

تاریخ دریافت مقاله:

تاریخ پذیرش مقاله: ۹۹/۰۵/۲۵

نوع مقاله: مروری

مروری بر روش‌های پیش‌بینی خرابی در سیستم‌های توزیع شده مقیاس بزرگ

احسان شیرزاد

کارشناسی ارشد فناوری اطلاعات، دانشکده برق و کامپیوتر، دانشگاه بیرجند، بیرجند، ایران.
پست الکترونیکی: e71.shirzad@gmail.com

حمید سعادت‌فر*

استادیار گروه مهندسی کامپیوتر، دانشکده برق و کامپیوتر، دانشگاه بیرجند، بیرجند، ایران.
پست الکترونیکی: saadatfar@birjand.ac.ir

چکیده

(سخت‌افزار و نرم‌افزار) در یک سیستم توزیع شده (شامل خوشه‌های کامپیوتری و سیستم‌های مشبک) در دنیای واقعی پرداخته است.

به‌طور کلی، بررسی این مطالعات نشان می‌دهد که پیش‌بینی برخط (به علت استفاده از ویژگی‌های بیشتر) نتایج بهتری دارد. اما، پیش‌بینی غیر برخط منابع بیشتری را حفظ می‌کند. به همین جهت، پیش‌بینی ترکیبی بهترین گزینه به نظر می‌رسد. همچنین در سال‌های اخیر، مطالعه بر روی داده‌های ثبت وقایع مرتبط با سیستم‌های زیرساخت ابری (مانند خوشه‌های گوگل) محبوبیت بیشتری داشته است. واژه‌های کلیدی: پیش‌بینی خرابی، سیستم توزیع شده، طبقه‌بندی، یادگیری ماشین، داده‌کاوی.

امروزه به علت رشد تولید اطلاعات در جهان، استفاده از سیستم‌های توزیع شده مقیاس بزرگ همه‌گیر شده و کاربردهای فراوانی نه تنها در علوم کامپیوتر بلکه در علوم دیگر نظیر اقلیم‌شناسی، پزشکی و زیست‌شناسی پیدا کرده است. این‌گونه سیستم‌ها به علت طبیعت پویا و پیچیده‌ای که دارند همواره دچار خرابی‌های مختلف می‌گردند. راهبرد بیشتر این سیستم‌ها در هنگام مواجه شدن با خرابی این است که برنامه را مجدداً بر روی بخش دیگری از سیستم اجرا نمایند که این روش موجب هدر رفتن منابع، زمان و انرژی می‌شود؛ بنابراین وجود یک سیستم واکنشی پیشگیرانه که خرابی را قبل از وقوع آن پیش‌بینی و متوقف کند بسیار مفید به نظر می‌رسد. در این مقاله تلاش شده است تا بخشی از پژوهش‌هایی که در چند سال گذشته به جهت پیش‌بینی خرابی در سیستم‌های توزیع شده مقیاس بزرگ انجام گرفته است مرور و دسته‌بندی شود. تمرکز اصلی مقاله روی مطالعاتی می‌باشد که با استفاده از فایل‌های ثبت وقایع، به پیش‌بینی خرابی برنامه‌ها یا منابع

۱-مقدمه

در جهان امروز که استفاده از اینترنت شدت بسیاری پیدا کرده است، همه چیز در حال تولید داده‌های الکترونیکی هستند. یک کاربر ممکن است در هر لحظه از روز خود در حال تولید داده‌های مختلف از طریق تعاملات در شبکه‌های

* نویسنده مسئول

اجتماعی، بارگیری یک فیلم، مسیریابی با تلفن هوشمند و غیره باشد [۱]. حتی اشیاء برقی اطراف ما هم می‌توانند به اینترنت متصل شوند و با استفاده از حسگرها و رادیوشناسه داده تولید کنند [۲]. همچنین، منابع جدید تولید اطلاعات (مانند تلفن‌های هوشمند)، افزایش حجم متعلق به هر داده (مانند فایل ویدئو که حجم زیادی نسبت به فایل متنی دارد) و زیرشاخه‌های جدید داده (مانند تراکنش‌های مالی) از عواملی هستند که باعث رشد سریع حجم داده‌های ذخیره شده در جهان شده‌اند [۳]. رشد داده‌ها به حدی است که دانشمندان با انبوهی از داده‌ها مواجه هستند که از قانون مور پیروی می‌کند که بیان‌کننده این قضیه است که داده‌های موجود در جهان هر دو سال دو برابر می‌شوند [۴]. بنابراین تخمین زده می‌شود که داده‌های موجود در جهان به ۳۵ زتابایت در ۲۰۲۰ [۵] و ۱۶۳ زتابایت در ۲۰۲۵ [۶] برسد. داده‌های بزرگ نیازمند سیستم‌های کامپیوتری بزرگی برای پردازش هستند؛ به همین جهت از سیستم‌های توزیع‌شده مقیاس بزرگ استفاده می‌شود.

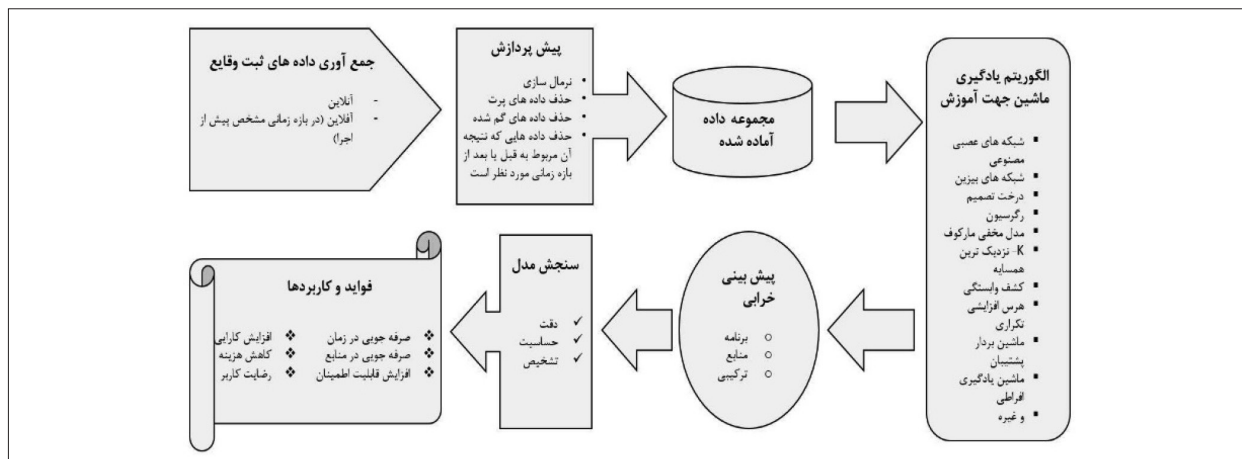
یک سیستم توزیع‌شده [۷] به سیستمی گفته می‌شود که اجزای آن، که شامل رایانه‌های مستقل هستند، بر روی یک شبکه متصل قرار دارند و توسط یک میان‌افزار (یا یک چارچوب نرم‌افزاری خاص) با یکدیگر مرتبط و هماهنگ هستند تا به یک هدف مشترک دست پیدا کنند. اجزای این‌گونه سیستم‌ها در به اشتراک‌گذاری منابع و قابلیت‌هایشان به گونه‌ای تعامل دارند که برای کاربران به صورت یک سیستم منسجم و یکپارچه به نظر می‌رسند. بنابراین یک سیستم توزیع‌شده در مقیاس بزرگ، مجموعه‌ای متصل از سیستم‌های رایانه‌ای همگن یا ناهمگن است که در جهت حل یک مسئله بزرگ همکاری می‌کنند؛ به این نحو که ابتدا مسئله را به چندین وظیفه مستقل تقسیم می‌کند و آن‌ها را بین هر سیستم برای پردازش فردی توزیع می‌کند و سپس نتایج به‌دست‌آمده توسط هر یک را دریافت و تجمیع می‌نماید تا جواب نهایی مسئله به دست آید [۸]. امروزه سیستم‌های توزیع‌شده مقیاس بزرگ به علت کارایی و

سرعت بالا علاوه بر علوم کامپیوتر در علوم دیگری نظیر فیزیک [۹]، اقلیم‌شناسی [۱۰]، نجوم [۱۱]، زیست‌شناسی [۱۲] و پزشکی [۱۳، ۱۴] کاربردهای بسیاری دارند.

سیستم‌های توزیع‌شده مقیاس بزرگ مانند خوشه‌های کامپیوتری^۱ و سیستم‌های مشبک^۲، به علت طبیعت پویا و پیچیده‌ای که دارند همواره با مشکلاتی از جمله خرابی (شکست) برنامه‌ها و منابع (حافظه، پردازنده، سیستم‌عامل و غیره) روبه‌رو هستند. یک خرابی زمانی اتفاق می‌افتد که خروجی به‌دست‌آمده با خروجی مورد انتظار متفاوت باشد [۱۵]. خرابی در سیستم‌های توزیع‌شده معمولاً به علت اشکالات نرم‌افزاری و سخت‌افزاری مانند فساد داده‌ها، معلق شدن عملیات، برگرداندن مقادیر غلط، ماشین‌آلات ناسازگار، قطع شبکه و فضای ناکافی دیسک اتفاق می‌افتد [۱۶]. خرابی‌ها علاوه بر هدر دادن منابع، انرژی و زمان یک سیستم توزیع‌شده می‌توانند باعث کاهش کارایی و قابلیت اطمینان سیستم، نقض توافقات سطح خدمات و در نهایت از دست دادن مشتری شوند [۱۷]؛ بنابراین یک رویکرد مدیریتی پیشگیرانه در سیستم‌های توزیع‌شده که بتواند یک شکست را پیش از آن که اتفاق بیفتد پیش‌بینی نماید بسیار مفید به نظر می‌رسد.

تاکنون مطالعات بسیاری به جهت کاهش خرابی و افزایش کارایی در سیستم‌های توزیع‌شده مقیاس بزرگ مانند خوشه‌های کامپیوتری و سیستم‌های مشبک انجام گرفته است که شاخه‌ای از آن به بررسی فایل‌های ثبت وقایع^۳ به جا مانده از این‌گونه سیستم‌ها و استفاده از الگوریتم‌های یادگیری ماشین^۴ به جهت پیش‌بینی خرابی پرداخته است. در فایل‌های ثبت وقایع آمار مربوط به رخداد هر رویداد خاص در چرخه‌های زمانی متفاوت به‌علاوه مقادیر مربوط به هر رویداد ذخیره می‌شود [۱۸]. شکل (۱) شمای کلی پیش‌بینی خرابی در سیستم‌های توزیع‌شده با استفاده از داده‌های ثبت وقایع آن‌ها را (با توجه به مقالات

1- Computer clusters
2- Grid computing systems
3- Log files (Trace data)
4- Machine learning



شکل ۱: شمای کلی پیش بینی خرابی در سیستم های توزیع شده [۱۷، ۱۹]

الگوهای پنهان در داده ها است [۲۵]؛ و یادگیری با نظارت (پیش بینی) که به معنی آموزش چگونگی استنتاج عملیات از یک مجموعه داده آموزش برچسب دار می باشد [۲۶]. بیشتر کارهایی که در این مقاله مرور می شوند مرتبط با یادگیری با نظارت هستند.

۲-۲- سنجش نتایج پیش بینی

زمانی که یک فرآیند پیش بینی به انجام می رسد، نتایج آن باید سنجیده شوند. هدف از پیش بینی خرابی این است که هرچقدر امکان دارد خرابی به درستی پیش بینی گردد. به علاوه این که دقت کل پیش بینی نیز در حد مورد قبولی باشد. در ادامه ابتدا مفاهیم مرتبط با ماتریس آشفتگی^۵ و سپس پرکاربردترین معیارهای سنجش روش های پیش بینی بیان می شوند.

۲-۲-۱- ماتریس آشفتگی

ماتریس آشفتگی یا سردرگمی که به عنوان ماتریس طبقه بندی نیز شناخته می شود، برای ارزیابی دقت پیش بینی یک مدل استفاده می شود؛ این ماتریس اندازه گیری می کند که یک مدل چقدر در پیش بینی هایی که انجام داده دچار سردرگمی شده است [۲۷]. جدول (۱) قالب یک ماتریس آشفتگی را برای یک مسئله طبقه بندی با دو رده (خرابی و عدم خرابی) نشان می دهد (دقت شود که چون هدف طبقه بندی، پیش بینی خرابی است پس رده مثبت همان

مرور شده این حوزه) نشان می دهد.

در مقاله حاضر به بررسی کارهایی پرداخته شده است که در زمینه پیش بینی خرابی در سیستم های توزیع شده مقیاس بزرگ (شامل خوشه های کامپیوتری و سیستم های مشبک) با استفاده از فایل های ثبت وقایع آن ها انجام گرفته است. از آن جا که عملیات پیش بینی وابسته به یادگیری ماشین است، در بخش دوم تعاریف مرتبط با یادگیری ماشین بیان می شود. در بخش سوم کارهایی که در حوزه پیش بینی خرابی در سیستم های مذکور انجام گرفته است، تقسیم بندی و مرور می شوند. بخش چهارم نیز بخش پایانی مقاله و شامل بحث و نتیجه گیری نهایی است.

۲- تعاریف

۲-۱- یادگیری ماشین

یادگیری ماشین یکی از شاخه های هوش مصنوعی است که سیستم را قادر می سازد تا با استفاده از تجارب خود به طور خودکار یاد بگیرد و بهبود یابد [۱۹]. امروزه یادگیری ماشین علاوه بر پیش بینی خرابی، کاربردهای بسیار دیگری از جمله پزشکی [۲۰]، بازیابی تصاویر [۲۱]، طبقه بندی ترافیک شبکه [۲۲]، پردازش زبان طبیعی [۲۳، ۲۴] و غیره دارد. به طور کلی دو نوع روش یادگیری ماشین وجود دارد: یادگیری بدون نظارت (توصیفی) که به معنی آموزش با استفاده از داده های بدون برچسب برای کشف

جدول ۱: ماتریس آشفتگی یک پیش‌بینی دو رده‌ای (رده مثبت: خرابی، رده منفی: عدم خرابی) [۲۷]

مقادیر واقعی		ماتریس آشفتگی (درهم ریختگی)	
		مثبت	منفی
FP (مثبت کاذب)	TP (مثبت واقعی)	مثبت	مقادیر پیش‌بینی شده
		منفی	TN (منفی واقعی)
FN (منفی کاذب)	TN (منفی واقعی)	مثبت	
		منفی	

خرابی می‌باشد). TP به معنی پیش‌بینی صحیح خرابی، FP به معنی پیش‌بینی اشتباه خرابی، TN به معنی پیش‌بینی صحیح عدم خرابی و FN به معنی پیش‌بینی اشتباه عدم خرابی می‌باشد.

۲-۲-۲- معیارهای سنجش

به منظور بررسی کیفیت الگوریتم‌های پیش‌بینی خرابی و مقایسه قدرت آن‌ها نیازمند معیارهای استاندارد برای سنجش الگوریتم‌های یادگیری ماشین هستیم. معروف‌ترین معیارهای سنجش برای مسائل پیش‌بینی شامل دقت^۶، صحت^۷، تشخیص^۸ و حساسیت^۹ هستند [۲۸]. جدول (۲) نیز معیارهای مذکور را به‌طور خلاصه بیان می‌کند. دقت درصد کل پیش‌بینی‌های صحیح را نشان می‌دهد. صحت بیان می‌کند که از میان تمام مواردی که پیش‌بینی شده که به رده خرابی تعلق دارند، چه درصدی صحیح بوده است. حساسیت نشان می‌دهد که چه درصدی از خرابی درست تشخیص داده شده است. تشخیص نیز بیان‌کننده درصد پیش‌بینی صحیح عدم خرابی می‌باشد.

۳- پیش‌بینی خرابی

همان‌طور که قبلاً نیز اشاره شد، وقوع خرابی در سیستم‌های توزیع‌شده مقیاس بزرگ باعث کاهش کارایی و قابلیت اطمینان این‌گونه سیستم‌ها و هدر رفتن منابع و انرژی می‌شود و مشکلی است که بسیار اتفاق می‌افتد. به عنوان مثال، در سیستم مشبک داس ۲^{۱۰} نرخ خرابی کارها بیش از ۱۰٪ بوده [۲۹] و در تراگرید^{۱۱} بین ۱۰ تا

جدول ۲: پرکاربردترین معیارهای سنجش روش‌های پیش‌بینی [۲۸]

نام معیار	فرمول	نام‌های دیگر
دقت	$\frac{TP + TN}{TP + TN + FP + FN}$	
صحت	$\frac{TP}{TP + FP}$	اطمینان ^۱
حساسیت (TP rate)	$\frac{TP}{TP + FN}$	ارجاع ^۲ ، پشتیبانی ^۳
تشخیص (TN rate)	$\frac{TN}{TN + FP}$	

1- Confidence

2-

3- Support

۴۵ درصد نرخ خرابی وجود داشته است [۳۰]. در آوریل ۲۰۰۸، ۷۰٪ کارهای انجام شده در خوشه^{۱۲} ام ۴۵ یاهو^{۱۳} شکست خورده‌اند [۳۱]. بیش از ۸٪ از کارهای خوشه اوپن‌کلود^{۱۴} و ۱۹ درصد از کارهای خوشه وب ماینینگ^{۱۵} خراب شده است [۳۲]. در خوشه‌های کامپیوتری لوس آلاموس^{۱۵} نیز در سال ۲۰۰۱ بیش از ۶٪ نرخ خرابی در گره‌های کامپیوتری آن وجود داشته [۳۳] و همچنین نرخ خرابی ماشین‌ها در خوشه‌های مایکروسافت حدود ۸٪ بوده است [۳۴]. به‌علاوه در خوشه اوپن‌کلود، حدود ۵۳٪ از زمان فعالیت سیستم صرف اجرای کارهای ناموفق شده است [۳۵]. بنابراین وجود یک سیستم مدیریتی پیشگیرانه در سیستم‌های توزیع‌شده که خرابی را قبل از اتفاق افتادن آن پیش‌بینی کند مهم به نظر می‌رسد و به همین جهت پیش‌بینی خرابی در این‌گونه سیستم‌ها یکی از موضوعات مورد توجه پژوهشگران در چند سال اخیر بوده است.

