. 65–66.
- [17] N. M. Gonzalez et al., “Fog computing: Data analytics and cloud distributed processing on the network edges,” in *Proc. 35th Int. Conf. Chilean Comput. Sci. Soc. (SCCC)*, Oct. 2016, pp. 1–9.
- [18] J. Li, T. Zhang, J. Jin, Y. Yang, D. Yuan, and L. Gao, “Latency estimation for fog-based Internet of Things,” in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2017, pp. 1–6.
- [19] “Fog computing and the Internet of Things: Extend the cloud to where the things are,” Cisco, San Jose, CA, USA, White Paper, 2015.
[Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/./computing-overview.pdf
- [20] L. M. Vaquero and L. Roderio-Merino, “Finding your way in the fog: Towards a comprehensive definition of fog computing,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.
- [21] (Sep. 2016). What is Fog Computing? [Online]. Available: <https://www.ibm.com/blogs/cloud-computing/2014/08/fog-computing/>
- [22] Cisco Systems. (2016). Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. [Online]. Available: http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computingoverview.pdf
- [23] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
- [24] (Aug. 2017). Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017. [Online]. Available: <http://www.gartner.com/>

smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emergingtechnologies-2017/

- [25] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generat. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.
- [26] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-fog interoperability in IoT-enabled healthcare solutions," in *Proc. 19th Int. Conf. Distrib. Comput. Netw. (ICDCN)*. New York, NY, USA: ACM, 2018, pp. 32:1–32:10.
- [27] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun. (Mar. 2016). "Fog computing: Focusing on mobile users at the edge." pp. 1–11. [Online]. Available: <https://arxiv.org/abs/1502.01815>
- [28] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 3, Jul. 2015, pp. 53–59.
- [29] M. H. Syed, E. B. Fernandez, and M. Ilyas, "A pattern for fog computing," in *Proc. 10th Travelling Conf. Pattern Lang. Programs*, 2016, p. 13.
- [30] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.
- [31] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
- [32] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 369–392, Feb. 2014.
- [33] P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *Proc. 3rd ACM Workshop Mobile Cloud Comput. Services*, 2012, pp. 21–28.
- [34] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, "The role of cloudlets in hostile environments," *IEEE Pervas. Comput.*, vol. 12, no. 4, pp. 40–49, Oct. 2013.
- [35] M. T. Beck, M. Werner, S. Feld, and T. Schimper, "Mobile edge computing: A taxonomy," in *Proc. 6th Int. Conf. Adv. Future Internet*, 2014, pp. 48–54.

- [36] G. I. Klas. (2015). Fog Computing and Mobile Edge Cloud Gain Momentum Open Fog Consortium ETSI MEC and Cloudlets. [Online]. Available: <http://yucianga.info/?p=938>
- [37] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.
- [38] Y. Wang, "Cloud-dew architecture," *Int. J. Cloud Comput.*, vol. 4, no. 3, pp. 199–210, 2015.
- [39] K. Skala, D. Davidović, E. Afgan, I. Sović, and Z. Šojat, "Scalable distributed computing hierarchy: Cloud, fog and dew computing," *Open J. Cloud Comput.*, vol. 2, no. 1, pp. 16–24, Mar. 2015.
- [40] Y. Wang and Y. Pan, "Cloud-dew architecture: Realizing the potential of distributed database systems in unreliable networks," in *Proc. Int. Conf. Parallel Distrib. Process. Technol. Appl. (PDPTA)*, 2015, p. 85.
- [41] T. Mane. (2017). Fog-Dew Architecture for Better Consistency. [Online]. Available: <https://eye3i.wordpress.com/2016/07/02/adressinginconsistency-in-dew-computing-using-fog-computing/>
- [42] (2018). Akamai Mobile Cloud Computing. [Online]. Available: <https://www.akamai.com/us/en/resources/mobile-cloud-computing.jsp>
- [43] (2018). EC-IoT Solution. [Online]. Available: <http://e.huawei.com/en/solutions/technical/sdn/agile-iot>
- [44] (Mar. 2016). Cisco IOx Local Manager Pages and Options. http://www.cisco.com/c/en/us/td/docs/routers/access/800/software/guides/iox/lm/reference-guide/1-0/iox_local_manager_ref_guide/ui_reference.html
- [45] (2018). Hyrax Crowd-Sourcing Mobile Devices to Develop Edge Clouds. [Online]. Available: <http://hyrax.dcc.fc.up.pt/>
- [46] (2018). Open-RAN Mobile Edge Computing Platform. [Online]. Available: <https://www.sdxcentral.com/products/saguna-open-ranmobile-edge-computing-platform/>
- [47] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, "Fog data: Enhancing telehealth big data through fog computing," in *Proc. ASE BigData SocialInform.*, 2015, p. 14.
- [48] M. Ahmad, M. Bilal, S. Hussain, B. Ho, T. Cheong, and S. Lee, "Health fog: A novel framework for health and wellness applications," *J. Supercomput.*, vol. 72, no. 10, pp. 3677–3695, 2016.

