ششمین همایش بین‌المللی افق های نویـن در
مهندسی برق، کامپیوتـر و مکانیک
6th International Conference on the New Horizons in
Electrical Engineering, Computer and Mechanical
www.mhconf.ir

مروری بر تکنیک ها و ابزارهای پردازش جریان در کلان داده

[1]رویا جعفری، [2]اصغر درویشی

[1]دانشجوی دکتری تخصصی نرم افزار دانشگاه آزاد اسلامی واحد تهران جنوب، Jafarir.ir@gmail.com

[2]استادیار و عضو هیئت علمی گروه مهندسی نرم افزار دانشگاه آزاد اسلامی واحد تهران جنوب، A_Darvishy@azad.ac.ir

چکیده

با رشد روزافزون دستگاههای دیجیتال و ارتباطات شبکه‌ای گسترده میان آنها بطور ویژه در دستگاه‌های اینترنت اشیا و همچنین شبکه‌های اجتماعی در دهه‌های اخیر، حجم بزرگی از داده در حال تولید می‌باشد که کلان‌داده نامیده می‌شود. کلان‌داده را می‌توان توالی نامحدود و پیوسته‌ای از داده در نظر گرفت که با سرعت در حال تولید هستند و به نام جریان داده می‌باشند. جریانی که اگر در زمان مناسب پردازش شود، منافع زیادی برای بسیاری در پی خواهد داشت و در غیراینصورت، داده مورد نظر ارزش خود را از دست می‌دهد. درکنار نگرانی هایی مانند چگونگی ذخیره این حجم از داده و نیز در دسترس بودن آن، پردازش به موقع آن، چالشی مهم برای بسیاری از سازمان‌ها می‌باشد. روش های پردازش داده به دو صورت دسته ای و مبتنی بر جریان می باشند که راه حلی برای پردازش حجم زیادی داده ایستا که از قبل در سیستم ذخیره شده اند می باشد بدین صورت که داده پس از جمع‌آوری از منابع مختلف تولیدکننده داده و ذخیره و نگهداری در سیستم، بصورت دسته ای مورد پردازش قرار می‌گیرد که انجام آن زمانبر است ولی در روش‌های مبتنی بر جریان، داده به محض دریافت و بصورت بلادرنگ پردازش می‌شود و همین امر موجب افزایش سرعت پردازش و در دسترس‌بودن داده را فراهم می‌نماید. پردازش جریان داده، نیاز به روش‌ها و ابزارهایی برای دریافت داده و تجزیه و تحلیل بر آن دارد که راهکارهایی برای حل چالش‌های پردازش جریان پیوسته داده می‌باشد. در مطالعه پیش‌رو، روش‌ها و تکنیک‌های کلان داده و فریمورک‌های محبوب که سرعت پردازش داده را افزایش می‌دهند مورد بررسی قرار گرفته است. رویکرد پردازش جریان برای زبان طبیعی، روش‌های توزیع شده پردازش جریان داده، موتورهای پردازش و در نهایت، یک دسته‌بندی سیستماتیک صورت گرفته درباره چالش‌ها و راهکارهای آن‌ها در حوزه پردازش جریان داده برای انتخاب فریمورک مناسب‌تر ارائه داده شده است.

واژه های کلیدی:

کلان داده، پردازش جریان، پردازش دسته ای، بلادرنگ، موتورهای پردازش

۱- مقدمه

در دهه های اخیر بویژه دهه جاری، جهان مورد تهاجم همه انواع تکنولوژی در بخش های مختلف می باشد[4]. دیجیتالی شدن هر حوزه ای از زندگی بشر منجر به انفجاری در تولید داده شده است[5] که شامل افزایش پیوسته حجم و جزئیات داده های جذب شده توسط سازمان ها از قبیل ظهور رسانه های اجتماعی، اینترنت اشیا و مالتی مدیا که منجر به ایجاد یک جریان بسیار عظیم داده در شکل های مختلفی مانند ساختاریافته، نیمه ساختاریافته و ساختارنیافته شده است[6] خلق داده با سرعت بی سابقه ای در حال ایجاد است که آن را کلان داده می نامیم. کلان داده ها به هر مجموعه بسیار وسیع و پیچیده ای اطلاق می شود که پردازش آن با استفاده از برنامه های پردازش سنتی مشکل خواهد بود. کلان داده ها مجموعه ای از تکنیک ها و فناوری هایی است که نیازمند فرم های جدیدی برای آشکار نمودن مقادیر پنهان بر روی مجموعه داده های متنوع، پیچیده با مقیاس بزرگ می باشند[1]. در واقع قابلیت استخراج اطلاعات مفید از مجموعه داده های بزرگ و یا جریان های داده بدلیل حجم، تنوع و سرعت داده ها است، امکانی که قبلا وجود نداشت[۱۳] که یکی از تکنیک های رایج استخراج داده، خوشه بندی می باشد، مجموعه ای از خوشه های محدود شده که اغلب روی تعداد خوشه ها تعریف می شود[۱۴].