در این بخش تلاش شده است تا برخی از کارهایی که در چند سال اخیر بر روی پیش‌بینی خرابی در سیستم‌های توزیع‌شده مقیاس بزرگ (شامل خوشه‌های کامپیوتری و سیستم‌های مشبک) انجام شده است، از جنبه‌های مختلف (نوع پیش‌بینی، نوع ویژگی‌های استفاده شده، زمان پیش‌بینی و غیره) بررسی گردد. مقالات بررسی شده را می‌توان به صورت اجمالی در جدول (۳) مشاهده نمود.

12- Yahoo M45 Cluster

13- OpenCloud

14- Web Mining

15- Los Alamos National Laboratory (LANL)

6- Accuracy

7- Precision

8- Specificity

9- Sensitivity

10- Recall

11- TeraGrid

جدول ۳: خلاصه‌ای از کارهای بررسی شده در زمینه پیش‌بینی خرابی در سیستم‌های توزیع شده مقیاس بزرگ به ترتیب سال چاپ مقاله

سال	نویسنده	نوع خرابی پیش‌بینی شده	زمان پیش‌بینی	الگوریتم استفاده شده	سیستم مورد بررسی
۲۰۰۱	کستلی ^۱ و همکارانش [۳۸]	منابع (نرم‌افزاری)	برخط	شبکه پاداش تصادفی ^۲	سرویس دهنده آی‌بی‌ام ^۳
۲۰۰۲	لی ^۴ و همکارانش [۳۹]	منابع (نرم‌افزاری)	برخط	مدل خودهمبسته میانگین متحرک ^۵	سرویس دهنده وب آپاچی ^۶
۲۰۰۳	برنجی ^۷ و همکارانش [۴۰]	منابع (عناصر سخت‌افزاری)	برخط	تابع پایه شعاعی ^۸	سیستم مرکز تحقیقات ایمز ناسا ^۹
۲۰۰۷	سالفنر ^{۱۰} و مالک ^{۱۱} [۳۷]	ترکیبی (نوع خرابی)	برخط	مدل نیمه مارکوف مخفی ^{۱۲}	یک سیستم ارتباطی تجاری
۲۰۰۷	فو ^{۱۳} و زو ^{۱۴} [۴۱]	منابع (سخت‌افزاری و نرم‌افزاری)	غیر برخط / برخط	شبکه‌های عصبی مصنوعی ^{۱۵} و شبکه‌های بی‌زین ^{۱۶}	خوشه لوس آلاموس / سیستم مشبک دانشگاه ایالتی وین
۲۰۰۷	لیانگ ^{۱۷} و همکارانش [۴۲]	ترکیبی (هرگونه رویداد مهلک که منجر به خرابی می‌شود)	برخط	هرس افزایشی تکراری برای کاهش خطا ^{۱۸} ، ماشین بردار پشتیبان ^{۱۹} و نزدیک‌ترین همسایه ^{۲۰}	کامپیوترهای بلوجین/ال ^{۲۱}

- 1- Castelli
- 2- Stochastic Reward Nets (SRN)
- 3- IBM
- 4- Li
- 5- Autoregressive Moving Average (ARMA) model
- 6- Apache
- 7- Berenji
- 8- Radial Basis Function
- 9- NASA Ames Research Center
- 10- Salfner
- 11- Malek
- 12- Hidden Semi-Markov Model (HSMM)
- 13- Fu
- 14- Xu
- 15- Artificial Neural Network
- 16- Bayesian Network
- 17- Liang
- 18- Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
- 19- Support Vector Machine (SVM)
- 20- Nearest Neighbor
- 21- Bluegene/L

۲۰۰۸	چاولا ^{۲۲} و همکارانش [۴۳]	برنامه‌ها (کار)	غیر برخط	درخت تصمیم C4.5 و هرس افزایشی تکراری برای کاهش خطا	سیستم مشبک دانشگاه پردو به نام تراگرید
۲۰۰۸	زینعلی‌پور ^{۲۳} و همکارانش [۴۴]	منابع (سایت‌ها)	غیر برخط	درخت تصمیم	سیستم مشبک ای‌جی‌ای ^{۲۴}
۲۰۰۹	هکر ^{۲۵} و همکارانش [۴۵]	منابع (گره‌ها)	برخط	مدل نیمه مارکوف ^{۲۶}	کامپیوترهای بلوجین
۲۰۱۰	فو و زو [۴۶]	منابع (سخت‌افزاری و نرم‌افزاری)	غیر برخط / برخط	شبکه‌های عصبی	سیستم مشبک دانشگاه ایالتی وین
۲۰۱۱	هافمن ^{۲۷} و همکارانش [۴۷]	ترکیبی (نوع خرابی)	برخط	زنجیره مارکوف و تابع پایه شعاعی عمومی	یک سیستم ارتباطی تجاری
۲۰۱۲	بالدونی ^{۲۸} و همکارانش [۴۸]	منابع (نرم‌افزاری)	برخط	پردازش رویداد پیچیده ^{۲۹} و مدل پنهان مارکوف	یک سیستم کنترل ترافیک هوایی مبتنی بر میان‌افزار کوربا
۲۰۱۲	فو و همکارانش [۴۹]	ترکیبی (رویداد مهلک و خرابی در گره‌ها و برنامه‌ها)	برخط	مدل کشف وابستگی بر اساس آپریوری ^{۳۰}	کامپیوترهای بلوجین/ال، خوشه‌های هادوپ موبایل چین و خوشه لوس آلاموس
۲۰۱۲	سعادت‌فر ^{۳۱} و همکارانش [۵۰]	برنامه‌ها (کار)	غیر برخط	شبکه‌های بی‌زین	سیستم مشبک آورگرید ^{۳۲}
۲۰۱۴	چن ^{۳۳} و همکارانش [۵۱]	برنامه‌ها (کار) / وظیفه	برخط	شبکه‌های عصبی راجعه ^{۳۴}	خوشه گوگل
۲۰۱۴	سعادت‌فر و دلداری ^{۳۵} [۵۲]	برنامه‌ها (کار)	غیر برخط	شبکه‌های بی‌زین	خوشه لوس آلاموس

- 22- Chawla
- 23- Zeinalipour-Yazti
- 24- EGEE
- 25- Hacker
- 26- Semi Markov Process (SMP)
- 27- Hoffmann
- 28- Baldoni
- 29- Complex Event Processing (CEP)
- 30- Apriori
- 31- Saadatfar
- 32- AuverGrid
- 33- Chen
- 34- Recurrent Neural Network (RNN)
- 35- Deldari

خوشه گوگل	مدل دستگاه یادگیری افراطی متوالی برخط ^{۵۲} و ماشین بردار پشتیبان متوالی برخط ^{۵۳}	غیر برخط / برخط	برنامه‌ها (کار)	لیو ^{۵۲} و همکارانش [۶۱]	۲۰۱۷
خوشه تحقیقاتی اوپن کلود	درخت تصمیم گرادیان بوستینگ ^{۵۵} ، K نزدیک‌ترین همسایه، جنگل تصادفی ^{۵۶} و رگرسیون لجستیک	غیر برخط	برنامه‌ها (کار)	هانگیان و همکارانش [۶۲]	۲۰۱۷
خوشه گوگل، خوشه تحقیقاتی اوپن کلود و خوشه لوس آلاموس	جنگل‌های تصادفی	غیر برخط / برخط	برنامه‌ها (کار) / وظیفه	ال سید و همکارانش [۶۳]	۲۰۱۷
خوشه گوگل	مدل خطی عمومی ^{۵۷} ، جنگل تصادفی، شبکه عصبی، بوستینگ ^{۵۸} ، درخت و درخت شرطی ^{۵۹}	غیر برخط	برنامه‌ها (وظیفه)	سولهیا و همکارانش [۶۴]	۲۰۱۸
سرویس دهنده نرم‌افزاری نتفلیکس ^{۶۱}	شبکه‌های بی‌زین	برخط	منابع (اجزاء)	پیتاکرات ^{۶۰} و همکارانش [۶۵]	۲۰۱۸
خوشه محاسباتی لوس آلاموس	ماشین بردار پشتیبان، جنگل تصادفی، K نزدیک‌ترین همسایه، درخت طبقه‌بندی و رگرسیون و تحلیل افتراقی خطی	غیر برخط	ترکیبی (خرابی‌های سیستم و برنامه‌ها)	محمد ^{۶۲} و همکارانش [۶۶]	۲۰۱۹
خوشه گوگل	طبقه‌بند تقویت گرادیان ایکس ^{۶۴}	غیر برخط	برنامه‌ها (وظیفه)	شتی ^{۶۳} و همکارانش [۶۷]	۲۰۱۹

خوشه گوگل	تحلیل افتراقی خطی ^{۳۷} ، تحلیل افتراقی درجه دوم ^{۳۸} و رگرسیون لجستیک ^{۳۹}	برخط	برنامه‌ها (کار)	روزا ^{۳۶} و همکارانش [۵۳]	۲۰۱۵
خوشه گوگل	شبکه‌های عصبی	برخط	برنامه‌ها (کار) / وظیفه	روزا و همکارانش [۵۴]	۲۰۱۵
خوشه گوگل و خوشه تحقیقاتی اوپن کلود	رگرسیون لجستیک چند جمله‌ای ^{۴۲} و درخت طبقه‌بندی و رگرسیون ^{۴۲}	غیر برخط / برخط	برنامه‌ها (کار)	ال سید ^{۴۰} و شرودر ^{۴۱} [۵۵]	۲۰۱۵
خوشه شخصی با ۱۱ گره	مدل مخفی مارکوف ^{۴۵}	برخط	منابع (گره‌ها)	آگراوال ^{۴۴} و همکارانش [۵۶]	۲۰۱۵
خوشه گوگل	مدل طبقه‌بندی بر اساس K-نزدیک‌ترین همسایه و مدل خوشه‌بندی بر اساس K-میانگین	برخط	برنامه‌ها (وظیفه)	هانگیان ^{۴۶} و همکارانش [۵۷]	۲۰۱۶
خوشه تحقیقاتی مرکز محاسبات علمی پژوهشی انرژی ملی به نام جین پول ^{۴۸}	درخت تصمیم، جنگل‌های تصادفی، بی‌زین ساده، رگرسیون لجستیک، ماشین بردار پشتیبان	برخط	برنامه‌ها (کار)	یو ^{۴۷} و همکارانش [۵۸]	۲۰۱۶
خوشه گوگل	شبکه‌های عصبی	غیر برخط / برخط	برنامه‌ها (کار) / رویداد	روزا و همکارانش [۵۹]	۲۰۱۷
خوشه گوگل	حافظه طولانی کوتاه-مدت ^{۵۱}	برخط	برنامه‌ها (کار) / وظیفه	اسلام ^{۴۹} و مانیوانان ^{۵۰} [۶۰]	۲۰۱۷

- 52- Liu
53- Online Sequential Extreme Learning Machine (OS-ELM)
54- Online Sequential Support Vector Machine (OS-SVM)
55- Gradient Boosting Decision Tree (GBDT)
56- Random Forest
57- General Linear Model (GLM)
58- Boost
59- Conditional Tree
60- Pitakrat
61- Netflix
62- Mohammed
63- Shetty
64- XGboost Classifier

- 36- Rosa
37- Linear Discriminant Analysis (LDA)
38- Quadratic Discriminant Analysis (QDA)
39- Logistic Regression
40- El-Sayed
41- Schroeder
42- Multinomial Logistic Regression (MLR)
43- Classification And Regression Tree (CART)
44- Agrawal
45- Hidden Markov Model (HMM)
46- Hongyan
47- Yoo
48- Genepool
49- Islam
50- Manivannan
51- Long Short-Term Memory (LSTM)

۲۰۲۰	شیرزاد و سعادت فر [۶۸]	برنامه‌ها (کار)	غیر برخط	شبکه‌های بیزین، شبکه‌های عصبی، ماشین بردار پشتیبان، رگرسیون لجستیک و درخت‌های C5.0 و CART	خوشه تحقیقاتی اوپن کلود
------	------------------------	-----------------	----------	---	-------------------------

در ادامه این بخش خلاصه‌ای از کارهای ذکر شده در جدول (۳) بیان می‌شود. به منظور درک بهتر، مقالات از لحاظ نوع خرابی پیش‌بینی شده به دو دسته منابع و برنامه‌ها تقسیم می‌شوند؛ همچنین هر بخش از لحاظ زمان پیش‌بینی به دو نوع غیر برخط و برخط نیز طبقه‌بندی می‌گردد. پیش‌بینی غیر برخط خرابی نوعی طبقه‌بندی یا مدل‌سازی است که قبل از شروع کار سیستم یا اجرای برنامه انجام می‌گیرد و مرتبط با معیارهای کیفیت، قابلیت اطمینان و پایداری در روندهای طولانی‌مدت می‌باشد [۳۶]؛ اما پیش‌بینی برخط خرابی به این معنی است که وضعیت فعلی سیستم به منظور پیش‌بینی وقوع شکست در آینده نزدیک مورد ارزیابی قرار می‌گیرد و تمرکز اصلی آن انجام پیش‌بینی‌های کوتاه‌مدت بر اساس وضعیت در حال اجرای برنامه یا سیستم است [۳۷].

به‌طور کلی، در بین مطالعات لیست شده در جدول (۳)، پیش‌بینی برخط به این دلیل که علاوه بر ویژگی‌هایی که قبل از شروع اجرای برنامه در دسترس هستند (مانند ساعت ورود، حافظه و پردازنده درخواستی و وضعیت بارکاری سیستم)، می‌تواند از ویژگی‌های بعد از شروع اجرا (مانند حافظه و پردازنده مصرفی) نیز در فرایند یادگیری استفاده کند و در نهایت نتایج بهتری به دست آورد، محبوب‌تر بوده است. اما در سال‌های اخیر، توجه به پیش‌بینی غیر برخط (و ترکیبی) به این علت که منابع و زمان بیشتری را حفظ می‌کند و رضایت کاربری بیشتری ایجاد می‌نماید، افزایش یافته است. همچنین در سال‌های اخیر، مطالعه پیش‌بینی خرابی برنامه‌ها نسبت به پیش‌بینی خرابی منابع مورد توجه بیشتری قرار گرفته است. چراکه پیشرفت‌هایی که

در سال‌های اخیر در زیرساخت سیستم‌های توزیع شده به خصوص منابع سخت‌افزاری انجام شده است، قابلیت اطمینان این سیستم‌ها را به حد قابل قبولی رسانده است. همچنین خرابی در برنامه‌ها به این علت که کاربر در اجرای آن‌ها دخیل است، بیشتر اتفاق می‌افتد. در بین الگوریتم‌های یادگیری نیز، روش‌هایی که بر پایه شبکه‌های عصبی، شبکه‌های بیزین و درخت‌های تصمیم بوده‌اند، محبوبیت بیشتری داشته‌اند. همچنین در سال‌های اخیر، مطالعات بیشتری بر روی خوشه‌های محاسباتی زیرساخت ابری به علت ماهیت تجاری آن‌ها، به خصوص خوشه‌های گوگل که داده‌های ثبت وقایع خود را به صورت گسترده به اشتراک گذاشته است، به انجام رسیده است.

۳-۱- پیش‌بینی غیر برخط خرابی منابع

در این قسمت مطالعاتی که در زمینه پیش‌بینی هرگونه مشکل و خرابی در منابع (سخت‌افزار و نرم‌افزار) قبل از شروع کار سیستم (غیر برخط) انجام گرفته است بررسی می‌شود؛ لازم به ذکر می‌باشد که اگر پژوهشی شامل پیش‌بینی در هر دو حالت غیر برخط و برخط بوده نیز در این قسمت قرار گرفته است.