- [49] B. Tang et al., "Incorporating intelligence in fog computing for big data analysis in smart cities," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2140–2150, Oct. 2017.
- [50] B. Tang, Z. Chen, G. Heffernan, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proc. ASE BigData SocialInform.*, 2015, p. 28.
- [51] W. Zhang, Z. Zhang, and H.-C. Chao, "Cooperative fog computing for dealing with big data in the Internet of vehicles: Architecture and hierarchical resource management," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 60–67, Dec. 2017.
- [52] B. Yin, W. Shen, Y. Cheng, L. X. Cai, and Q. Li, "Distributed resource sharing in fog-assisted big data streaming," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–6.
- [53] R. Pecori, "A virtual learning architecture enhanced by fog computing and big data streams," *Future Internet*, vol. 10, no. 1, pp. 1–30, 2018.
- [54] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proc. 29th IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2015, pp. 687–694.
- [55] H. R. Arkian, A. Diyanat, and A. Pourkhalili, "MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications," *J. Netw. Comput. Appl.*, vol. 82, pp. 152–165, Mar. 2017.
- [56] M. A. Nadeem and M. A. Saeed, "Fog computing: An emerging paradigm," in *Proc. 6th Int. Conf. Innov. Comput. Technol. (INTECH)*, Aug. 2016, pp. 83–86.
- [57] M. Taneja and A. Davy, "Resource aware placement of data analytics platform in fog computing," *Procedia Comput. Sci.*, vol. 97, pp. 153–156, Jan. 2016.
- [58] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Netw.*, vol. 5, no. 2, pp. 23–29, 2016.
- [59] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung, "Developing IoT applications in the fog: A distributed dataflow approach," in *Proc. 5th Int. Conf. Internet Things (IOT)*, Oct. 2015, pp. 155–162.
- [60] F. Hosseinpour, J. Plosila, and H. Tenhunen, "An approach for smart management of big data in the fog computing context," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2016, pp. 468–471.

- [61] OpenFog Consortium Architecture Working Group, "OpenFog architecture overview," OpenFog Consortium, Tokyo, Japan, White Paper OPFWP001.0216, Feb. 2016.
- [62] K. Intharawijitr, K. Iida, and H. Koga, "Analysis of fog model considering computing and communication latency in 5G cellular networks," in Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops), Mar. 2016, pp. 1–4.
- [63] Y. Sun and N. Zhang, "A resource-sharing model based on a repeated game in fog computing," Saudi J. Biol. Sci., vol. 24, no. 3, pp. 687–694, 2017.
- [64] A. Munir, P. Kansakar, and S. U. Khan, "IFCloT: Integrated fog cloud IoT: A novel architectural paradigm for the future Internet of Things," IEEE Consum. Electron. Mag., vol. 6, no. 3, pp. 74–82, Jul. 2017.
- [65] (Dec. 2016). Cisco Fog Data Services. [Online]. Available: <http://www.cisco.com/c/en/us/products/cloud-systems-management/fog-dataservices/index.html>
- [66] (Jan. 2017). Cisco Fog Director. [Online]. Available: <http://www.cisco.com/c/en/us/products/cloud-systems-management/fog-director/index.html>
- [67] G. Albeanu and F. Popentiu-Vladicescu, "A reliable e-learning architecture based on fog-computing and smart devices," in Proc. Int. Sci. Conf. eLearn. Softw. Edu., vol. 4, 2014, p. 9.
- [68] Metro Network Traffic Growth, "An architecture impact study," Alcatel-Lucent Bell Labs, Murray Hill, NJ, USA, White Paper NP2013113265EN, 2013.
- [69] S. Grehl et al., "Research perspective-mobile robots in underground mining," in Proc. AusIMM Bull., Feb. 2017, p. 44.
- [70] V. Turner. (2016). IDC Link: IBM and Cisco Bridge the IoT Edge and Enterprise. pp. 1–2. [Online]. Available: <https://www01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WWL12359USEN>
- [71] J. Biswas, F. Naumann, and Q. Qiu, "Assessing the completeness of sensor data," in Proc. Int. Conf. Database Syst. Adv. Appl. Berlin, Germany: Springer, 2006, pp. 717–732.

- [72] M. Kocakulak and I. Butun, "An overview of Wireless Sensor Networks towards Internet of Things," in Proc. 7th IEEE Annu. Comput. Commun. Workshop Conf. (CCWC), Jan. 2017, pp. 1–6.
- [73] K. Aziz, S. Tarapiah, S. H. Ismail, and S. Atalla, "Smart real-time healthcare monitoring and tracking system using GSM/GPS technologies," in Proc. 3rd MEC Int. Conf. Big Data Smart City (ICBDSC), Mar. 2016, pp. 1–7.
- [74] A. H. Butt, A. Moschetti, L. Fiorini, P. Dario, and F. Cavallo, "Wearable sensors for gesture analysis in smart healthcare applications," in Human Monitoring, Smart Health and Assisted Living: Techniques and Technologies. Rijeka, Croatia: InTech, 2017, p. 79.
- [75] N. Constant, D. Borthakur, M. Abtahi, H. Dubey, and K. Mankodiya. (2017). "Fog-assisted wIoT: A smart fog gateway for end-to-end analytics in wearable Internet of Things." [Online]. Available: <https://arxiv.org/abs/1701.08680>
- [76] B. Kang, D. Kim, and H. Choo, "Internet of everything: A large-scale autonomic IoT gateway," IEEE Trans. Multi-Scale Comput. Syst., vol. 3, no. 3, pp. 206–214, Jul./Sep. 2017.
- [77] S. Guoqiang, C. Yanming, Z. Chao, and Z. Yanxu, "Design and implementation of a smart IoT gateway," in Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Social Comput., Aug. 2013, pp. 720–723.
- [78] Z. Amin, H. Singh, and N. Sethi, "Review on fault tolerance techniques in cloud computing," Int. J. Comput. Appl., vol. 116, no. 18, pp. 1–17, 2015.
- [79] D. Poola, S. K. Garg, R. Buyya, Y. Yang, and K. Ramamohanarao, "Robust scheduling of scientific workflows with deadline and budget constraints in clouds," in Proc. 28th IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA), May 2014, pp. 858–865.
- [80] S. Paul, R. Jain, M. Samaka, and J. Pan, "Application delivery in multicloud environments using software defined networking," Comput. Netw., vol. 68, pp. 166–186, Aug. 2014.
- [81] R. Jhawar, V. Piuri, and M. Santambrogio, "A comprehensive conceptual