فو و زو در سال ۲۰۰۷ [۴۱] یک چارچوب پیش‌بینی خرابی در سیستم‌های توزیع شده به نام اچ‌پرفکت^{۱۶} پیشنهاد دادند. این چارچوب روی گره اصلی پیاده‌سازی می‌شود و جزئی از زمان‌بند کار است و تلاش می‌کند تا با استفاده از ویژگی‌های استخراج شده از فایل‌های ثبت وقایع سیستم هرگونه رویداد خرابی را پیش‌بینی نماید. منظور از رویداد خرابی در اینجا هرگونه مشکل سخت‌افزاری (شامل حافظه و پردازنده) و هرگونه مشکل نرم‌افزاری (شامل سیستم‌عامل و فایل سیستم) می‌باشد. هدف چارچوب این است که تمامی رویدادها را ضبط و ثبت کند و سپس رکوردهای مرتبط با خرابی را استخراج نماید و برای پیش‌بینی کننده بفرستد. برای تشخیص خرابی، چارچوب نیازمند چندین متغیر عملکردی است

16- hPREFECT

که از مشخصات سیستم به دست می‌آیند و نشان دهنده تفاوت بین حالت‌های سیستم در زمان اجرای بدون مشکل و اجرای خراب هستند. این ویژگی‌های عملکرد از فایل‌های ثبت وقایع سیستم به دست می‌آید و شامل شماره شناسه رویداد خرابی، مکان خرابی (شناسه خوشه کامپیوتری و شناسه گره)، نوع خرابی (حافظه، پردازنده، سیستم‌عامل یا فایل)، برچسب زمانی که از زمان اتفاق افتادن شکست فعال می‌شود، میزان مصرف پردازنده، حجم داده‌های ورودی و خروجی، تعداد بسته‌های منتقل شده، رابطه زمانی، رابطه مکانی و غیره می‌باشد. رابطه زمانی و مکانی اصلی‌ترین ویژگی‌های مربوط به رویدادها در این پژوهش است. برای به دست آوردن روابط زمانی بین رویدادها از یک مدل کوواریانس کروی به جهت اندازه‌گیری همبستگی زمانی بین وقایع خرابی استفاده می‌شود که برای محاسبه مقدار کوواریانس، فاصله زمانی بین نمونه‌های مربوطه اندازه‌گیری می‌گردد که میزان همبستگی آن‌ها را مشخص می‌کند. همچنین برای به دست آوردن روابط مکانی از توزیع احتمالی شکست‌ها برای محاسبه کوواریانس فضایی بین نمونه‌های مربوطه استفاده می‌شود. در واقع این چارچوب پدیده‌های انتشار خرابی را با بررسی ارتباطات شکست در دامنه‌های زمان و فضا مدل می‌کند. پژوهشگران این مقاله در نهایت چارچوب پیشنهادی خود را روی خوشه‌های محاسباتی آزمایشگاه ملی لوس آلاموس (LANL) در حالت غیر برخط و روی سیستم مشبک دانشگاه ایالتی وین (WSU) در حالت برخط پیاده‌سازی کردند و با استفاده از شبکه‌های عصبی مصنوعی و شبکه‌های بیزین به پیش‌بینی خرابی پرداختند و به میزان دقت ۷۶,۵ درصد برای حالت غیر برخط و ۷۰,۳ برای حالت برخط دست یافتند. نویسندگان این مقاله بعدها در سال ۲۰۱۰ [۴۶] مطالعه کامل‌تری در مورد همبستگی‌های زمانی و مکانی در سیستم‌های توزیع‌شده و با تمرکز روی سیستم مشبک WSU ارائه دادند و به پیش‌بینی خرابی‌های سخت‌افزاری و نرم‌افزاری در سطح خوشه و گره با استفاده از شبکه‌های عصبی مصنوعی در حالت غیر برخط و برخط پرداختند

که به میزان ۷۲,۷٪ الی ۸۵,۳ درصد (وابسته به پنجره آموزش) می‌توانست خرابی را به درستی پیش‌بینی نماید. همچنین زینعلی‌پور و همکارانش در سال ۲۰۰۸ [۴۴] یک چارچوب جدید به نام فیل‌رنک^{۱۷} برای ادغام و رتبه‌بندی منابع اطلاعاتی که نشان دهنده ویژگی‌های خرابی و بگاه‌ها در سیستم‌های مشبک است ارائه دادند. آن‌ها با بیان ویژگی‌های پویای سیستم‌های مشبک و عدم وجود مدیریت مرکزی در آن‌ها و ناهمگون بودن اجزای تشکیل‌دهنده آن‌ها، تأکید کردند که پیدا کردن و تحلیل تک‌تک شکست‌ها در این سیستم‌های بزرگ کار سخت و زمانبری است. بنابراین شناسایی وضعیت کلی سیستم و شناسایی و بگاه‌هایی که بیشترین پتانسیل ایجاد خرابی را دارند و سپس حذف این و بگاه‌ها از زمان‌بند کارها می‌تواند بسیار کارآمدتر از شناسایی خرابی‌های فردی باشد و قابلیت اعتماد و انطباق زیرساخت را بالا ببرد. در واقع هدف کلی این است که با استفاده از ویژگی‌های مختلف سخت‌افزاری، نرم‌افزاری و بارکاری مشخص شود که هر و بگاه چقدر احتمال ایجاد خرابی دارد. معماری فیل‌رنک شامل چهار جزء اصلی است: یک ماتریس به نام FSM که پارامترهای مؤثر در تشخیص خرابی که توسط منابع بازخورد به دست آمده‌اند را به‌طور متراکم نمایش می‌دهد. یک دنباله زمانی از FSM ها که بر روی دیسک محلی ذخیره می‌شود. یک ماژول رتبه‌بندی که به‌طور مداوم ماتریس‌های FSM را از لحاظ پتانسیل ایجاد خرابی رتبه‌بندی می‌کند و تعداد K و بگاهی که بیشترین پتانسیل را دارند را مشخص می‌نماید (K توسط کاربر تعیین می‌شود). آخرین جزء این چارچوب هم مجموعه‌ای از ابزارهای کاوش داده‌ای است که استخراج روندها، شباهت‌ها، آموزش و پیش‌بینی خرابی را ممکن می‌سازد. منابع بازخورد که پارامترهای مؤثر بر خرابی (ویژگی‌های پیش‌بینی) از آن‌ها به دست می‌آید همان ابزار پرس‌وجو و پایشی هستند که در سیستم‌های مشبک تعبیه شده‌اند که شامل پایش دسترسی به سرور (سام^{۱۸})، پرس‌وجوهای

17- FailRank
18- SAM

شاخص اطلاعات (بی‌دی ۲^{۱۹})، آمار شبکه (جی‌استات^{۲۰})، داده‌های حسگر میزبان (گرید آی‌سی‌ای^{۲۱}) و داده‌های توموگرافی شبکه (اسموک‌پینگ^{۲۲}) می‌باشند. برای عملیات پیش‌بینی نیز از درخت تصمیم استفاده می‌گردد که قوانین طبقه‌بندی را از میان داده‌ها استخراج می‌کند. این قوانین نشان می‌دهند که تحت چه شرایطی و با چه احتمالی یک وبگاه ایجاد خرابی می‌نماید؛ بنابراین وبگاه‌هایی که قوانین خرابی را برآورده می‌کنند از روند زمان‌بندی کارها خارج می‌شوند و کارها روی وبگاه‌های دیگر اجرا می‌گردند. نویسندگان مقاله با پیاده‌سازی چارچوب پیشنهادی روی داده‌های ثبت وقایع متعلق به ۳۲ روز از سیستم مشبک ای‌جی‌ای‌ای نشان دادند که این چارچوب می‌تواند تا ۹۳٪ موارد منجر به خرابی را درست پیش‌بینی نماید.

محمد و همکارانش نیز در سال ۲۰۱۹ [۶۶] با تمرکز بر روی الگوریتم‌های طبقه‌بندی به پیش‌بینی خرابی‌های سیستم و برنامه‌ها در سیستم‌های محاسباتی با کارایی بالا^{۲۳} پرداختند. آن‌ها الگوریتم‌های ماشین بردار پشتیبان، جنگل تصادفی، k نزدیک‌ترین همسایه، درخت طبقه‌بندی و رگرسیون و تحلیل افتراقی خطی را روی داده‌های به‌دست‌آمده از ابررایانه‌های لوس آلاموس پیاده‌سازی نمودند و نشان دادند که در این شرایط الگوریتم ماشین بردار پشتیبان با دقت ۹۰٪ بهترین توانایی پیش‌بینی خرابی را دارد.

به‌طور کلی، مطالعه بر روی خرابی منابع به جهت پیش‌بینی با استفاده از داده‌های ثبت وقایع، قدمت بیشتری دارد و در سال‌های اخیر از محبوبیت کمتری برخوردار بوده است. علت این امر می‌تواند با زیرساخت‌های سیستم‌های توزیع شده مرتبط باشد؛ بر اساس مقالات بررسی شده، تا قبل از گسترش سیستم‌های محاسبات ابری، خرابی در سیستم‌های توزیع‌شده که بیشتر خوشه‌های کوچک یا سیستم‌های مشبک بودند، با منابع

بخصوص سخت‌افزار مرتبط بوده است. همچنین چون کاربری این سیستم‌ها بیشتر در دست متخصصان بوده است (یعنی یک متخصص به اجرای برنامه‌ها بر روی سیستم می‌پرداخت)، خرابی در برنامه‌ها عموماً به خاطر خرابی منابع اتفاق می‌افتاد. در این بین، مطالعات فو و زو [۴۱، ۴۶] تحلیل مناسبی نسبت به خرابی منابع در دو سیستم توزیع شده (خوشه محاسباتی LANL و سیستم مشبک WSU) ارائه می‌دهد. همچنین پیش‌بینی خرابی منابع توسط آن‌ها در این سیستم‌ها با استفاده از شبکه‌های عصبی نتایج قابل قبولی داشته است. در مطالعه زینعلی‌پور و همکارانش [۴۴] نیز نتایج خوبی در پیش‌بینی خرابی غیر برخط در یک سیستم مشبک با استفاده از حدود یک ماه از داده‌های ثبت وقایع آن حاصل شده است.

۳-۲- پیش‌بینی برخط خرابی منابع

در این قسمت مطالعاتی که در زمینه پیش‌بینی هرگونه مشکل و خرابی در منابع (سخت‌افزار و نرم‌افزار) بعد از شروع کار سیستم (برخط) انجام گرفته است بررسی می‌شود. در آخر این بخش نیز پژوهش‌هایی که به صورت ترکیبی به پیش‌بینی هرگونه خرابی (منابع و برنامه‌ها) پرداخته‌اند آورده شده است.

کستلی و همکارانش در سال ۲۰۰۱ [۳۸] روند پیری نرم‌افزار را در یک خوشه از سرویس دهنده‌های آی‌بی‌ام بررسی کردند. آن‌ها برای پیش‌بینی فرسودگی منابع نرم‌افزاری و خرابی آن‌ها، با استفاده از یک شبکه پاداش تصادفی (نوعی شبکه پتری) که ابزار قدرتمندی برای مدل‌سازی هم‌روندی است، یک روند خطی را بر زمان کار (سن) نرم‌افزار سیستم تطبیق دادند و توانستند پیری نرم‌افزار را با ۹۰٪ موفقیت پیش‌بینی نمایند.

همچنین لی و همکارانش در سال ۲۰۰۲ [۳۹] پارامترهای مختلفی مانند فضای مبادله مورد استفاده را از یک سرویس دهنده وب آپاچی جمع‌آوری کردند و برای پیش‌بینی روند افزایش استفاده از منابع نرم‌افزاری سیستم که منجر به فرسودگی و خرابی آن می‌شود به کار بردند.

19- BDII
20- GStat
21- GridICE
22- SmokePing
23- High Performance Computing (HPC)

آن‌ها برای پیش‌بینی این روند، از مدل خودهمبسته میانگین متحرک که معمولاً برای سنجش داده‌های سری زمانی در علم پردازش سیگنال مورد استفاده قرار می‌گیرد، استفاده نمودند و با برآورد زمان فرسودگی منابع، خرابی را پیش‌بینی کردند. آن‌ها روش خود را با روش کستلی و همکاران مقایسه کردند و نشان دادند که در مجموعه داده‌های آن‌ها، مدل‌سازی پیشنهادی منجر به پیش‌بینی‌های دقیق‌تر شده است.

برنجی و همکارانش در سال ۲۰۰۳ [۴۰] از یک قاعده تابع پایه شعاعی استفاده کردند تا طبقه‌بندی نمایند که آیا یک مؤلفه سخت‌افزاری معیوب است یا خیر. در مدل آن‌ها یک سیگنال تشخیصی بر اساس مقادیر ورودی و خروجی اجزای مختلف، از صفر (بدون خطا) تا ۱ (معیوب) محاسبه می‌شود. پایه قاعده به صورت الگوریتمی مشتق شده از خوشه‌بندی داده‌های آموزش است که شامل جفت مقادیر ورودی / خروجی برای هر دو مورد معیوب و بدون خطا می‌باشد. داده‌های آموزش از مدل‌های شبیه‌سازی اجزای سخت‌افزاری که سعی می‌کنند رفتار ورودی / خروجی اجزای سیستم را تقلید کنند تولید می‌شوند. سپس، رویکرد مشابه در سطح سلسله‌مراتب بعدی استفاده می‌شود تا مدل‌های تشخیصی سرتاسر سیستم به دست آید. رویکرد نویسندگان مقاله در نهایت در یک مرکز تحقیقاتی ایمن ناسا استفاده شد.

هکر و همکارانش در سال ۲۰۰۹ [۴۵] به مطالعه خرابی گره‌ها در کامپیوترهای بلوچین شرکت آی‌بی‌ام^{۲۴} پرداختند. آن‌ها فایل‌های ثبت وقایع این سیستم را به جهت توصیف آماری خرابی‌ها و ارائه مدلی برای پیش‌بینی احتمال خرابی گره‌ها تجزیه و تحلیل نمودند. سیستم بلوچین دارای امکانات گسترده‌ای در ثبت وقایع رویدادها دارد و تمامی ویژگی‌ها و نتایج رویدادها را در یک پایگاه داده DB2 ذخیره می‌کند. فایل‌های ثبت وقایع رویدادها در این سیستم شامل خطاهای سخت‌افزاری و نرم‌افزاری، خطاهای شبکه و اطلاعات مربوط به پایان کارها هستند که ویژگی‌هایی مانند

24- IBM

زمان رویداد، شدت رویداد (اطلاع، هشدار، شدید، خطا، مهلک و خرابی)، شرح دهنده نوع رویداد و مکان رویداد را در بر می‌گیرد. نویسندگان این مقاله با مطالعه داده‌های ثبت وقایع دریافتند که سازوکارهای حاکم بر رویدادها و خرابی‌ها از یک توزیع وایبول پیروی می‌کند. همچنین آن‌ها از مدل نیمه مارکوف برای اختصاص دادن میزان قابلیت اطمینان به گره‌ها و پیش‌بینی خرابی آن‌ها استفاده نمودند. با استفاده از نتایج این پیش‌بینی مدیران سیستم می‌توانند گره‌های خراب را شناسایی و برای اصلاح آن‌ها اقدام کنند؛ همچنین می‌توانند از نتایج رتبه‌بندی قابلیت اطمینان گره‌ها به منظور متعادل کردن اندازه صف‌ها استفاده کنند و زمان‌بند کارها را طوری تنظیم نمایند که کارهای کوتاه‌تر را به گره‌هایی با قابلیت اطمینان کمتر بسپارد و بالعکس، که این باعث کاهش خرابی برنامه‌ها نیز می‌شود.

بالدونی و همکارانش در سال ۲۰۱۲ [۴۸] یک معماری به نام کَاسپر^{۲۵} برای نظارت و پیش‌بینی خرابی‌های نرم‌افزاری در یک سیستم کنترل ترافیک هوایی واقعی ارائه دادند. این رویکرد، ترکیبی از پردازش رویداد پیچیده و مدل پنهان مارکوف است تا بتواند علائم خرابی را که ممکن است در قالب شرایط غیرعادی در معیارهای عملکرد خود را نشان دهد تجزیه و تحلیل نماید. سیستم کنترل ترافیک هوایی هدف مجموعه‌ای از چند مجموعه گره پردازشی است که از طریق شبکه محلی (LAN) متصل هستند و توسط برنامه‌های مبتنی بر میان‌افزار کوربا اجرا می‌شوند و داده‌های ثبت وقایع مورد نیاز هم از پرونده‌های گذشته و هم در حین اجرا به‌دست می‌آیند. نتایج به دست آمده توسط نویسندگان نشان داده است که روش پیشنهادی درصد کمی خطا دارد و به خاطر سرعت بالای پیش‌بینی، سیستم اجازه دارد تا روش‌های بهبودی مناسبی را اعمال کند.

آگراوال و همکارانش در سال ۲۰۱۵ [۵۶] بر روی مطالعه رویدادهای خطا به منظور پیش‌بینی خرابی گره‌ها در خوشه‌های هادوپ تمرکز کردند. هادوپ یک چارچوب

نرم‌افزاری متن‌باز بر اساس نگاشت-کاهش است که برای کار با کلان داده‌ها استفاده می‌شود [۶۹]. نویسندگان مقاله از مدل‌های مخفی مارکوف (HMM) برای آموزش ویژگی‌های پیام‌های ثبت وقایع و پیش‌بینی خرابی استفاده کرده‌اند. استفاده از HMM از این جهت مناسب است که پیام‌های خطا در دسترس هستند اما دانشی نسبت به خرابی وجود ندارد (مخفی است). برای پیاده‌سازی روش پیشنهادی از فایل‌های ثبت وقایع به دست آمده از یک خوشه شخصی با ۱۱ گره استفاده شده است. در روش پیشنهادی تمام داده‌های ثبت وقایع مرتبط با هادوپ یعنی داده‌های ثبت شده گره‌های داده و گره‌های نام استفاده می‌شود. قبل از هر چیز داده‌های ثبت وقایع باید پیش‌پردازش شوند و پیام‌های غیر مفید و نوفه‌دار حذف گردند. هر پیام شامل اطلاعات مفیدی نظیر برچسب زمانی، وضعیت خطا، نوع خطا، شناسه گره، شناسه کاربر و غیره است؛ اما برای آن که پیام‌ها به صورت داده‌های ورودی برای مدل HMM در آیند باید به دنباله‌ای از خطا تبدیل شوند. برای ایجاد این دنباله، از یک الگوریتم خوشه‌بندی بر روی داده‌پیام‌ها استفاده می‌شود، سپس هر خوشه به دست آمده به عنوان یکی از انواع خطاها انتخاب می‌شود و برچسب e با شماره مربوطه را می‌گیرد و به این ترتیب دنباله‌ای از خطاهای e به دست می‌آید که برای مدل یادگیری HMM مناسب هستند که در داده‌های ثبت وقایع به کار رفته، از روش خوشه‌بندی K میانگین استفاده شده و ۶ نوع خطا (از e1 تا e6) به دست آمده است. با استفاده از داده‌های آموزش برچسب شده‌ای که به شکل دنباله‌ای از خطاها حاصل شده‌اند، روش HMM برای ارزیابی توالی درست‌نمایی بیشینه که برای به‌روزرسانی پارامترهای مدل استفاده شده است به کار می‌رود. سپس با تحلیل الگوهای به دست آمده، تقلیدی از آن برای پیش‌بینی آینده به دست می‌آید. نتایج آزمایش‌ها نیز نشان داده است که روش پیشنهادی قادر به پیش‌بینی خرابی با دقت ۹۱٫۲۵٪ و حساسیت ۹۱٪ می‌باشد. پیتاکرات و همکارانش در سال ۲۰۱۷ [۶۵] برای غلبه بر پیچیدگی‌های موجود در پیش‌بینی خرابی در سیستم‌های

توزیع شده، یک رویکرد سلسله مراتبی پیش‌بینی برخط خرابی به نام هورا^{۲۶} پیشنهاد کردند. ایده اصلی آن‌ها ترکیب مدل‌های معماری با تکنیک‌های پیش‌بینی خرابی اجزاء سیستم بوده است. مدل پیشنهادی دو جزء اصلی دارد؛ ADM که وابستگی بین اجزاء سیستم را به دست می‌آورد و FPM که از شبکه‌های بیزین برای نشان دادن مسیرهای خرابی با در نظر گرفتن احتمالات استفاده می‌کند. اجزاء مورد بررسی شامل متعادل کننده بار (LB)، PT1 و PT2 (در لایه نمایش)، BT1، BT2 و BT3 (در لایه کسب‌وکار) و در نهایت پایگاه داده (DB) و ویژگی‌های مربوطه شامل مصرف پردازنده، مصرف حافظه و مصرف هرم هستند. آن‌ها مدل پیشنهادی خود را روی سرویس دهنده نرم‌افزاری نتفلیکس پیاده‌سازی نمودند و در نتایج پیش‌بینی به حساسیت بالای ۸۰٪ و تشخیص بالای ۴۰٪ دست پیدا کردند.

برخی از پژوهش‌ها نیز به ارائه روشی برای پیش‌بینی هرگونه رویداد خرابی (هم منابع و هم برنامه‌ها) پرداخته‌اند؛ مثلاً لیانگ و همکارانش در سال ۲۰۰۷ [۴۲] به مطالعه فایل‌های ثبت وقایع متعلق به ۱۴۲ روز از سیستم‌های بلو جین/ال پرداختند. تمرکز اصلی آن‌ها روی پیش‌بینی هرگونه رویداد مهلك یا خرابی در سیستم مورد بررسی بوده است. در این سیستم، فایل‌های ثبت وقایع شامل ویژگی‌های مختلف مرتبط با رویدادهای اتفاق افتاده در سیستم مانند حالت رویداد (اطلاع، هشدار، شدید، خطا، مهلك و خرابی)، برچسب زمانی مرتبط با رویداد و مکان رویداد هستند. در واقع هدف مقاله این بوده است که ابتدا ویژگی‌های مفید در پیش‌بینی را از میان ویژگی‌های موجود رویدادها استخراج کند و سپس از الگوریتم‌های طبقه‌بندی در یادگیری ماشین برای پیش‌بینی حالت مهلك یا خرابی رویدادها استفاده نماید. ایده اصلی پژوهش این است که زمان فعالیت سیستم به چند بازه زمانی (پنجره پیش‌بینی) تقسیم شود و در هر بازه سعی شود تا با استفاده از ویژگی‌های مرتبط با رویدادهای انجام شده در آن بازه،

رویدادهای مهلك بازه بعدی پیش‌بینی شود (از آنجایی که رویداد خرابی در نتیجه رویداد مهلك اتفاق می‌افتد، پیش‌بینی رویداد مهلك منجر به پیش‌گیری از خرابی خواهد شد). بنابراین پیش‌بینی انجام شده یک پیش‌بینی برخط برای طبقه‌بندی رویدادها به دو حالت مهلك و غیر مهلك می‌باشد که الگوریتم‌های طبقه‌بندی مورد استفاده شامل هرس افزایشی تکراری برای کاهش خطا، ماشین بردار پشتیبان و نزدیک‌ترین همسایه هستند. نویسندگان مقاله نهایتاً روش خود را آزمایش نمودند و به صحت حدود ۵۰٪ و حساسیت حدود ۸۰٪ دست پیدا کردند.

به‌علاوه سالفنر و مالک در سال ۲۰۰۷ [۳۷] تلاش نمودند تا یک پیش‌بینی برخط خرابی بر اساس خطاهای رویدادمحور در داده‌های میدانی یک سیستم ارتباطی تجاری ارائه دهند. در روش پیشنهادی آن‌ها از مدل نیمه مارکوف مخفی برای کشف الگوهای خرابی با در نظر گرفتن زمان وقوع اشتباه و نوع رویداد اشتباه استفاده می‌شود. این ترکیب منجر به درک الگوهای خطا می‌گردد. با این رویکرد، وظیفه پیش‌بینی شکست به یک مسئله تشخیص الگو تبدیل می‌شود. هافمن و همکارانش نیز در سال ۲۰۱۱ [۴۷] دو روش مدل‌سازی داده محور برای نظارت بر خطاها و پیش‌بینی هرگونه خرابی در یک سیستم ارتباطی تجاری ارائه دادند. یک مدل زنجیره مارکوف و یک تابع پایه شعاعی عمومی که ترکیب این دو مدل منجر به پیش‌بینی وقوع خرابی به عنوان تابعی از اندازه‌گیری گسسته و پیوسته متغیرهای سیستم شده است. داده‌ها شامل پرونده‌های ثبت وقایع مبتنی بر رویداد و زمان اندازه‌گیری مداوم حالات سیستم هستند. آن‌ها در نهایت با آزمایش و مقایسه روش پیشنهادی خود با مدل خودهمبسته میانگین متحرک ساده و خطی، برتری رویکرد خود را نشان دادند.

همچنین فو و همکارانش در سال ۲۰۱۲ [۴۹] با تأکید بر تفاوت فایل‌های ثبت وقایع رویدادها با دیگر داده‌های تراکنشی و وابستگی شدید داده‌های ثبت وقایع به نظم

زمانی، تلاش کردند تا مجموعه‌ای از الگوریتم‌های کشف روابط بین رویدادها را در قالب یک چارچوب پیش‌بینی خرابی به نام لاگ‌مستر^{۳۷} ارائه دهند. رویدادها در این سیستم ۹ مشخصه اصلی دارند شامل برجسب زمانی، شناسه ثبت، شناسه گره، شناسه رویداد، درجه شدت، نوع رویداد، نام برنامه، شناسه پردازش و شناسه کاربر. شدت رویداد شامل ۵ حالت اطلاع، هشدار، اشتباه، خطا و خرابی است و نوع رویداد شامل سخت‌افزار، سیستم، برنامه، فایل سیستم و شبکه می‌باشد. چارچوب پیشنهادی سه رکن اصلی دارد به نام‌های مأمور ثبت، سرویس دهنده ثبت و پایگاه داده ثبت. مأمور ثبت در هر گره، فایل‌های ثبت وقایع را جمع‌آوری و پیش‌پردازش می‌کند، رویدادهای تکراری و دوره‌ای را پالایش می‌نماید، قالب‌های مختلف آن‌ها (چون وقایع از منابع متفاوتی به دست می‌آیند پس قالب متفاوتی نیز دارند) را به مجموعه ۹ ویژگی تبدیل می‌کند و سپس آن‌ها را برای سرویس دهنده به جهت کشف قوانین رویدادها می‌فرستد که این داده‌ها در پایگاه داده ثبت ذخیره می‌شوند. در این زمان سرویس دهنده ثبت مجموعه‌ای از نمودارهای همبستگی رویدادها (ECG) را برای نشان دادن روابط بین رویدادها ایجاد می‌کند که در نهایت به پیش‌بینی خرابی آن‌ها ختم می‌شود. برای پیش‌پردازش و حذف رویدادهای تکراری یا دوره‌ای از یک الگوریتم آماری ساده استفاده می‌شود. به این صورت که ابتدا فاصله زمانی بین رویداد فعلی و رویداد پیشین که شناسه ثبت یکسان دارند اندازه‌گیری می‌شود. اگر رویداد فعلی فاصله زمانی کمتری از یک مقدار از پیش تعریف شده داشته باشد، به عنوان یک رویداد تکراری قلمداد شده و حذف می‌گردد. برای رویدادهای دوره‌ای نیز، برای هر شناسه ثبت، تعداد رویدادها در دوره‌های زمانی مختلف شمرده می‌شود و هر کدام که از یک مقدار آستانه بیشتر باشد به عنوان یک چرخه تکراری در نظر گرفته و تنها یک رویداد برای هر چرخه ثبت می‌شود. برای کشف همبستگی‌های بین رویدادها از الگوریتم‌های کشف وابستگی مبتنی

بر آپیوری استفاده می‌شود. به این صورت که ابتدا فایل‌های ثبت وقایع سیستم در مدت زمان مشخص به اندازه یک پنجره زمانی در میانگیر ذخیره می‌شود و سپس این داده‌ها تجزیه و تحلیل می‌گردند تا قوانین رویدادها استخراج شوند. نویسندگان مقاله روش پیشنهادی خود را بر روی فایل‌های ثبت وقایع سه سیستم توزیع شده مقیاس بزرگ شامل کامپیوترهای بلوچین/ال، خوشه‌های هادوپ در مرکز تحقیقاتی موبایل چین و خوشه‌های محاسباتی آزمایشگاه ملی لوس آلاموس (LANL) پیاده‌سازی نمودند و به ترتیب به صحت ۸۳٫۶۶٪، ۸۱٫۱۹٪ و ۷۹٫۸۲٪ برای پیش‌بینی رویدادهای مهلك و خرابی دست پیدا کردند.

به‌طور کلی، در بین کارهایی که به پیش‌بینی خرابی منابع می‌پردازند، پیش‌بینی برخط به علت در دسترس بودن اطلاعات بیشتر برای مدل‌سازی، گسترش بیشتری داشته است. در این بین، هکر و همکارانش [۴۵] که بر روی داده‌های ثبت وقایع سیستم بلوچین مطالعه کرده‌اند، تحلیل‌های آماری جامعی از سیستم مورد بررسی ارائه داده‌اند و این به خاطر کامل بودن اطلاعات موجود در پرونده‌های ثبت وقایع مورد مطالعه است. مطالعه آگراوال و همکارانش [۵۶] نیز از این جهت که جزء معدود کارهایی است که در زمینه پیش‌بینی خرابی منابع سخت‌افزاری (گره) در هادوپ پرداخته است، مورد توجه است. اما، با این که نتایج مناسبی از پیش‌بینی حاصل گردیده است، لازم می‌باشد که روش پیشنهادی بر روی یک خوشه واقعی آزمایش گردد. همچنین چارچوب پیشنهادی فو و همکارانش [۴۹] که تمام جنبه‌های خرابی را در نظر می‌گیرد و بر روی سه سیستم توزیع شده معتبر نیز آزمایش گردیده است، مدل خوبی است. البته نتایج خیلی بالایی برای پیش‌بینی حاصل نشده است؛ اما به علت جامع بودن پیش‌بینی، نتایج قابل قبولی است.

۳-۳- پیش‌بینی غیر برخط خرابی برنامه‌ها

در این قسمت مطالعاتی که در زمینه پیش‌بینی هرگونه مشکل و خرابی در برنامه‌ها (کارها و وظیفه‌ها) قبل از

اجرای برنامه (غیر برخط) انجام گرفته است بررسی می‌شود. نقطه قوت پیش‌بینی غیر برخط خرابی‌ها این است که می‌تواند از اجرای هر برنامه مشکل‌دار قبل از شروع جلوگیری کند و در نتیجه منابع بسیاری را حفظ نماید. لازم به ذکر می‌باشد که اگر پژوهشی شامل پیش‌بینی در هر دو حالت غیر برخط و برخط بوده نیز در این قسمت قرار گرفته است.

چاولا و همکارانش در سال ۲۰۰۸ [۴۳] تلاش نمودند تا یک چارچوب کشف دانش برای عیب‌یابی در سیستم‌های مشبک ارائه دهند تا بتواند طبیعت کارهای شکست خورده و موفق را درک کند و به پیش‌بینی خرابی کارها بپردازد. آن‌ها یک وبگاه ارائه دادند که به کاربر این امکان را می‌دهد تا با بارگذاری فایل‌های ثبت وقایع مرتبط با کارهای خود بتواند نمایش قابل فهمی از آمارها و ویژگی‌های سیستم و کارهایش داشته باشد. فایل‌های ثبت وقایع کاربر همچنین می‌تواند برای تحلیل کارها و مدل‌سازی خرابی و کشف قوانین مرتبط با خرابی استفاده شود که به پیش‌بینی خرابی منجر می‌گردد. تمرکز اصلی مقاله بر روی سیستم‌های مشبکی است که از چارچوب نرم‌افزاری کندر^{۲۸} استفاده می‌کنند. کندر یک چارچوب نرم‌افزاری متن‌باز برای به انجام رساندن موازی وظایف توزیع شده در شبکه‌ای از رایانه‌های مختلف (حتی رایانه‌های رومیزی بیکار) است [۷۰]. در سیستمی که از این چارچوب استفاده می‌کند، هر کار یا ماشین توسط یک ساختار داده متشکل از جفت نام-مقدار توصیف می‌شود که شامل حدود ۱۰۰ ویژگی مختلف است که می‌توانند با استفاده از تکنیک‌های یادگیری ماشین به جهت استخراج ویژگی‌های کار یا ماشین که مرتبط با خرابی هستند به کار گرفته شوند. منظور از خرابی کار در این مقاله، هر کاری است که صراحتاً توسط میزبان از اجرا خارج شود و یا در پایان وضعیت غیر از موفق (صفر) داشته باشد. نویسندگان مقاله در نهایت به اجرای دو الگوریتم طبقه‌بندی درخت C4.5 و هرس افزایشی تکراری برای کاهش خطا بر روی بارکاری‌های سیستم

مشبک محاسباتی دانشگاه پردو (تراگرید) که از چارچوب نرم‌افزاری کُنذر استفاده می‌کند پرداختند تا خرابی کارها را یک بار با استفاده از ویژگی‌های ماشین‌ها و بار دیگر با استفاده از ویژگی‌های کارها پیش‌بینی نمایند.

سعادت‌فر و همکارانش در سال ۲۰۱۲ [۵۰] با تمرکز بر مسئله لزوم وجود مدیریت پیشگیرانه در سیستم‌های مشبک به این دلیل که محیط‌های مشبک به علت طبیعت پویا و پیچیده خود دچار خرابی زیادی در اجرای کارها می‌شوند، به مطالعه ویژگی‌های بارکاری‌ها، سخت‌افزار و کارها در سیستم مشبک آورگرید و پیش‌بینی خرابی کارها با استفاده از فایل‌های ثبت وقایع سیستم پرداختند. آن‌ها پس از تشریح داده‌های ثبت وقایع موجود، ویژگی‌های مختلف را از لحاظ آماری بررسی کردند و الگوهای مختلف خرابی کارها را کشف نمودند. تجزیه و تحلیل زمانی کارها نشان داد که توزیع خرابی کارها در ماه‌های سال متفاوت است و همچنین در ساعت ۸ صبح الی ۸ شب خرابی بیشتری در کارها وجود دارد که نشان دهنده تأثیر شلوغی سیستم بر خرابی کارها می‌باشد. بررسی صف اجرای کارها بیان کرد که با افزایش اندازه صف، خرابی کارها افزایش می‌یابد. بررسی‌ها همچنین نشان داد که با افزایش زمان درخواستی برای اجرای کارها خرابی کاهش می‌یابد. نویسندگان مقاله یک پارامتر سخت‌افزاری جدید نیز برای کارها به نام شدت وابستگی به پردازنده معرفی نمودند که از نسبت زمان اجرای کار در پردازنده‌های سیستم به کل زمان اجرای کار به دست می‌آید و افزایش این مقدار وابستگی باعث افزایش خرابی می‌شود. آن‌ها همچنین نشان دادند که میزان خرابی در وبگاه‌های مختلف متفاوت است و هرچه پردازنده‌های بیشتری در یک سایت آزاد باشند خرابی کاهش می‌یابد. آن‌ها با استفاده از تجزیه و تحلیل‌های انجام شده، تعداد ۹ ویژگی مفید مرتبط با کارها را از میان ویژگی‌های موجود در داده‌های ثبت وقایع برای پیش‌بینی خرابی انتخاب کردند که شامل زمان ثبت، مدت زمان انتظار، مدت زمان اجرا، متوسط طول زمان استفاده از

پردازنده، زمان اجرای درخواستی، شناسه وبگاه، شناسه گروه، شناسه صف و شناسه بخش می‌باشند. همچنین ۵ ویژگی جدید را از مشخصات کارها استخراج کردند که در فرایند پیش‌بینی مفید هستند شامل ساعت خروج کار، شدت وابستگی به پردازنده، میزان حافظه استفاده شده، درصد پردازنده اشغال‌شده در شبکه و درصد بار صف. آن‌ها نهایتاً با استفاده از الگوریتم یادگیری شبکه‌های بیزین به پیش‌بینی خرابی کارها در سیستم مورد بررسی پرداختند. همچنین با آزمایش اندازه‌های مختلف برای پنجره آموزش نشان دادند که اگر برای آموزش طبقه‌بندی کارها از داده‌های موجود تا ۵۰ روز قبل ثبت کار استفاده گردد نتیجه بهینه حاصل می‌شود. میزان دقت ۹۶٪ نتیجه ثبت شده برای پیش‌بینی خرابی در این مقاله است. همچنین سعادت‌فر و دلداری [۵۲] در ادامه کار گذشته خود [۵۰] به طراحی یک زمان‌بند کار برپایه پیش‌بینی خرابی کارها پیش از اجرای آن‌ها پرداختند. آن‌ها از الگوریتم شبکه‌های بیزین برای آموزش طبقه‌بندی در داده‌های ثبت وقایع خوشه لوس آلاموس استفاده نمودند. ویژگی‌های به کار رفته در پیش‌بینی شامل مشخصات کار (ساعت ورود، حافظه درخواستی و عنصر پردازشی)، مشخصات کاربر (شناسه کاربر) و شرایط سیستم (بارکاری سیستم و نرخ در دسترس بودن گره‌ها) بودند.