- system-level approach to fault tolerance in cloud computing,” in Proc. IEEE Int. Syst. Conf. (SysCon), Mar. 2012, pp. 1–5.
- [82] W. Zhao, P. Melliar-Smith, and L. E. Moser, “Fault tolerance middleware for cloud computing,” in Proc. 3rd IEEE Int. Conf. Cloud Comput. (CLOUD), Jul. 2010, pp. 67–74.
- [83] R. Jhawar, V. Piuri, and M. Santambrogio, “Fault tolerance management in cloud computing: A system-level perspective,” IEEE Syst. J., vol. 7, no. 2, pp. 288–297, Jun. 2013.
- [84] C.-T. Yang, J.-C. Liu, C.-H. Hsu, and W.-L. Chou, “On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism,” J. Supercomput., vol. 69, no. 3, pp. 1103–1122, 2014.
- [85] J. Liu, J. Zhou, and R. Buyya, “Software rejuvenation based fault tolerance scheme for cloud applications,” in Proc. 8th IEEE Int. Conf. Cloud Comput. (CLOUD), Jun./Jul. 2015, pp. 1115–1118.
- [86] B. Mohammed, M. Kiran, K. M. Maiyama, M. M. Kamala, and I.-U. Awan, “Failover strategy for fault tolerance in cloud computing environment,” Softw., Pract. Exper., vol. 47, no. 9, pp. 1243–1274, 2017.
- [87] J. Liu, S. Wang, A. Zhou, S. Kumar, F. Yang, and R. Buyya, “Using proactive fault-tolerance approach to enhance cloud service reliability,” IEEE Trans. Cloud Comput., to be published.
- [88] A. M. Sampaio and J. G. Barbosa, “A comparative cost study of faulttolerant techniques for availability on the cloud,” in Proc. Int. Symp. Ambient Intell. Springer, 2017, pp. 263–268.
- [89] H.-W. Kim, G. Yi, J. H. Park, and Y.-S. Jeong, “Adaptive resource management using many-core processing for fault tolerance based on cyber–physical cloud systems,” Future Generat. Comput. Syst., to be published.
- [90] H. Madsen, B. Burtschy, G. Albeanu, and F. Popentiu-Vladicescu, “Reliability in the utility computing era: Towards reliable fog computing,” in Proc. 20th Int. Conf. Syst., Signals Image Process. (IWSSIP), Jul. 2013, pp. 43–46.
- [91] E. Aghasian, S. Garg, L. Gao, S. Yu, and J. Montgomery, “Scoring users’

- privacy disclosure across multiple online social networks,” IEEE Access, vol. 5, pp. 13118–13130, 2017.
- [92] (Mar. 2018). Here’s Everything You Need to Know About the Cambridge Analytica Scandal. [Online]. Available: <https://www.cnn.com/2018/03/21/facebook-cambridge-analytica-scandal-everything-you-need-to-know.html>
- [93] F. Li, M. Vöegler, M. Claeßens, and S. Dustdar, “Efficient and scalable IoT service delivery on cloud,” in Proc. IEEE 6th Int. Conf. Cloud Comput. (CLOUD), Jun./Jul. 2013, pp. 740–747.
- [94] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, “Mobile fog: A programming model for large-scale applications on the Internet of Things,” in Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput., 2013, pp. 15–20.
- [95] A. V. Dastjerdi and R. Buyya, “Fog computing: Helping the Internet of Things realize its potential,” Computer, vol. 49, no. 8, pp. 112–116, Aug. 2016.
- [96] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, “Software defined networking-based vehicular adhoc network with fog computing,” in Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM), May 2015, pp. 1202–1207.
- [97] N. K. Giang, V. C. M. Leung, and R. Lea, “On developing smart transportation applications in fog computing paradigm,” in Proc. 6th ACM Symp. Develop. Anal. Intell. Veh. Netw. Appl., 2016, pp. 91–98.
- [98] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular fog computing: A viewpoint of vehicles as the infrastructures,” IEEE Trans. Veh. Technol., vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
- [99] J. K. Zao et al., “Augmented brain computer interaction based on fog computing and linked data,” in Proc. Int. Conf. Intell. Environ. (IE), Jun./Jul. 2014, pp. 374–377.
- [100] A. M. Rahmani et al., “Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach,” Future Gener. Comput. Syst., vol. 78, pp. 641–658, Jan. 2018.
- [101] A. Giordano, G. Spezzano, and A. Vinci, “Smart agents and fog computing for smart city applications,” in Proc. Int. Conf. Smart Cities. London,

U.K.: Springer, 2016, pp. 137–146.