ال‌سید و شرودر در گزارش علمی خود در سال ۲۰۱۵ [۵۵] به مطالعه خرابی کارها در خوشه‌های کامپیوتری بزرگ پرداختند و تلاش کردند تا با استفاده از فاکتورهای مؤثر مانند پارامترهای پیکربندی، منابع درخواستی، منابع مصرفی و غیره خرابی کارها را پیش‌بینی نمایند. داده‌های ثبت وقایع مورد مطالعه در این مقاله متعلق به خوشه‌های کامپیوتری شرکت گوگل و دانشگاه کارنگی ملون هستند. بخش تحلیل خرابی کارها که جهت انتخاب و استخراج ویژگی‌های مؤثر در پیش‌بینی انجام گرفته از مطالعه ارتباط بین وضعیت کارها با عوامل درجه موازی‌سازی، پارامترهای زمان‌بندی، مدت زمان و مصرف منابع

تشکیل شده است. مطالعه درجه موازی‌سازی که به تعداد وظیفه‌های کارها مربوط می‌شود نشان داده است که این عامل الگوهایی در خرابی کارها دارد؛ مثلاً در خوشه گوگل برای کارهای با اولویت دسته و تولید، بیش از ۹۰٪ کارهای تک وظیفه‌ای موفق بوده و بیش از ۹۰٪ کارهای چند وظیفه‌ای در نهایت کشته شده‌اند. یا در خوشه اوپن‌کلود (متعلق به دانشگاه کارنگی ملون) که حدود ۶۰٪ کارهای آن صفر یا یک وظیفه کاهش و بیش از ۸۰٪ کارهای آن حداقل دو وظیفه نگاشت داشته‌اند تفاوت چندانی در میزان موفقیت در بین کارهای تک وظیفه‌ای یا چند وظیفه‌ای نبوده است. مطالعه مدت زمان کارها نشان داده است که در هر دو خوشه کارهای موفق کوتاه‌ترین طول زمان و کارهای کشته شده طولانی‌ترین مدت زمان را داشته‌اند. پارامترهای زمان‌بندی شامل اولویت کار، رده زمان‌بندی و منابع درخواست شده می‌باشد که هر یک جداگانه بررسی شده است؛ مطالعه اولویت کارها که داده‌های مربوط به آن تنها در فایل‌های ثبت وقایع خوشه گوگل وجود داشته است، نشان داده که این پارامتر بر عدم موفقیت کارها بخصوص کشته شدن آن‌ها تأثیر بسزایی داشته است. اولویت وظیفه‌ها از اولویت تعیین شده در پیکربندی کارها نشئت می‌گیرد و از ۰ (کمترین) تا ۱۱ (بیشترین) متغیر است. وقتی که زمان‌بندی می‌خواهد وظیفه‌ها را بین ماشین‌های اجراکننده توزیع کند، وظیفه‌هایی که اولویت بالاتری دارند مقدم هستند، بنابراین در زمانی که خوشه منابع لازم برای اجرای تمام وظیفه‌ها را نداشته باشد وظیفه‌هایی که اولویت پایین‌تری دارند را می‌کشد. همچنین مطالعه داده‌ها نشان داده است که افزایش اولویت کارها باعث افزایش میزان موفقیت آن‌ها می‌شود. رده زمان‌بندی کارها مشخص‌کننده میزان حساسیت یک کار به تأخیر است و از عدد ۰ (کمترین حساسیت) تا ۳ (بیشترین حساسیت) مقدار می‌گیرد. مطالعه این پارامتر در خوشه گوگل نشان داده که رده زمان‌بندی ۰ و ۱ بیشترین استفاده را برای کارهای کاربران که اولویت تولید و دسته داشته‌اند بوده است که در کارهای تولید بالاتر رفتن عدد

رده زمان‌بندی عموماً باعث کاهش شانس موفقیت شده و در کارهای دسته روند عکس داشته است. میزان منابع درخواستی کارها نیز جزئی از تنظیمات پیکربندی است و مشخص می‌کند که یک کار نهایتاً اجازه مصرف چه میزان از منابع را دارد. مطالعه میزان منابع درخواستی نشان داده است که کارهای ناموفق (شکست خورده و کشته شده) میانه منابع درخواستی بیشتری در خوشه گوگل داشته‌اند. در کارهای خوشه اوپن‌کلود نیز میزان منابع درخواستی کارهای کشته شده بیشتر بوده است. بر اساس آمار گرفته شده میزان مصرف واقعی منابع که داده‌های مربوط به آن تنها در فایل‌های ثبت وقایع خوشه گوگل وجود داشته در کارهای شکست خورده بسیار بیشتر از کارهای موفق بوده است. نویسندگان مقاله بعد از انجام تحلیل‌های مورد نیاز برای کشف عوامل مؤثر بر خرابی کارها به پیش‌بینی موفقیت/شکست کارها پرداختند که برای انجام فرایند یادگیری از دو تکنیک طبقه‌بندی به نام‌های رگرسیون لجستیک چندجمله‌ای و درخت طبقه‌بندی و رگرسیون استفاده و در حالات مختلف پیاده‌سازی نمودند. آن‌ها پیش‌بینی مورد نظرشان را هم با استفاده از ویژگی‌هایی که قبل از شروع کار وجود دارند (مانند تنظیمات پیکربندی) و هم ویژگی‌هایی که بعد از شروع اجرای کار به دست می‌آیند (مانند منابع استفاده شده) به انجام رساندند و با استفاده از پارامترهای سنجش استاندارد شامل صحت، دقت، حساسیت و تشخیص مورد ارزیابی و مقایسه قرار دادند و نشان دادند که در مجموع الگوریتم کارت^{۲۹} توانایی بیشتری در پیش‌بینی خرابی کارها در سیستم‌های مورد نظر داشته است. آن‌ها همچنین در پژوهش بعدی خود [۶۳] به پیش‌بینی نتیجه اجرای وظیفه‌ها و کارها در دو حالت غیر برخط (پیش از شروع اجرا) و برخط (۵ دقیقه پس از شروع اجرا) در خوشه‌های گوگل، اوپن‌کلود و لوس آلاموس پرداختند. تفاوت اصلی این کار با مطالعه قبلی آن‌ها استفاده از الگوریتم جنگل تصادفی برای آموزش طبقه‌بندی بوده که نتایج پیش‌بینی خرابی را بهبود بخشیده

است. همچنین آن‌ها چارچوب پیشنهادی خود را روی خوشه کامپیوتری لوس آلاموس نیز به کار بردند. روزا و همکارانش در سال ۲۰۱۷ [۵۹] تلاش کردند تا مطالعه کامل‌تری نسبت به کارهای قبل خود [۵۳، ۵۴] روی فایل‌های ثبت وقایع خوشه‌های گوگل ارائه دهند. مقاله شامل دو بخش تحلیل اجراهای ناموفق و ارائه مدل برخط پیش‌بینی می‌باشد. اجراهای مورد بررسی مرتبط با کارها، وظیفه‌ها و رویدادها هستند. رویداد در اینجا به معنی وقایع مرتبط با وظیفه‌ها می‌باشد مثل رویداد ارسال که به معنی واگذاری یک وظیفه به خوشه برای اجرا است یا رویداد صف‌بندی که مرتبط با صف‌بندی وظیفه برای اجرا است. بنابراین کار مجموعه‌ای از وظیفه‌ها و وظیفه مجموعه‌ای از رویدادها است. پس وضعیت نهایی وظیفه‌ها وابسته به وضعیت رویدادها می‌باشد و به همین جهت در این مطالعه به تحلیل و پیش‌بینی وضعیت رویدادها به جای وظیفه‌ها پرداخته می‌شود. وضعیت نهایی کارها/وظیفه‌ها/رویدادها ۴ حالت مختلف دارد؛ اتمام (پایان موفقیت‌آمیز)، خرابی (پایان شکست خورده)، کشته شده (توقف توسط کاربر)، اخراجی (توقف توسط اجراکننده). در این مقاله به بررسی وضعیت آینده کارها در دو حالت موفق (اتمام) و ناموفق (مجموع حالات خرابی، اخراجی یا کشته شده) و وضعیت آینده رویدادها در چهار حالت ذکر شده پرداخته می‌شود. در بخش تحلیل، زمان پاسخ، زمان صف، زمان اجرا، هم‌زمانی ماشین (تعداد وظیفه‌هایی که روی ماشین مورد نظر در یک زمان اجرا شده)، اندازه کار (تعداد وظیفه‌ها)، محلی بودن ماشین (کسر تعداد ماشین‌هایی که رویدادهای کار را انجام داده‌اند روی اندازه کار) و غیره بررسی شده‌اند که منجر به انتخاب ویژگی‌های مناسب در فرآیند پیش‌بینی گردیده است. برای پیش‌بینی از مدل شبکه‌های عصبی در دو لایه استفاده شده است؛ به این صورت که ابتدا به صورت غیر برخط و با استفاده از ویژگی‌هایی که در زمان ورود کار تعیین می‌شوند مثل اندازه کار و میانگین اولویت وظیفه‌ها و منابع درخواستی (پردازنده، حافظه و دیسک) به طبقه‌بندی کارها به دو رده موفق و

ناموفق پرداخته شود، و سپس برای رویدادهای هر کار پیش‌بینی انجام شود که به کدام یک از ۴ رده اتمام، خرابی، کشته شده یا اخراجی تعلق دارد. برای پیش‌بینی رویدادها علاوه بر ویژگی‌های غیر برخط از ویژگی‌های مربوط به گذشته که بعد از شروع اجرا به دست می‌آیند مانند نوع رویدادهای گذشته (تعداد رویدادهای گذشته در ۴ حالت) و اطلاعات رویدادهای گذشته (اولویت، منابع درخواستی، هم‌زمانی ماشین و زمان صف و اجرا) هم استفاده شده است. در نهایت، سنجش روش پیشنهادی نشان داد که این روش می‌تواند با صحت ۹۴٫۴٪ و ۷۶٫۸٪ به ترتیب برای کارها و رویدادها پیش‌بینی انجام دهد.

لیو و همکارانش در سال ۲۰۱۷ [۶۱] مطالعه‌ای به منظور ارائه یک روش پیش‌بینی غیر برخط/برخط خرابی کارها در جریان‌های کاری خوشه‌های گوگل انجام دادند. تمرکز اصلی نویسندگان بر روی فرآیند یادگیری و نشان دادن برتری‌های مدل دستگاه یادگیری افراطی متوالی برخط که یک الگوریتم آموزش سریع افزایشی برخط بر پایه ماشین یادگیری افراطی است می‌باشد. در معماری روش پیشنهادی، فایل‌های ثبت وقایع کارهای ثبت شده و عملیات دسته‌ای، بیدرنگ جمع‌آوری و پاکسازی و سپس تحلیل‌های آماری انجام می‌شوند. طبقه‌بندی به دو رده پایان معمولی (موفق) و پایان غیر معمولی (مشکل‌دار؛ هر حالتی غیر از موفق) بعد از شروع اجرای کار و با استفاده از اطلاعاتی که در هر لحظه به روز می‌شوند انجام می‌گیرد. اگر پیش‌بینی شود که کار بدون مشکل انجام می‌گردد، کار مربوطه به اجرای خود ادامه می‌دهد؛ اما در صورت تشخیص خرابی، کار مربوطه قطع و دوباره ثبت می‌شود. در مرحله پیش‌پردازش، داده‌هایی که ویژگی‌های نامتعارف داشته باشند مثل کارهایی که قبل از بازه جمع‌آوری وقایع شروع شده‌اند یا بعد از این بازه پایان خواهند یافت و یا آن‌هایی که اطلاعاتشان گم شده است، حذف می‌گردند. از آنجایی که اطلاعات مرتبط به کارها محدود است علاوه بر ویژگی‌های کارها نظیر زمان صف‌بندی، رده زمان‌بندی و

تعداد وظیفه‌ها از داده‌های مربوط به ویژگی‌های وظیفه‌ها برای ورودی‌های مدل استفاده می‌شود که شامل میانگین و انحراف معیار اولویت و منابع درخواستی (پردازنده، حافظه و دیسک) می‌باشد. در مدل پیشنهادی مقاله، ابتدا یک پیش‌بینی غیر برخط پیش از اجرای کار انجام می‌شود و سپس در حین اجرای وظیفه‌های کار اطلاعات بروز شده و یک پیش‌بینی برخط با سرعت و دقت بالاتر انجام می‌گیرد. نویسندگان مقاله در نهایت مدل پیشنهادی خود را آزمایش کردند و ضمن مقایسه با مدل‌های ماشین بردار پشتیبان و ماشین بردار پشتیبان متوالی برخط نشان دادند که مدل با دقت ۹۳٪ و پایین‌ترین سرعت آموزش بهترین نتیجه را در پی دارد.

هانگیان و همکارانش در سال ۲۰۱۷ [۶۲] به منظور بررسی و پیش‌بینی خرابی‌های مرتبط با تنظیمات پیکربندی اشتهاب در کارها به مطالعه فایل‌های ثبت وقایع به جا مانده از کارهای اجرا شده در یک خوشه هادوپ پرداختند. خوشه مورد بررسی یک خوشه تحقیقاتی به نام اوپن‌کلود است که متعلق به دانشگاه کارنگی ملون می‌باشد. هدف، استفاده از ویژگی‌هایی که بعد از ورود و قبل از اجرای کار به دست می‌آیند به عنوان ورودی مدل یادگیری و به منظور طبقه‌بندی کارها به دو رده موفق و ناموفق (شکست خورده و کشته شده) است. ویژگی‌ها شامل دو نوع تنظیمات پیکربندی و معیارهای سیستم در زمان اجرا هستند. براساس داده‌های موجود تنها سه معیار سیستم را می‌توان به عنوان ورودی برای کارها در نظر گرفت که شامل مجموع تعداد کارهای موجود در سیستم در زمان ورود کار، مجموع تعداد وظیفه‌های موجود در سیستم در زمان ورود کار و اندازه حافظه رزرو شده سیستم در زمان ورود کار می‌باشند. اما ویژگی‌های اصلی که تمرکز مقاله هم روی آن‌ها است تنظیمات پیکربندی کارها هستند که شامل شناسه کار، تعداد وظیفه‌های نگاشت در کار، تعداد وظیفه‌های کاهش در کار، بیشترین تعداد تلاش مجاز برای وظیفه‌های نگاشت، بیشترین تعداد تلاش مجاز برای

وظیفه‌های کاهش، حافظه درخواستی برای وظیفه‌های نگاشت، حافظه درخواستی برای وظیفه‌های کاهش، فعال بودن اجرای گمانه‌زنی برای نگاشت‌ها (زمانی که در بین وظیفه‌های یک کار، یک وظیفه برخلاف سایر وظیفه‌ها اجرای کندی داشته و پس از گذشتن یک زمان مشخص نتیجه‌ای برنگردانده باشد، یک نسخه جدید از وظیفه مربوطه اجرا می‌شود و هر کدام که زودتر تمام شود خروجی آن استفاده شده و وظیفه دیگر کشته می‌شود، به این اجرای دوباره وظیفه اجرای گمانه‌زنی می‌گویند)، فعال بودن اجرای گمانه‌زنی برای کاهش‌ها و فعال بودن فشرده‌سازی برای خروجی نگاشت‌ها می‌باشد. در نهایت ۴ مدل طبقه‌بندی برای عملیات پیش‌بینی استفاده شده‌اند؛ درخت تصمیم گرادیان بوس‌تینگ، K نزدیک‌ترین همسایه، جنگل تصادفی و رگرسیون لجستیک که درخت تصمیم گرادیان بوس‌تینگ با دقت ۹۲٪، تشخیص ۷۸٪ و حساسیت ۵۲٪ به عنوان بهترین مدل انتخاب شده است.

سولها و همکارانش در سال ۲۰۱۸ [۶۴] به کار بر روی داده‌های ثبت وقایع به دست آمده از خوشه‌های گوگل در بازه زمانی یک ماهه (سال ۲۰۱۱) پرداختند. آن‌ها ابتدا اجرای برنامه‌ها در سطح کار و وظیفه را از منظر ویژگی‌های مختلف آن‌ها مانند مدت زمان اجرا، مدت زمان صف‌بندی، میزان منابع مورد استفاده، بارکاری ماشین‌ها و زمان انتظار مورد تجزیه و تحلیل قرار دادند و سپس الگوریتم‌های مدل خطی عمومی، جنگل تصادفی، شبکه عصبی، بوس‌تینگ، درخت و درخت شرطی را برای پیش‌بینی نتیجه اجرای یک وظیفه پیش از شروع آن استفاده نمودند که جنگل تصادفی با صحت ۹۷٫۴٪ و ارجاع ۹۶٫۲٪ بهترین نتیجه را در کار آن‌ها داشت. آن‌ها در نهایت نحوه کار خوشه گوگل را با استفاده از شبیه‌ساز کلودسیم^{۳۰} بازسازی کردند و روش پیشنهادی خود را روی آن اجرایی نمودند و نشان دادند که با پیش‌بینی خرابی وظیفه‌ها پیش از اجرا می‌توانند آمار وظیفه‌های کامل شده را تا ۴۰٪ بهبود ببخشند.

شتی و همکارانش در سال ۲۰۱۹ [۶۷] به تحلیل و

30- CloudSim

پیش‌بینی خرابی وظیفه‌ها در خوشه‌های گوگل پرداختند. آن‌ها از سه الگوریتم متفاوت شامل کم کردن تصادفی نمونه‌ها^{۳۱}، زیاد کردن تصادفی نمونه‌ها^{۳۲} و اس‌موت^{۳۳} برای متعادل‌سازی داده‌های آموزش استفاده نمودند و پس از به کارگیری رده‌بند تقویت‌گرایان ایکس برای یادگیری طبقه‌بندی، نتایج حاصل در حالات مختلف را با یکدیگر مقایسه کردند. آن‌ها در نهایت اثبات کردند که روش اس‌موت بهترین نتیجه را با صحت ۹۲٪ و ارجاع ۹۴٫۸٪ حاصل می‌نماید.

شیرزاد و سعادت‌فر نیز در سال ۲۰۲۰ [۶۸] فایل‌های ثبت وقایع به دست آمده از کارهای انجام شده در خوشه تحقیقاتی اوپن‌کلود متعلق به ده ماه از سال ۲۰۱۲ را مورد مطالعه قرار دادند. آن‌ها ابتدا به تحلیل ویژگی‌هایی از کارها پرداختند که با توجه به مطالعه قبلی آن‌ها [۷۱] دارای روندهایی در ایجاد خرابی برنامه‌ها بودند. سپس، ویژگی‌های موثر بر خرابی برنامه‌ها را انتخاب کردند و چند ویژگی جدید نیز با توجه به تحلیل‌های انجام شده، تعریف نمودند. در نهایت نیز چند مدل یادگیری مشهور را برای انجام پیش‌بینی خرابی برنامه‌ها پیش از اجرا به کار بستند و نشان دادند که برای داده‌های به کار رفته، درخت C5.0 بهترین نتایج را در بر دارد. آن‌ها در نهایت برتری مدل پیشنهادی خود را نسبت به کارهای مشابه نشان دادند.

همانطور که پیش‌تر نیز اشاره شد، توجه به پیش‌بینی خرابی برنامه‌ها با استفاده از داده‌های ثبت وقایع، در سال‌های اخیر محبوبیت بیشتری یافته است. با این حال، مطالعات کمتری به پیش‌بینی غیر برخط در این زمینه پرداخته‌اند. در این بین، سعادت‌فر و همکارانش مطالعات خوبی در زمینه استفاده از شبکه‌های بیزین در پیش‌بینی خرابی برنامه‌ها در سیستم‌های توزیع شده انجام دادند [۵۰، ۵۲] که در [۵۰] تحلیل خوبی از خرابی کارها در سیستم مشبک آورگرید ارائه گردیده است. همچنین در

میان مطالعاتی که در زمینه پیش‌بینی خرابی برنامه‌ها در خوشه‌های کامپیوتری انجام شده است، توجه ویژه‌ای به خوشه‌های گوگل وجود دارد که پرونده‌های ثبت وقایع آن به رایگان اشتراک‌گذاری گردیده است. در بین این مطالعات، کارهای [۵۵، ۵۹، ۶۴، ۶۷] به تحلیل و پیش‌بینی غیر برخط خرابی برنامه‌ها در این خوشه‌ها پرداخته‌اند. اما، مطالعه [۶۱] بیشتر بر روی فرایند پیش‌بینی تمرکز داشته و توجه کمتری به تحلیل خرابی نشان داده است. در کارهای [۵۵، ۶۲، ۶۸] نیز به تحلیل و پیش‌بینی خرابی کارهای هادوپ در خوشه تحقیقاتی اوپن‌کلود پرداخته شده است. در این بین، برتری مطالعه شیرزاد و سعادت‌فر [۶۸] در تعریف ویژگی‌های جدید با توجه به تحلیل ویژگی‌های موجود می‌باشد که تاثیر مثبتی در فرایند مدل‌سازی داشته و نتایج پیش‌بینی را بهبود بخشیده است.

۳-۴- پیش‌بینی برخط خرابی برنامه‌ها

در این قسمت مطالعاتی که در زمینه پیش‌بینی هرگونه مشکل و خرابی در برنامه‌ها (کارها و وظیفه‌ها) در حین اجرای برنامه (برخط) انجام گرفته است بررسی می‌شود. نقطه قوت پیش‌بینی برخط خرابی‌ها دقت بالاتر و احتمال خطای کمتر می‌باشد؛ همچنین روش‌های برخط شدت وابستگی کمتری به سیستم دارند و راحت‌تر می‌توانند در سیستم‌های مشابه پیاده‌سازی شوند.

چن و همکارانش در سال ۲۰۱۴ [۵۱]، تلاش کردند تا چارچوبی برای پیش‌بینی خرابی کارهای نگاشت-کاهش در خوشه‌های کامپیوتری گوگل با استفاده از فایل‌های ثبت وقایع آن ارائه دهند. نگاشت-کاهش یک مدل برنامه‌نویسی برای پردازش داده‌ها به صورت موازی و توزیع‌شده است؛ نگاشت به معنی تقسیم مسئله به مسائل کوچک‌تر توسط گره اصلی و توزیع این زیر مسائل در بین گره‌های کارگر است و کاهش به معنی دریافت جواب زیر مسائل از گره‌های کارگر و ساخت خروجی (جواب مسئله اصلی) توسط گره اصلی می‌باشد [۷۲]. در خوشه‌های گوگل که هدف انجام محاسبات ابری است از نگاشت-کاهش برای

31- Random Undersampling (RUS)

32- Random Oversampling (ROS)

33- SMOTE (Synthetic Minority Oversampling Technique)

کار با کلان داده‌ها استفاده می‌شود و به همین جهت خرابی برنامه‌ها زیاد اتفاق می‌افتد که باعث هدر رفتن منابع زیادی در این سیستم می‌گردد. بنابراین مطالعه و پیش‌بینی خرابی کارها و وظیفه‌ها در این‌گونه سیستم‌ها اهمیت بسزایی دارد. ویژگی‌هایی که در فرآیند پیش‌بینی این مقاله استفاده می‌شوند با توجه به تحلیل‌هایی که توسط نویسندگان مقاله در کار قبلی آن‌ها [۷۳] انجام گرفته است انتخاب شده‌اند و شامل این موارد هستند: ۱- اولویت وظیفه‌ها که یکی از محدودیت‌های زمان‌بندی است و تعیین می‌کند که یک وظیفه چه زمانی برای اجرا به گره داده شود؛ انواع کلی اولویت‌ها شامل تولید، دسته و آزاد هستند که طبق بررسی‌های انجام شده میزان خرابی در نوع دسته بیشتر بوده است. ۲- ثبت دوباره وظیفه‌ها در زمان حیات یک کار که این ویژگی مربوط به تعداد ثبت دوباره وظیفه‌های یک کار است؛ زمانی که یک وظیفه نگاشت یا کاهش خراب یا کشته می‌شود، با توجه به تنظیمات مشخص شده می‌تواند دوباره ثبت و اجرا شود که طبق بررسی‌ها تعداد ثبت دوباره وظیفه‌ها در کارهای شکست خورده خیلی بیشتر از کارهای تمام شده بوده است. ۳- استفاده از منابع که طبق بررسی‌ها تفاوت بسیار زیادی در میزان منابع مصرف شده بین وظیفه‌های پایان‌یافته و شکست خورده در یک کار وجود داشته است. ۴- پروفایل کاربر، همان‌طور که خوشه‌بندی نشان داده است مرکز گروه‌های پروفایل‌های کاربران با نسبت خرابی کارها به تمامی حالت‌های اتمام کارها همبستگی دارد. چارچوب پیشنهادی نیز شامل چهار مرحله می‌باشد؛ ۱- نظارت و ذخیره‌سازی معیارهای برنامه‌ها و سیستم، ۲- پردازش داده‌ها به قالب ساخت‌یافته حاوی اطلاعات مکانی و زمانی آن‌ها، ۳- پیش‌بینی خرابی با استفاده از تکنیک‌های یادگیری ماشین و ۴- مدیریت اصلاح خرابی براساس نتایج پیش‌بینی می‌باشد. معیارهای برنامه‌ها و سیستم از داده‌های ثبت وقایع به دست می‌آیند. هدف پیش‌پردازش داده‌ها این است که داده‌های جمع‌آوری شده را به ساختارهای لایه‌ای

برنامه محور که برای مدل‌های یادگیری مناسب باشد تبدیل کند. جداول اصلی شامل ویژگی‌های مختلف مرتبط با کارها، وظیفه‌ها، گره‌ها و کاربران هستند که در فایل‌های جداگانه ذخیره می‌شوند که باید برای دو سطح کار و وظیفه جداسازی و ادغام شوند؛ یعنی ویژگی‌های مورد نیاز برای طبقه‌بندی وظیفه‌ها در یک جدول و ویژگی‌های مورد نیاز برای طبقه‌بندی کارها در جدولی دیگر به جهت پیش‌بینی خرابی قرار بگیرند. در مرحله سوم یعنی مرحله پیش‌بینی تلاش می‌شود تا با استفاده از داده‌های زمانی/ مکانی در دو سطح ذکر شده به عنوان ورودی مدل‌سازی، خرابی یا موفقیت اجرای برنامه‌ها پیش‌بینی شود. برای پیش‌بینی از مدل شبکه‌های عصبی راجعه استفاده می‌شود که در مرحله آزمایش توانست به عدد ۸۴٪، ۸۶٪ و ۸۰٪ به ترتیب برای دقت، حساسیت و تشخیص در سطح پیش‌بینی خرابی وظیفه‌ها دست پیدا کند. اما برای پیش‌بینی وضعیت آتی کارها، نویسندگان مقاله سه زمان پیش‌بینی را در دو حالت محافظه کارانه و تهاجمی در نظر گرفتند؛ زمان پیش‌بینی شامل پیش‌بینی بعد از گذشت یک چهارم از زمان شروع کار، پیش‌بینی بعد از گذشت نیمی از زمان شروع کار و پیش‌بینی نزدیک به اتمام کار است. حالت محافظه کارانه مقدار صحت و تشخیص بالاتری دارد اما در حالت تهاجمی که پیش‌بینی خرابی اولویت دارد مقدار حساسیت بالاتر می‌باشد. مقدار به‌دست‌آمده برای صحت، حساسیت و تشخیص در حالت محافظه کارانه به ترتیب در حدود ۸۰٪ الی ۸۴ درصد، ۱۵٪ الی ۴۱ درصد و ۸۱٪ الی ۹۵ درصد؛ و در حالت تهاجمی به ترتیب در حدود ۳۵٪ الی ۴۵ درصد، ۴۵٪ الی ۷۰ درصد و ۳۰٪ الی ۴۲ درصد بوده است (در هر دو حالت هر چه پیش‌بینی دیرتر انجام شود نتایج بهتری حاصل می‌گردد). در مرحله چهارم چارچوب پیشنهادی، مدیریت خطای پیش‌گیرانه ساده‌ای پیشنهاد شده است که در آن کارهایی که وضعیت آن‌ها شکست پیش‌بینی می‌گردد باید کشته شوند؛ البته در این روش اگر پیش‌بینی اشتباه انجام شود هدر رفتن منابع را در پی دارد

چراکه کار تا حدی انجام شده است ولی کشته و دوباره اجرا می‌گردد. ایراد دیگری که متوجه چارچوب پیشنهادی می‌باشد این است که برای کارهایی که خیلی زود شکست می‌خورند کاربردی ندارد.

روزا و همکارانش در سال ۲۰۱۵ [۵۳] ضمن تأکید بر لزوم داشتن درک عمیق در مورد خرابی کارها به جهت کاهش تأثیرات منفی آن بر منابع سیستم و کارایی برنامه‌ها، به تحلیل و پیش‌بینی خرابی کارها در خوشه‌های گوگل پرداختند. کارهای کلان‌داده در مراکز داده‌ای مقیاس بزرگ مانند گوگل به خاطر ناهمگونی‌های سخت‌افزاری، وابستگی‌های پیچیده بین وظیفه‌ها، اولویت‌های متفاوت در اجرای کارها و سیستم زمان‌بندی سطح بالا بسیار پیچیده هستند و خرابی‌های متعددی در آن‌ها اتفاق می‌افتد. بنابراین برای جلوگیری از هدر رفتن منابع و اشغال بیهوده سیستم، پیش‌بینی خرابی کارها با استفاده از آموزش برخط براساس داده‌های گذشته بسیار مفید است. برای آموزش مدل از ویژگی‌های سیستمی و آماری کارها استفاده می‌شود. ویژگی‌های آماری شامل تعداد وظیفه‌ها، منابع درخواستی و اولویت تعیین شده هستند. از آنجایی که ویژگی‌های آماری برای وظیفه‌ها تعیین می‌شوند، برای تعداد وظیفه‌ها میانگین و برای میزان پردازنده، حافظه و دیسک درخواستی و اولویت مشخص شده میانگین و انحراف معیار در نظر گرفته می‌شود. طبق بررسی‌های انجام گرفته تفاوت‌های زیادی در ویژگی‌های ذکر شده بین کارهای موفق و شکست خورده وجود دارد مثلاً به‌طور متوسط کارهای شکست خورده تعداد بیشتری وظیفه اما با اولویت‌های پایین‌تر داشته‌اند یا در مورد منابع درخواستی کارهای موفق میزان حافظه و دیسک کمتری درخواست کرده بودند. بررسی انحراف معیارها نیز نشان می‌دهد که تفاوت در میان وظیفه‌های کارهای موفق کمتر بوده است. به‌علاوه ویژگی‌های سیستمی که همان ویژگی‌های مرتبط با بارکاری سیستم هستند شامل وارد شده، بازده و تعداد وظیفه‌ها می‌باشند که به ترتیب نشان دهنده افزایش بار،

کاهش بار و حالت لحظه‌ای بار می‌باشند. برای به دست آوردن این ویژگی‌ها، در یک بازه زمانی خاص (مثلاً یک ساعت) و برای هر اولویت به صورت جداگانه، تعداد وظیفه‌های ثبت شده در سیستم به عنوان وارد شده، تعداد کل وظیفه‌ها به عنوان تعداد وظیفه‌ها و تعداد وظیفه‌های خارج شده از سیستم (چه موفق چه ناموفق) به عنوان بازده سیستم در نظر گرفته می‌شود. کارها به دو گروه شکست خورده و موفق تقسیم‌بندی شده‌اند که با استفاده از ویژگی‌های آماری و سیستمی ذکر شده و با به کارگیری ۳ روش طبقه‌بندی تلاش می‌شود که پیش‌بینی گردد که هر کاری که در سیستم ثبت و اجرا می‌شود به کدام دسته تعلق دارد و در صورت پیش‌بینی خرابی، کار متوقف شود تا از هدر رفتن منابع جلوگیری گردد. روش‌های یادگیری شامل تحلیل افتراقی خطی، تحلیل افتراقی درجه دوم و رگرسیون لجستیک هستند. در نهایت نویسندگان مقاله با آزمایش روش‌های یادگیری در شرایط متفاوت (تفاوت در پنجره آموزش و ویژگی‌های استفاده شده) نشان دادند که روش استفاده شده می‌تواند تا ۶۷٫۷٪، ۴۷٫۱٪، ۴۰٫۵٪ و ۳۳٫۲٪ به ترتیب در منابع پردازنده، حافظه، دیسک و زمان محاسبه صرفه‌جویی کند.

روزا و همکارانش در مطالعه دیگری [۵۴] در ادامه بررسی و پیش‌بینی خرابی برنامه‌ها در خوشه‌های گوگل یک پیش‌بینی دو مرحله‌ای با استفاده از شبکه‌های عصبی برای پیش‌بینی خرابی کارها و سپس وظیفه‌ها پیشنهاد کردند. ایده اصلی این است که پس از پیش‌بینی خرابی یک کار، وظیفه‌هایی که آینده غیر موفقی دارند برای آن کار پیش‌بینی شوند که در این صورت وظیفه شناسایی شده متوقف و از هدر رفتن منابع جلوگیری می‌گردد. ویژگی‌های مورد استفاده برای پیش‌بینی همان ویژگی‌های ذکر شده در مقاله قبل [۵۳] بودند که برای کارها انحراف معیار و میانگین آن‌ها محاسبه شده‌اند. پیاده‌سازی روش پیشنهادی آن‌ها توانست به صحت بالایی دست پیدا کند و فقط ۱٪ از وظیفه‌های موفق را اشتباه پیش‌بینی نماید

و همچنین به میزان ۴۹٪ صرفه‌جویی در مصرف منابع ایجاد کند.

هانگیان و همکارانش در سال ۲۰۱۶ [۵۷] با مطالعه و بررسی فایل‌های ثبت وقایع مرتبط با وظیفه‌ها در خوشه‌های گوگل متوجه شدند که وظیفه‌هایی وجود دارند که بعد از گذشت مدت زمان کوتاهی از اجرا دچار خرابی می‌شوند و سپس برای اجرای مجدد در صف‌بندی قرار می‌گیرند اما باز هم شکست می‌خورند و دوباره اجرا می‌شوند که این رویه مرتباً اتفاق می‌افتد. نویسندگان مقاله این‌گونه وظیفه‌ها که به دفعات اجرا شده و شکست می‌خورند و باعث هدر رفتن منابع بسیار سیستم، ایجاد سربرار زمان‌بندی و ایجاد اختلال در عملکرد آن هستند را وظیفه‌های قاتل نامیدند که باید بیدرنگ از میان انبوهی از وظیفه‌های در حال اجرا شناسایی شوند. علت ایجاد وظیفه‌های قاتل این است که در خوشه‌های گوگل، سیاست برخورد با خرابی وظیفه‌ها زمان‌بندی و اجرای مجدد وظیفه شکست خورده است که تعداد این اجرای دوباره توسط کاربر تعیین می‌شود. بررسی توزیع خرابی وظیفه‌ها نشان داده است که تعداد کمی از وظیفه‌ها تعداد زیادی از خرابی را تجربه کرده‌اند که آن‌هایی که بیش از ۲۰۰ تکرار اجرا بدون موفقیت داشته‌اند به عنوان قاتل تعیین شده‌اند. وظیفه‌های اجرا شده ویژگی‌های مشخصی دارند (شامل رده‌ زمان‌بندی، اولویت، مدت زمان، وضعیت خاتمه و سرعت شکست) که الگوهای خاصی در میان وظیفه‌های قاتل پیدا کرده‌اند که در فرایند پیش‌بینی مؤثر هستند؛ مثلاً بیشتر وظیفه‌های قاتل در رده‌ زمان‌بندی سطح پایین (۰ و ۱) قرار دارند، دارای اولویت بالایی (تولید) هستند (یعنی وظیفه‌هایی که مرتب شکست می‌خورند اولویت بالاتری در اجرا دارند و باعث ایجاد اختلال در اجرای سایر وظیفه‌ها می‌شوند)، با این که مدت اجرای هر بار آن‌ها کوتاه است (۸۵٪ آن‌ها دارای سرعت شکست ۲۵۰۰ خرابی در ساعت هستند) و به سرعت بعد از شروع اجرا شکست می‌خورند اما به علت تعداد اجرای بسیار زیاد مدت زمان کلی اجرای

آن‌ها بسیار طولانی است و در نهایت هیچ وضعیت خاتمه موفقیت‌آمیزی نداشته و ۹۹٪ آن‌ها توسط کاربر یا مدیر کشته شده‌اند. همچنین بررسی میزان مصرف منابع در وظیفه‌های قاتل تفاوت‌های بسیاری با سایر وظیفه‌ها دارد؛ این وظیفه‌ها شکاف ناگهانی بسیاری در الگوی مصرف خود دارند (نوسانات بسیار بالا). برای شناسایی وظیفه‌های قاتل از الگوریتم‌های یادگیری ماشین استفاده می‌شود که با توجه به ویژگی‌های ذکر شده بخصوص الگوهای مصرف منابع بتواند پیش‌بینی نماید که از میان وظیفه‌های در حال اجرا کدام یک قاتل خواهد شد. نویسندگان مقاله دو نوع الگوریتم برای این منظور پیشنهاد دادند؛ یک روش یادگیری با ناظر به نام K-هانتز^{۳۴} بر اساس تکنیک طبقه‌بندی K نزدیک‌ترین همسایه و یک روش یادگیری بدون ناظر به نام C-هانتز^{۳۵} بر اساس تکنیک خوشه‌بندی K میانگین^{۳۶}. ایده اصلی K-هانتز این است که از این حقیقت که روند تغییرات مصرف منابع وظیفه‌های قاتل مشخصاً شبیه دیگر وظیفه‌های قاتل اما متفاوت از دیگر وظیفه‌ها است برای طبقه‌بندی به دو رده قاتل و غیر قاتل استفاده کند. اما در C-هانتز از خوشه‌بندی برای قرار دادن وظیفه‌های مشابه در یک گروه که با گروه دیگر تفاوت دارد استفاده می‌گردد که به خاطر تفاوت در ویژگی‌ها تمام وظیفه‌های قاتل در یک خوشه قرار می‌گیرند. نتایج پیاده‌سازی و ارزیابی نشان داده است که روش ارائه شده می‌تواند وظیفه‌های قاتل را با ۹۷٪ صحت شناسایی کند و همچنین با تشخیص سریع این‌گونه وظیفه‌ها و جلوگیری از اجرای مجدد آن‌ها باعث صرفه‌جویی ۸۸٪ از منابع سیستم گردد.