- [102] A. A. Alsaffar, H. P. Pham, C.-S. Hong, E.-N. Huh, and M. Aazam, “An architecture of iot service delegation and resource allocation based on collaboration between fog and cloud computing,” *Mobile Inf. Syst.*, vol. 2016, Aug. 2016, Art. no. 6123234.
- [103] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, “Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.
- [104] A. Brogi, S. Forti, and A. Ibrahim, “How to best deploy your fog applications, probably,” in *Proc. Int. Conf. Edge Fog Comput.*, May 2017, pp. 105–114.
- [105] M. Taneja and A. Davy, “Resource aware placement of IoT application modules in fog-cloud computing paradigm,” in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 1222–1228.
- [106] M. Aazam, M. St-Hilaire, C.-H. Lung, I. Lambadaris, and E.-N. Huh, “IoT resource estimation challenges and modeling in fog,” in *Fog Computing in the Internet of Things*. Springer, 2018, pp. 17–31.
- [107] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, “Resource allocation strategy in fog computing based on priced timed petri nets,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1216–1228, 2017.
- [108] Z. Pooranian, M. Shojafar, P. G. V. Naranjo, L. Chiaraviglio, and M. Conti, “A novel distributed fog-based networked architecture to preserve energy in fog data centers,” in *Proc. 2017 IEEE 14th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Orlando, FL, USA, 2017, pp. 22–25.
- [109] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, “Analytic modeling of multitier Internet applications,” *ACM Trans. Web.*, vol. 1, no. 1, p. 2, 2007.
- [110] Y. Sharma, B. Javadi, W. Si, and D. Sun, “Reliability and energy efficiency in cloud computing systems: Survey and taxonomy,” *J. Netw. Comput. Appl.*, vol. 74, pp. 66–85, Oct. 2016.
- [111] Y. Liu, J. E. Fieldsend, and G. Min, “A framework of fog computing: Architecture, challenges, and optimization,” *IEEE Access*, vol. 5, pp. 25445–25454, 2017.

- [112] F.-C. Jiang and C.-H. Hsu, "Fault-tolerant system design on cloud logistics by greener standbys deployment with Petri net model," *Neurocomputing*, vol. 256, pp. 90–100, Sep. 2017.
- [113] M. S. A. Latiff et al., "A checkpointed league championship algorithmbased cloud scheduling scheme with secure fault tolerance responsiveness," *Appl. Soft Comput.*, vol. 61, pp. 670–680, Dec. 2017.
- [114] Y. Wu, H. Song, Y. Xiong, Z. Zheng, Y. Zhang, and G. Huang, "Model defined fault tolerance in cloud," in *Proc. 6th Asia–Pacific Symp. Internetwork Internetwork*, 2014, pp. 116–119.
- [115] I. P. Egwuotuoha, S. Chen, D. Levy, B. Selic, and R. Calvo, "Energy efficient fault tolerance for high performance computing (HPC) in the cloud," in *Proc. IEEE 6th Int. Conf. Cloud Comput. (CLOUD)*, Jun./Jul. 2013, pp. 762–769.
- [116] S. Fu, "Failure-aware resource management for high-availability computing clusters with distributed virtual machines," *J. Parallel Distrib. Comput.*, vol. 70, no. 4, pp. 384–393, 2010.
- [117] A. M. Sampaio and J. G. Barbosa, "Towards high-available and energyefficient virtual computing environments in the cloud," *Future Gener. Comput. Syst.*, vol. 40, pp. 30–43, Nov. 2014.
- [118] Y. Gao, S. K. Gupta, Y. Wang, and M. Pedram, "An energy-aware fault tolerant scheduling framework for soft error resilient cloud computing systems," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2014, pp. 1–6.
- [119] R. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.
- [120] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in Internet of Things, edge and fog computing environments," *Softw., Pract. Exper.*, vol. 47, no. 9, pp. 1275–1296, 2017.
- [121] L. Gao, T. H. Luan, B. Liu, W. Zhou, and S. Yu, "Fog computing and its applications in 5G," in *Proc. 5G Mobile Commun. Cham, Switzerland: Springer*, 2017, pp. 571–593.