یو و همکارانش در سال ۲۰۱۶ [۵۸] به منظور مطالعه و پیش‌بینی خرابی کارها در خوشه‌های تحقیقاتی به بررسی داده‌های ثبت وقایع متعلق به خوشه تحقیقاتی مرکز محاسبات علمی پژوهشی انرژی ملی به نام جین‌پول پرداختند. تمرکز اصلی مقاله بر روی پیش‌بینی برخط خرابی کارها و بررسی‌های متعدد در مورد نتایج آن بوده

34- K-Hunter
35- C-Hunter
36- K-Means

و توجه چندانی به تحلیل دلایل خرابی نشده است. برای پیش‌بینی از ۱۳ ویژگی که در میان ویژگی‌های متعدد مرتبط با کارها در داده‌های ثبت وقایع وجود داشته استفاده شده است. این ویژگی‌های شامل ساعت دیواری (مدت زمان بین شروع و پایان یک وظیفه)، زمان پردازنده کاربر (مجموع زمانی که در سطح کاربر از هسته‌های پردازنده سپری شده)، زمان پردازنده سیستم (مجموع زمانی که در سطح سیستم از هسته‌های پردازنده سپری شده)، زمان پردازنده (مجموع زمان‌های پردازنده در سطح کاربر و سیستم)، بیشترین اندازه مجموعه ساکن (بیشترین مقدار حافظه مورد استفاده حین اجرا)، اصلاحات صفحه (اشتباهات نرم صفحه بدون در نظر گرفتن ورودی/خروجی)، اشتباهات صفحه (اشتباهات سخت صفحه با در نظر گرفتن ورودی/خروجی)، عملیات ورودی بلوک (تعداد دفعاتی که فایل سیستم باید ورودی را مهیا کند)، عملیات خروجی بلوک (تعداد دفعاتی که فایل سیستم باید خروجی را مهیا کند)، سوئیچ‌های ارادی زمینه (تعداد دفعاتی که سوئیچ ارادی برای زمینه انجام می‌شود)، سوئیچ‌های غیر ارادی زمینه (تعداد دفعاتی که سوئیچ غیر ارادی برای زمینه انجام می‌شود)، حافظه (استفاده از حافظه یکپارچه در حجم گیگابایت \times زمان پردازنده در ثانیه)، ورودی/خروجی (مقدار داده منتقل شده در عملیات ورودی/خروجی) هستند. از آن جایی که بسیاری از این ویژگی‌ها بعد از اتمام کار به دست می‌آیند ولی هدف پیش‌بینی برخط این است که در مدت زمان کوتاهی بعد از شروع اجرا بتواند خرابی را پیش‌بینی نماید، ویژگی‌های ذکر شده باید نسبت به ساعت دیواری یا زمان پردازنده نرمال‌سازی شوند (نتایج آزمون و خطا نشان داده که تفاوت چندانی در نرمال‌سازی نسبت به این دو ویژگی وجود ندارد). بنابراین یک مدت زمان مشخصی که از اجرای کار گذشت، ویژگی‌های مختلف در آن بازه زمانی استخراج شده و نرمال‌سازی می‌شوند و برای پیش‌بینی وضعیت ادامه کار استفاده می‌گردند. برای پیش‌بینی نیز از چند الگوریتم طبقه‌بندی مشهور به نام‌های

درخت تصمیم، جنگل‌های تصادفی، بیزین ساده، رگرسیون لجستیک و ماشین بردار پشتیبان استفاده شده که طبق مقایسه‌های اولیه جنگل‌های تصادفی نتایج بهتری نسبت به سایر الگوریتم‌ها داشته است. نویسندگان مقاله در نهایت عملیات طبقه‌بندی با استفاده از جنگل‌های تصادفی را در شرایط مختلف (عمق جنگل، تعداد درخت‌ها) و بازه‌های زمانی متفاوت (روزانه، هفتگی) آزمایش کردند و به صحت بالای ۹۰٪ (در یکی از موارد تا ۹۹٫۸٪) دست پیدا کردند. اسلام و مانیوانان در سال ۲۰۱۷ [۶۰] مطالعه‌ای به منظور بررسی دلایل خرابی برنامه‌ها در خوشه‌های گوگل و پیش‌بینی خرابی کارها و وظیفه‌ها با استفاده از فایل‌های ثبت وقایع سیستم ارائه دادند. درک داده‌های ثبت وقایع و تحلیل علت خرابی برنامه‌ها منجر به شناسایی ضعف‌های سیستم و همچنین استخراج ویژگی‌های مهم و مؤثر برای پیش‌بینی می‌شود. طبق بررسی انجام شده بیش از ۴۰ درصد کارها و وظیفه‌های سیستم مورد مطالعه پایان موفق نداشته‌اند و اکثر اجراهای ناموفق شکست خورده یا کشته شده بودند. مجموع مصرف پردازنده و حافظه در کارهای شکست خورده و کشته شده تقریباً سه برابر مصرف کارهای موفق بوده است که این قضیه نشان دهنده لزوم مطالعه و پیش‌بینی خرابی روی این سیستم است. تمرکز اصلی مقاله روی وضعیت موفق یا شکست خورده می‌باشد. بررسی وضعیت اتمام کارها و وظیفه‌ها نشان داده که به‌طور متوسط تعداد ثبت دوباره وظیفه‌ها در کارهای شکست خورده به مراتب بیشتر از موفق بوده است. همچنین کارها و وظیفه‌های موفق زمان انتظار در صف خیلی کمتری نسبت به سایر کارها و وظیفه‌ها داشته‌اند و همچنین مدت اجرای برنامه‌های موفق کمتر بوده است. میزان مصرف نیز در برنامه‌های موفق و ناموفق تفاوت داشته است. بنابراین براساس بررسی‌های انجام شده ۱۳ ویژگی برای ورودی‌های طبقه‌بندی انتخاب شدند که شامل متوسط پردازنده مصرفی، متوسط حافظه مصرفی، حافظه نهان صفحه نگاشت نشده، متوسط ورودی/خروجی،

متوسط دیسک مصرفی، شناسه کاربر، شناسه کار، شاخص کار، اولویت، رده زمان بندی، مدت زمان کار، تعداد ثبت دوباره وظیفه‌ها و تأخیر زمان بندی هستند. تعدادی از این ویژگی‌ها در زمان ثبت کار مشخص می‌شوند اما سایر ویژگی‌ها باید با توجه به زمان نرمال‌سازی گردند؛ به علاوه یک بازه زمانی (۳۰ ثانیه یا ۱ دقیقه) تعیین می‌گردد که از زمان شروع کار تا پایان بازه زمانی ویژگی‌های مربوطه اندازه‌گیری می‌شوند. از الگوریتم طبقه‌بندی حافظه طولانی کوتاه-مدت که براساس شبکه‌های عصبی راجعه طراحی شده و طبق بررسی نویسندگان نتایج بهتری نسبت به ماشین بردار پشتیبان و شبکه‌های عصبی پیش‌خور داشته برای فرآیند یادگیری استفاده گردیده است. برای پیش‌بینی در سطح کار طبقه‌بندی به دو کلاس موفق و شکست خورده و در سطح وظیفه به دو رده موفق و ناموفق (مجموع ۳ حالت خرابی، اخراجی و کشته شده) گرفته که به میزان ۸۷٪ دقت، ۸۵٪ حساسیت و ۸۹٪ تشخیص در سطح وظیفه و ۸۱٪ دقت، ۸۳٪ حساسیت و ۸۰٪ تشخیص در سطح کار دست پیدا کرده است.

به‌طور کلی می‌توان گفت که مطالعه بر روی خرابی‌های منابع، قدیمی‌تر هستند و در سال‌های اخیر پژوهشگران بیشتر به پیش‌بینی خرابی برنامه‌ها (کار و وظیفه) در سیستم‌های توزیع‌شده مقیاس‌بزرگ پرداخته‌اند. به‌عنوان نمونه در پژوهش‌های مورد بررسی، تنها مطالعه آگراوال و همکارانش [۵۶] و پیتاکرات و همکارانش [۶۵] مطلقاً به پیش‌بینی خرابی منابع در ۵ سال اخیر پرداخته است. علت این امر احتمالاً به پیشرفت‌های سخت‌افزاری و افزایش قابلیت اطمینان منابع در سیستم‌های توزیع‌شده و بخصوص سیستم‌های ابری برمی‌گردد. همچنین براساس مطالعات صورت گرفته، سیستم‌های توزیع‌شده مقیاس بزرگ بخصوص سیستم‌های محاسبات ابری استفاده گسترده‌ای در سال‌های اخیر پیدا کرده است و بسیاری از این کاربردها توسط کاربران عادی و غیرمتخصص انجام می‌گیرد که این موضوع باعث رخ دادن خرابی‌های

بیشتری در سطح برنامه‌ها به علت اشتباهات کاربری مانند تنظیمات اولیه نادرست می‌شود. پس در سال‌های اخیر نیاز به پیشگیری از خرابی برنامه‌ها بیشتر حس گردیده است. به‌علاوه در سال‌های اخیر مطالعات بسیاری بر روی خوشه‌های گوگل که زیرساخت محاسبات ابری هستند انجام شده است که این امر هم به خاطر به اشتراک‌گذاری عمومی و رایگان داده‌های ثبت وقایع این خوشه‌ها توسط شرکت گوگل و هم به خاطر حساسیت و اهمیت روزافزون سیستم‌های ابری به علت گستردگی استفاده از سرویس‌های آن می‌باشد. در این میان روزا و همکارانش در مجموع چند پژوهش [۵۳، ۵۴، ۵۹] مطالعه نسبتاً کاملی را در بحث پیش‌بینی خرابی برنامه‌ها در خوشه‌های ابری گوگل انجام دادند. همچنین با بررسی نتایج طبقه‌بندی در مطالعات مرور شده، می‌توان گفت که الگوریتم جنگل تصادفی و شبکه‌های عصبی (در صورت تنظیمات درست) عملکرد بهتری در پیش‌بینی خرابی در سیستم‌های ابری حاصل می‌نمایند. به‌طور کلی نیز پیش‌بینی برخط به علت پویایی بیشتر، نزدیک‌تر بودن داده‌های آموزش به شرایط عملی سیستم و به‌کارگیری ویژگی‌های بیشتر نتایج بهتری در پیش‌بینی خرابی حاصل می‌کند؛ اما پیش‌بینی غیر برخط به علت تشخیص خرابی پیش از شروع برنامه توانایی بیشتری در صرفه‌جویی منابع، زمان و هزینه‌های کاربر دارد و در نتیجه رضایت بیشتری حاصل می‌نماید.

۴- نتیجه‌گیری

در این مقاله به بررسی و مرور مطالعاتی که در سال‌های اخیر با استفاده از فایل‌های ثبت وقایع در زمینه پیش‌بینی خرابی منابع یا برنامه‌ها در سیستم‌های توزیع‌شده مقیاس بزرگ شامل خوشه‌های کامپیوتری و سیستم‌های مشبک انجام گرفته است پرداخته شد. خرابی‌های سخت‌افزاری، نرم‌افزاری، کارها و غیره از مشکلات همیشگی سیستم‌های توزیع‌شده بوده است که طبق آمار، در سیستم‌های مشبک و خوشه‌ای در دنیا، همیشه درصدی از خرابی (حتی تا

بیش از ۵۰ درصد) وجود دارد و موجب هدر رفتن منابع سیستم، ایجاد هزینه‌های مضاعف، پایین آمدن قابلیت اطمینان و از دست دادن مشتری می‌شود؛ به همین دلیل مطالعه دلایل خرابی و ارائه یک روش پیش‌بینی خرابی در این سیستم‌ها همواره یکی از موضوعات روز جهان بوده است. پیش‌بینی خرابی به مفهوم استفاده از روش‌های آماری و یادگیری ماشین به منظور تشخیص خرابی در آینده و پیش از وقوع آن با استفاده از سابقه و اطلاعات گذشته رویدادهای سیستم و برنامه‌ها می‌باشد.

مقالات بررسی شده متعلق به سال‌های ۲۰۰۱ الی ۲۰۱۹ بوده‌اند که به ارائه یک روش یا چارچوب برای پیش‌بینی خرابی‌های منابع (سخت‌افزار مانند پردازنده و حافظه و نرم‌افزار مانند سیستم‌عامل و فایل سیستم) و یا برنامه‌ها (کار و وظیفه) به صورت برخط و یا غیر برخط پرداخته‌اند. تفاوت پیش‌بینی برخط و غیر برخط در زمان انجام پیش‌بینی است. به این صورت که پیش‌بینی غیر برخط قبل از شروع کار سیستم یا اجرای برنامه انجام می‌شود و از ویژگی‌هایی به عنوان ورودی استفاده می‌کند که قبل از شروع به دست می‌آیند، اما پیش‌بینی برخط مدتی بعد از شروع کار سیستم یا اجرای برنامه به انجام می‌رسد و علاوه بر ویژگی‌های قابل جمع‌آوری قبل از شروع، از ویژگی‌هایی نیز استفاده می‌کند که بعد از شروع اندازه‌گیری می‌شوند.

مقالات مرور شده با توجه به نوع پیش‌بینی و زمان پیش‌بینی، طبقه‌بندی و بررسی شدند. تمرکز اصلی در مرور مقالات روی ایده اصلی روش، شیوه مطالعه و کشف ویژگی‌های مؤثر، شناسایی ورودی‌های استفاده شده، بررسی الگوریتم به کار رفته و مطالعه نتایج پیش‌بینی، بوده است. بیشتر سیستم‌های مورد مطالعه مقالات، سیستم‌های مشهور و پر کاربردی مانند LANL، بلوجین/ال و خوشه‌های گوگل یا سیستم‌های محاسبات دانشگاهی مانند سیستم مشبک WSU و خوشه اوپن‌کلود هستند. در سال‌های اخیر به علت محبوبیت سیستم‌های

رایانش ابری و گسترش استفاده از آن و بخصوص ذات تجاری این سیستم‌ها، توجه به مطالعه فایل‌های ثبت وقایع و پیش‌بینی خرابی در خوشه‌های کامپیوتری که زیرساخت محاسبات ابری هستند بیشتر از سیستم‌های دیگر در این حوزه می‌باشد (بخصوص خوشه‌های کامپیوتری شرکت گوگل که به صورت عمومی داده‌های ثبت وقایع خوشه‌های خود را به اشتراک می‌گذارد). با وجود این که پژوهش‌های بسیاری بر روی پیش‌بینی خرابی در سیستم‌های توزیع‌شده مقیاس بزرگ انجام گرفته است، اما هنوز محدودیت‌ها و چالش‌هایی در این حوزه وجود دارد:

۱. اندازه و در نتیجه پیچیدگی این سیستم‌ها روز به روز در حال افزایش است.

۲. در مواقع بسیاری، این سیستم‌ها بخصوص سیستم‌های ابری، مورد استفاده کاربران کم مهارت یا حتی بدون مهارت قرار می‌گیرند.

۳. تهدیدات و حملات اینترنتی به این سیستم‌ها رو به رشد است.

۴. وجود پویایی (پیکربندی‌های مکرر، تنظیم مجدد، به روزرسانی‌های نرم‌افزاری و سخت‌افزاری و غیره) در این سیستم‌ها.

۵. گسترش یافتن این سیستم‌ها به برنامه‌های کاربردی در همه زمینه‌های فعالیت‌های انسانی.

۶. کمبود در شناخت صحیح این‌گونه سیستم‌ها که پیش‌نیاز پیش‌بینی با دقت بالا می‌باشد.

۷. محدود بودن اطلاعات موجود در فایل‌های ثبت وقایع این سیستم‌ها (به دلایل امنیتی و حفظ حریم خصوصی و یا ذخیره نکردن وقایع بیشتر توسط سیستم)

برای مقابله با این چالش‌ها و رفع محدودیت‌های این حوزه در درجه اول لازم است که مدیران این سیستم‌ها با محققان این حوزه همکاری بیشتری نمایند و اطلاعات بیشتری از سیستم‌های خود را در اختیار این محققان قرار دهند. مثلاً سیستم‌های بزرگ محاسبات ابری دیگر (غیر از گوگل) مانند آمازون، مایکروسافت و غیره می‌توانند با به

Mazet J. A., Joly D. O., et al., "Drivers of emerging infectious disease events as a framework for digital detection," *Emerging infectious diseases*, vol. 21, no. 8, pp. 1285-1292, 2015.

[5] Baru, C., Bhandarkar M., Nambiar R., M. Poess, and T. Rabl, "Setting the direction for big data benchmark standards," *Technology Conference on Performance Evaluation and Benchmarking*, pp. 197-208, Istanbul, Turkey, 2012.

[6] Huang, S. Xu, J., R. Liu, and H. Liao, "A novel compression algorithm decision method for spark shuffle process," *IEEE International Conference on Big Data*, pp. 2931-2940, Boston, MA, USA, 2017.

[7] Coulouris, G. F., Dollimore J., and T. Kindberg, *Distributed systems: concepts and design*, Addison-Wesley Longman Publishing Co., 2005.

[8] Singh, S., Singh K., and Kaur H., "Design and evaluation of policy based authorization model for large scale distributed systems," *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, no. 11, pp. 49-55, 2009.

[9] Takefusa, A., Tatebe O., Matsuoka S., and Morita Y., "Performance Analysis of Scheduling and Replication Algorithms on Grid Datafarm Architecture for High-Energy Physics Applications," *High Performance Distributed Computing*, 2003. *Proceedings. 12th IEEE International Symposium on*, WA, USA, 2003.

[10] Chervenak, A., Deelman E., Kesselman C., Allcock B., I. Foster, V. Nefedova, et al., "High-performance remote access to climate simulation data: a challenge problem for data grid technologies," *Parallel Computing*, vol. 29, no. 10, pp. 1335-1356, 2003.

[11] Deelman, E., Kesselman C., and Mehta G., "Building a Virtual Data Grid for Gravitational Wave Scientists," *Proceedings 11th IEEE International Symposium on High Performance Distributed Computing*, Edinburgh, UK, 2002.

[12] Maltsev, N., Glass E., Sulakhe D., Rodriguez A., M. H. Syed, T. Bompada, et al., "PUMA2—grid-based high-throughput analysis of genomes and metabolic pathways," *Nucleic acids research*, vol. 34, no. 1, pp. D369-D372, 2006.

[13] Chen, H., Fu, Z., "Hadoop-based healthcare information system design and wireless security communication implementation," *Mobile Information Systems*, vol. 11, no. 1, 2015.

[14] Wang, F., Ercegovic V., T. Syeda-Mahmood, A. Holder, E. Shekita, D. Beymer, et al., "Large-scale multimodal mining for healthcare with mapreduce," *Proceedings of the 1st ACM International Health Informatics Symposium*, pp. 479-483, Virginia, USA, 2010.

[15] Berman, O., Ashrafi N., "Optimization models for reliability of modular software systems," *IEEE Transactions on Software Engineering*, vol. 19, no. 11, pp. 1119-1123, 1993.

[16] Kola, G., Kosar T., and M. Livny, "Faults in large distributed systems and what we can do about them," *European Conference on Parallel Processing*, pp. 442-453, Lisbon, Portugal, 2005.

[17] Javadi, B., Kondo D., Iosup A., and D. Epema, "The Failure Trace Archive: Enabling the comparison of failure measurements and models of distributed systems," *Journal of Parallel and Distributed Computing*, vol. 73, no. 8, pp. 1208-1223, 2013.

اشتراک‌گذاری داده‌های خود در شناخت این‌گونه سیستم‌ها مؤثر باشند. همچنین وجود داده‌های ثبت وقایع کامل‌تر و دانشی از زیرساخت‌های این سیستم‌ها می‌تواند فرآیندهای تحلیل، داده‌کاوی و مدل‌سازی خرابی را قدرت ببخشد که برای این لازم است که سیستم‌های توزیع شده اطلاعات وقایع بیشتری از جنبه‌های مختلف منابع و برنامه‌های اجرا شده را ثبت و اشتراک‌گذاری نمایند.

لازم است که پژوهشگران نیز به عنوان مسیرهای تحقیقاتی پیش رو توجه بیشتری به چالش‌ها و محدودیت‌های ذکر شده داشته باشند. چارچوب‌های پیشنهادی باید انعطاف بیشتری داشته باشند تا بتوانند نتایج خود را در حالات مختلف حفظ کنند. تمامی ابعاد و ویژگی‌های مرتبط با منابع، اجرای برنامه‌ها و حتی کاربران باید مورد مطالعه و تحلیل قرار گیرند تا علاوه بر ارائه بینش‌های جدید جهت کاهش خرابی، فرایند مدل‌سازی نیز کامل‌تر و منعطف‌تر باشد. همچنین محققان می‌توانند چارچوب‌های کامل‌تری ارائه دهند که علاوه بر خرابی برنامه‌ها و منابع، سایر شرایط بحرانی محتمل مانند حملات شبکه‌ای را نیز پیش‌بینی نمایند؛ یا می‌توانند چارچوب پیش‌بینی کننده را با یک چارچوب زمان‌بندی ترکیب نمایند تا یک زمان‌بند آگاه به خرابی ارائه دهند. به علاوه لزوم مطالعه روی داده‌های ثبت وقایع چارچوب‌های جدید محاسبات توزیع شده مانند اسپارک^{۳۷}، استورم^{۳۸}، فلینک^{۳۹} و غیره نیز بسیار حس می‌شود.

مراجع

[1] Birkin, M., "Big data: big data for social science research," *Ubiquity*, vol. 19, no. 10, pp. 1-7, 2018.

[2] Gubbi, J., Buyya, R., S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, 2013.

[3] Cukier, K., "Data, data everywhere: A special report on managing information," *Economist Newspaper* vol. 394, no. 8671, London: Economist Newspaper, 2010.

[4] Olson, S. H., Benedum C. M., Mekaru S. R., Preston N. D.,

37- Spark
38- Storm
39- Flink

ed States, 2010.

[۳۵] احسان شیرزاد و حمید سعادت فر؛ «تحلیل عوامل موثر بر عدم موفقیت برنامه ها در چارچوب هادوپ براساس فایل های ثبت وقایع»، بیست و سومین کنفرانس ملی سالانه انجمن کامپیوتر ایران، دانشگاه صنعتی شریف، تهران، ۱۳۹۷.

[36] Feldt R., "Do system test cases grow old?," IEEE Seventh International Conference on Software Testing, Verification and Validation, pp. 343-352, Cleveland, OH, USA, 2014.

[37] Salfner, F. and Malek M., "Using hidden semi-Markov models for effective online failure prediction," 2007 26th IEEE International Symposium on Reliable Distributed Systems (SRDS 2007), pp. 161-174, Beijing, China, 2007.

[38] Castelli V., Harper R. E., Heidelberger P., Hunter S. W., Trivedi K. S., Vaidyanathan K., et al., "Proactive management of software aging," IBM Journal of Research and Development, vol. 45, no. 2, pp. 311-332, 2001.

[39] Li L., Vaidyanathan K., and Trivedi K. S., "An approach for estimation of software aging in a web server," Proceedings International Symposium on Empirical Software Engineering, pp. 91-100, Nara, Japan, 2002.

[40] Berenji H. R., Ametha J., and Vengerov D., "Inductive learning for fault diagnosis," The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03., pp. 726-731, St Louis, MO, USA, 2003.

[41] Fu S. and Xu C.-Z., « Exploring event correlation for failure prediction in coalitions of clusters," Proceedings of the 2007 ACM/IEEE conference on Supercomputing, p. 41, New York, United States 2007.

[42] Liang Y., Zhang Y., Xiong H., and Sahoo R., "Failure prediction in ibm bluegene/l event logs," Seventh IEEE International Conference on Data Mining, pp. 583-588, Omaha, USA, 2007.

[43] Chawla N. V., Thain D., Lichtenwalter R., and Cieslak D. A., "Data mining on the grid for the grid," IEEE International Symposium on Parallel and Distributed Processing, pp. 1-5, Miami, FL, USA, 2008.

[44] Zeinalipour-Yazti D., Neocleous K., Georgiou C., and Dikaiakos M. D., "Identifying Failures in Grids through Monitoring and Ranking," Seventh IEEE International Symposium on Network Computing and Applications, pp. 291-298, Cambridge, USA, 2008.

[45] Hacker T. J., Romero F., and Carothers C. D., "An analysis of clustered failures on large supercomputing systems," Journal of Parallel and Distributed Computing, vol. 69, no. 7, pp. 652-665, 2009.

[46] Fu S. and Xu C.-Z., "Quantifying event correlations for proactive failure management in networked computing systems," Journal of Parallel and Distributed Computing, vol. 70, no. 11, pp. 1100-1109, 2010.

[47] Hoffmann G. A., Salfner F., and Malek M., Advanced failure prediction in complex software systems, publishing at Humboldt University of Berlin, 2011. Available: <https://edoc.hu-berlin.de/handle/18452/3152>.

[48] Baldoni R., Lodi G., Montanari L., Mariotta G., and Rizzuto M., "Online black-box failure prediction for mission critical dis-

[18] Moc J, Carr DA., Understanding distributed systems via execution trace data. Proceedings 9th International Workshop on Program Comprehension. IEEE, Toronto, Canada, 2001.

[19] Bishop CM., Pattern recognition and machine learning. Springer publishing, 2006.

[۲۰] انقی زاده، حبیبی و زمانی؛ «مروری بر روش های پیش بینی بقا در همبودی سرطان با استفاده از تکنیک های یادگیری ماشین»، مجله علوم رایانشی، شماره ۷، زمستان ۱۳۹۶.

[۲۱] افرهادی و جمزاد؛ «بررسی معیارهای شباهت در بازیابی تصویر مبتنی بر محتوا»، مجله علوم رایانشی، شماره ۹، تابستان ۱۳۹۷.

[۲۲] امینی خوبی و عبدالله پوری؛ «طبقه بندی ترافیک شبکه با استفاده از الگوریتم جنگل تصادفی بهبود یافته»، مجله علوم رایانشی، شماره ۵، تابستان ۱۳۹۶.

[۲۳] اخیرمند پاریزی و نورمندی پور؛ «رفع ابهام معنایی کلمات فارسی با استفاده از رویکرد نظارت شده الگوریتم های BL»، مجله علوم رایانشی، شماره ۲، پاییز ۱۳۹۵.

[۲۴] اسلیمانان قره چیق، مومن فر و وفادار؛ «روش های نوین برای شناسایی نویسنده متون با ترکیب الگوریتم های بهینه سازی توده ذرات و ماشین بردار پشتیبان»، مجله علوم رایانشی، شماره ۱۲، بهار ۱۳۹۸.

[25] Hastie, T., Tibshirani R., and Friedman J., "Unsupervised learning," The elements of statistical learning (book), Springer publishing, 2009. Available: <https://web.stanford.edu/~hastie/ElemStatLearn>

[26] Hastie T., Friedman J., and Tibshirani R., "Overview of supervised learning," The elements of statistical learning (book), Springer publishing, 2001. Available: <https://www.springer.com/gp/book/9780387216065>

[27] Kantardzic M., Data mining: concepts, models, methods, and algorithms, John Wiley & Sons, 2011.

[28] Powers D. M., "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," CiteSeerX, 2011.

[29] Iosup A. and Epema D., "Grenchmark: A framework for analyzing, testing, and comparing grids," Sixth IEEE International Symposium on Cluster Computing and the Grid, pp. 313-320, Singapore, 2006

[30] Khalili O., He J., Olschanowsky C., Snavely A., and Casanova H., "Measuring the performance and reliability of production computational grids," Proceedings of the 7th IEEE/ACM international conference on grid computing, pp. 293-300, Barcelona, Spain, 2006.

[31] Kavulya S., Tan J., Gandhi R., and Narasimhan P., "An analysis of traces from a production mapreduce cluster," Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 94-103, Melbourne, Australia, 2010.

[32] Ren K., Kwon Y., Balazinska M., and Howe B., "Hadoop's adolescence: an analysis of Hadoop usage in scientific workloads," Proceedings of the VLDB Endowment, vol. 6, no. 10, pp. 853-864, 2013.

[33] Schroeder B. and Gibson G. A., "Understanding failures in petascale computers," Journal of Physics: Conference Series, pp. 012-022, vol. 78, no. 1, 2007.

[34] Vishwanath, K. V. and Nagappan N., "Characterizing cloud computing hardware reliability," Proceedings of the 1st ACM symposium on Cloud computing, pp. 193-204, New York, Unit-

of Jobs in Big Data System,” IEEE 41st Annual Computer Software and Applications Conference, pp. 772-777, Turin, Italy, 2017.

[63]El-Sayed N., Zhu H., and Schroeder B., “Learning from failure across multiple clusters: A trace-driven approach to understanding, predicting, and mitigating job terminations,” 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 1333-1344, Atlanta, USA, 2017.

[64]Soualhia M., Khomh F., and Tahar S., “Predicting Scheduling Failures in the Cloud: A Case Study with Google Clusters and Hadoop on Amazon EMR,” IEEE 15th International Conference on Embedded Software and Systems, New York, USA, 2018.

[65]Pitakrat T., Okanović D., van Hoorn A., and Grunske L., “Hora: Architecture-aware online failure prediction,” Journal of Systems and Software, vol. 137, no. 1, pp. 669-685, 2018.

[66]Mohammed B., Awan I., Ugail H., and Younas M., “Failure prediction using machine learning in a virtualised HPC system and application,” Cluster Computing, vol. 22, no. 2, pp. 471-485, 2019.

[67]Shetty J., Sajjan R., and Shobha G., “Task Resource Usage Analysis and Failure Prediction in Cloud,” 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 342-348, Noida, India, 2019.

[68] Shirzad E. and Saadatfar H. “Job failure prediction in Hadoop based on log file analysis.” International Journal of Computers and Applications, vol. 42, no. 7, pp. 1-10, 2020.

[69]Vance A., Hadoop, a free software program, finds uses beyond search, New York Times. March 2009, Available: <https://www.nytimes.com/2009/03/17/technology/business-computing/17cloud.html>

[70]Thain D., Tannenbaum T., and Livny M., “Distributed computing in practice: the Condor experience,” Concurrency and computation: practice and experience, vol. 17, no. 2-4, pp. 323-356, 2005.

[71] Shirzad, E. and Saadatfar, H., “A study on MapReduce job failures in Hadoop.” COMPUTER MODELLING & NEW TECHNOLOGIES, vol. 23, no. 1, pp. 7-21, 2019.

[72]Dean J. and Ghemawat S., “MapReduce: simplified data processing on large clusters,” Communications of the ACM, vol. 51, no. 1, pp. 107-113, 2008.

[73]Chen X., Lu C.-D., and Pattabiraman K., “Failure analysis of jobs in compute clouds: A google cluster case study,” in IEEE 25th International Symposium on Software Reliability Engineering, pp. 167-177, Naples, Italy, 2014.

tributed systems,” 31st International Conference on Computer Safety, Reliability, and Security, pp. 185-197, Magdeburg, Germany, 2012.

[49]Fu X., Ren R., Zhan J., Zhou W., Jia Z., and Lu G., “LogMaster: mining event correlations in logs of large-scale cluster systems,” IEEE 31st Symposium on Reliable Distributed System, pp. 71-80, Irvine, CA, USA, 2012.

[50]Saadatfar H., Fadishei H., and Deldari H., “Predicting job failures in AuverGrid based on workload log analysis,” New Generation Computing, vol. 30, no. 1, pp. 73-94, 2012.

[51]Chen X., Lu C.-D., and Pattabiraman K., “Failure prediction of jobs in compute clouds: A google cluster case study,” in IEEE International Symposium on Software Reliability Engineering Workshops, pp. 341-346, Naples, Italy, 2014.

[52]Saadatfar H. and Deldari H., “A job submission manager for large-scale distributed systems based on job futurity predictor,” International Journal of Grid and Utility Computing, vol. 5, no. 1, pp. 50-59, 2014.

[53]Rosa A., Chen L. Y., and Binder W., “Predicting and mitigating jobs failures in big data clusters,” 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Shenzhen, China, pp. 221-230, 2015.

[54]Rosa A., Chen L. Y., and Binder W., “Catching failures of failures at big-data clusters: A two-level neural network approach,” IEEE 23rd International Symposium on Quality of Service, pp. 231-236, Portland, USA, 2015.

[55]El-Sayed N. and Schroeder B., “How reliable are large-scale jobs in parallel clusters?”, CSRG-627, publishing at University of Toronto, 2015. Available: <ftp://ftp.cs.toronto.edu/cs/ftp/csrg-technical-reports/627>

[56]Agrawal B., Wiktorski T., and Rong C., “Analyzing and Predicting Failure in Hadoop Clusters Using Distributed Hidden Markov Model,” International Conference on Cloud Computing and Big Data in Asia, pp. 232-246, Huangshan, China, 2015.

[57]Hongyan T., Ying L., Jia T., and Wu Z., “Hunting Killer Tasks for Cloud System through Machine Learning: A Google Cluster Case Study,” IEEE International Conference on Software Quality, Reliability and Security, pp. 1-12, Vienna, Austria, 2016.

[58]Yoo W., Sim A., and Wu K., “Machine learning based job status prediction in scientific clusters,” SAI Computing Conference, IEEE, London, UK, 2016.

[59]Rosa A., Chen L. Y., and Binder W., “Failure analysis and prediction for big-data systems,” IEEE Transactions on Services Computing, vol. 10, no. 6, pp. 984 - 998, 2017.

[60]Islam T. and Manivannan D., “Predicting Application Failure in Cloud: A Machine Learning Approach,” IEEE International Conference on Cognitive Computing, pp. 24-31, Honolulu, USA, 2017.

[61]Liu C., Han J., Shang Y., C. Liu, Cheng B., and J. Chen, “Predicting of Job Failure in Compute Cloud Based on Online Extreme Learning Machine: A Comparative Study,” IEEE Access, vol. 5, no. 1, pp. 9359-9368, 2017.

[62]Hongyan T., Ying L., Long W., Jing G., and W. Zhonghai, “Predicting Misconfiguration-Induced Unsuccessful Executions