- [122] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, Mar./Apr. 2017.
- [123] E. K. Markakis et al., "EXEGESIS: Extreme edge resource harvesting for a virtualized fog environment," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 173–179, Jul. 2017.
- [124] N. Dragoni et al., "Microservices: Yesterday, today, and tomorrow," in *Present and Ulterior Software Engineering*. Cham, Switzerland: Springer, 2017, pp. 195–216.
- [125] M. Jung, S. Malleri, P. Dalbhanjan, P. Chapman, and C. Kassen, "Microservices on AWS," Amazon Web Services, Inc., New York, NY, USA, Tech. Rep., 2016.
- [126] B. Butzin, F. Gólatowski, and D. Timmermann, "Microservices approach for the internet of things," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Sep. 2016, pp. 1–6.
- [127] T. Vresk and I. Čavrak, "Architecture of an interoperable IoT platform based on microservices," in *Proc. 39th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2016, pp. 1196–1201.
- [128] M. S. de Brito et al., "A service orchestration architecture for fogenabled infrastructures," in *Proc. 2nd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, May 2017, pp. 127–132.
- [129] H. Khazaei, H. Bannazadeh, and A. Leon-Garcia, "End-to-end management of IoT applications," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, Jul. 2017, pp. 1–3.
- [130] L. Sun, Y. Li, and R. A. Memon, "An open IoT framework based on microservices architecture," *China Commun.*, vol. 14, no. 2, pp. 154–162, 2017.
- [131] T. Li, Y. Liu, L. Gao, and A. Liu, "A cooperative-based model for smartsensing tasks in fog computing," *IEEE Access*, vol. 5, pp. 21296–21311, 2017.
- [132] P. Krivic, P. Skocir, M. Kusek, and G. Jezic, "Microservices as agents in

- IoT systems,” in Proc. KES Int. Symp. Agent Multi-Agent Syst., Technol. Appl., 2017, pp. 22–31.
- [133] M. G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, and C. A. De Rose, “Performance evaluation of container-based virtualization for high performance computing environments,” in Proc. 21st Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP), Feb. 2013, pp. 233–240.
- [134] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in Proc. IEEE INFOCOM, Mar. 2012, pp. 945–953.
- [135] E. Bastug, M. Bennis, M. Médard, and M. Debbah, “Toward interconnected virtual reality: Opportunities, challenges, and enablers,” IEEE Commun. Mag., vol. 55, no. 6, pp. 110–117, Jun. 2017.
- [136] Y. Cao, S. Chen, P. Hou, and D. Brown, “FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation,” in Proc. IEEE Int. Conf. Netw., Archit. Storage (NAS), Aug. 2015, pp. 2–11.
- [137] V. Stantchev, A. Barnawi, S. Ghulam, J. Schubert, and G. Tamm, “Smart items, fog and cloud computing as enablers of servitization in healthcare,” Sensors Transducers, vol. 185, no. 2, p. 121, 2015.
- [138] J. A. Khan, C. Westphal, and Y. Ghamri-Doudane, “Offloading content with self-organizing mobile fogs,” in Proc. 29th Int. Teletraffic Congr. (ITC), vol. 1, Jul. 2017, pp. 223–231.
- [139] H. Shi, N. Chen, and R. Deters, “Combining mobile and fog computing: Using coap to link mobile device clouds with fog computing,” in Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst. (DSDIS), Dec. 2015, pp. 564–571.
- [140] T. Wang et al., “Data collection from WSNs to the cloud based on mobile fog elements,” Future Generation Computer Systems, to be published.
- [141] R. Roman, J. Lopez, and M. Mambo, “Mobile edge computing, Fog et al.:

A survey and analysis of security threats and challenges,” *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

- [142] OpenFog Reference Architecture for Fog Computing, OpenFog Consortium Archit. Working Group, Tokyo, Japan, Feb. 2017.



RANESH KUMAR NAHA received the B.Sc. degree in computer science and engineering from the State University of Bangladesh in 2008, the M.Sc. degree from the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, in 2015. He is currently pursuing the Ph.D. degree in reliable resource allocation and scheduling in fog computing environment with the University of Tasmania. He has been

awarded the Tasmania Graduate Research Scholarship for supporting his studies. He served as a Lecturer with the Daffodil Institute of Information Technology, Bangladesh, until 2011. His research interests include wired and wireless network, parallel and distributed computing, cloud computing, Internet of Things (IoT), and fog computing. During his master's study, he has been awarded the Commonwealth Scholarship provided by the Ministry of Higher Education, Malaysia.



wireless networks, and ad hoc networks.

SAURABH GARG is currently a Lecturer with the University of Tasmania, Australia. He is one of the few Ph.D. students who completed in less than three years from the University of Melbourne. He has authored over 40 papers in highly cited journals and conferences. During his Ph.D., he has received various special scholarships for his Ph.D. candidature. His research interests include resource management, scheduling, utility and grid computing, Cloud computing, green computing,



including Telcordia Technologies (where he helped found two of Telcordia's Research Centers in Austin, Texas, and Poznan, Poland); Microelectronics and Computer Corporation (MCC), Austin, TX, USA; GTE (currently Verizon) Laboratories, Boston, MA, USA; and Bell Communications Research, Piscataway, NJ, USA. He was also a Full Professor at RMIT University, and he is currently an Adjunct Professor at Australian National University and a CSIRO Adjunct Fellow. He has produced 170+ journal and conference publications in the areas of IoT, process management, and data management, and has 10 500+ lifetime citations.

DIMITRIOS GEORGAKOPOULOS is currently a Professor in computer science and the Director of the Key Lab for IoT, Swinburne University of Technology, Melbourne, VIC, Australia. Before that he was the Research Director of the CSIRO's ICT Centre and the Executive Director of the Information Engineering Laboratory, which was the largest Computer Science Program in Australia. Before CSIRO, he held research and management positions at several industrial laboratories in USA,



IEA/AIE-2010, and contributor to several industry awards including Black Duck Rookie of the Year Award for Open IoT project (www.openiot.eu). He has (co) authored 75+ journal, conference, and book chapter publications that have received 1200+ google scholar citations (h-index: 20), including two seminal papers in IoT (published by Springer) and Industry 4.0 (published by IEEE).

PREM PRAKASH JAYARAMAN is a Senior Lecturer and Head of the Digital Innovation Lab in the Department of Computer Science and Software Engineering, Faculty of Science, Engineering and Technology at Swinburne University of Technology. Previously, he was a PostDoctoral Research Scientist at CSIRO. He is broadly interested in emerging areas of Internet of Things (IoT), and Mobile and Cloud Computing. He is the recipient of two best paper awards at HICSS 2017 and



patent, monograph, book chapter, journal and conference papers. Some of his publications have been published in the top venue, such as IEEE TMC, IEEE IoT, IEEE TDSC, and IEEE TVT. He received the 2012 Chinese Government Award for Outstanding Students Abroad (Ranked No.1 in Victoria and Tasmania consular districts). He is active in the IEEE Communication Society. He has served as the TPC co-chair, a publicity co-chair, an organization chair, and a TPC member for many international conferences.

LONGXIANG GAO (SM'17) received the Ph.D. degree in computer science from Deakin University, Australia. He is currently a Lecturer with the School of Information Technology, Deakin University. Before joined Deakin University, he was a Post-Doctoral Research Fellow at IBM Research & Development, Australia. His research interests include data processing, mobile social networks, Fog computing, and network security.

Dr. Gao has over 30 publications, including



security and privacy, signal and image processing, data analytics and machine intelligence, and Internet of Things. He is an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS and the IEEE ACCESS. He has served as a program chair, a TPC chair, a symposium chair, and a session chair for a number of international conferences.

YONG XIANG received the Ph.D. degree in electrical and electronic engineering from The University of Melbourne, Australia. He is currently a Professor and the Director of the Artificial Intelligence and Image Processing Research Cluster, School of Information Technology, Deakin University, Australia. He has authored or co-authored over two monographs, over 100 refereed journal articles, and numerous conference papers in these areas. His research interests include information



peer-reviewed papers (seven books, 25 journals, 25 conferences, and five book chapters). His h-index is 20, with a lifetime citation count of 1660+ (Google Scholar). His papers have also received 140+ ISI citations. 70% of his journal papers and 60% of conference papers have been A*/A ranked ERA publication. He has been invited to serve as the Guest Editor for leading distributed systems journals, including the IEEE TRANSACTIONS ON CLOUD COMPUTING, *Future Generation Computing Systems*, and *Software Practice and Experience*. One of his papers was in 2011's Top Computer Science Journal, IEEE COMMUNICATION SURVEYS AND TUTORIALS.

RAJIV RANJAN received the Ph.D. degree in engineering from the University of Melbourne in 2009. He is a Reader at Newcastle University, U.K. Prior to this position, he was a Research Scientist and a Julius Fellow with the CSIRO Computational Informatics Division (formerly known as CSIRO ICT Centre). His expertise is in datacenter cloud computing, application provisioning, and performance optimization. He has authored or co-authored 62 scientific,