کلان داده را می توان متشکل از توالی نامحدود و بزرگی از عناصر داده ای که به صورت پیوسته و با سرعت زیاد در حال تولید می باشند نامید که به آن جریان داده می گوییم. بدلیل توالی نامحدود جریان کلان داده و سرعت تولید آن، امکان ذخیره سازی تمام داده های جریان وجود ندارد. داده ها با

سرعت زیاد و به صورت بلادرنگ در حال ایجاد هستند. در بعضی کاربردها نیاز آن است که به محض وارد شدن داده به درخواست کاربر پاسخ داده شود و ممکن است نتوانیم به اندازه کافی صبر کنیم تا یک گزارش در سیستم برای مدت طولانی پردازش شود[2].

تا سال ۲۰۱۹، ۲۶ میلیارد از دستگاههای اینترنت اشیا بهم متصل بودند و انتظار می رود که این رقم تا ۲۰۲۵ به ۸۰ میلیارد دستگاه در جهان برسد. سیستم های پردازش جریان به سرعت در حال تبدیل شدن به عنصر اصلی پشته پردازش داده در بسیاری از شرکت ها می شوند[۱۵] این در حالی است که الگوریتم های سنتی پردازش جریان بدلیل استفاده بسیار زیاد از منابع (مانند حافظه) برای پردازش پویا و توزیع شده داده قادر به انجام فعالیت در مقابل این جریان داده وسیع نیستند. عناوینی مانند: تکامل جریان داده، زمان اجرا، مصرف حافظه، چند بعدی بودن وسیع، مفاهیم جهش، تاخیر و عدم تعادل از جمله چالش های الگوریتم های سنتی در مواجه با نوع جدید جریان داده می باشند[4]. همانگونه که تکنولوژی های پردازش جریان در مسیر بلوغ قدم می گذارند، سازمان های بیشتری جذب این بخش شده و بر روی تحول دیجیتال سرمایه گذاری می کنند. مجموعه داده های جهانی و بدون مرز مانند وب لاگ ها، آمار استفاده از تلفن های همراه و شبکه های حسگر روز به روز در حال افزایش است و برنامه های جدید تجزیه و تحلیل جریان در طیف گسترده ای از صنایع، شناسایی و پیاده سازی می شوند. جریان داده بسیاری از پلت فرم های شبکه های اجتماعی مانند: فیس بوک، توئیتر و لینکدین شاهرگ زندگی آنها می باشد و لذا استخراج معنی دار و آگاهی به موقع از داده های نامحدود، چالش پیش رو در بیشتر بخش ها می باشد[7]. پردازش جریان داده به عنوان کلیدی در حوزه پردازش جریان اطلاعات است و هدف آن درک رفتار داده و استخراج اطلاعات برای تصمیم گیری بهتر می باشد[8]. در ابتدا پردازش داده ها بصورت دسته ای بود که بعد از جمع آوری داده ها از منابع گوناگون و انجام عملیات استخراج، انتقال و بارگذاری[1] در مخزنی از داده، با توجه به نوع کسب و کار و نیازهای سازمان شروع به ساخت و ایجاد گزارش های داشبوردی برای سطوح مختلف مدیران صورت می پذیرفت که به این مخزن انباره داده می گویند که شامل پردازش داده های ایستا می باشد. در حال حاضر، جریان داده بصورت پویا و پیوسته است که نیاز به پردازش لحظه ای را ایجاد می کند. پردازش هایی با کمترین تاخیر، معنی دار و مقیاس پذیر که کمبودهای سیستم های پردازش دسته ای را جبران کنند و نیاز کاربران را در لحظه، پاسخگو باشند، به اصطلاح بلادرنگ و در زمان باشند. از این دست پردازش های بلادرنگ می توان به سناریوهایی مانند کشف تقلب در تراکنش های مالی و یا تجزیه و تحلیل سامانه های مراقبت سلامت و بهداشتی که شامل حسگرهای دیجیتال و اینترنت اشیا هستند نام برد. به دلیل آنکه انواع خاصی از جریان های داده مانند ارزش سهام، تراکنش های کارت اعتباری،ترافیک داده های حساس به زمان بیمار و روند اخبار در صورتی که فوری پردازش نشوند، ارزش خود را از دست خواهند داد لذا توانایی مدیریت و پردازش جریان های مداوم و پیوسته داده ها به یکی از اصلی ترین بخش ها در سازمان های داده محور امروزی تبدیل شده است[7].

آنچه پیش روی این بررسی می باشد بدین صورت خواهد بود: در بخش ۲ مروری بر تکنیک های کلان داده خواهیم داشت. رویکرد جریان برای پردازش زبان طبیعی را در بخش ۳ بیان می کنیم. در بخش ۴، مروری بر پردازشگرهای بلادرنگ خواهیم داشت. فریمورک های توزیع شده پردازش جریان داده در بخش ۵ بیان شده اند و دسته بندی سیستماتیک برای انتخاب بهتر ابزارهای پردازش جریان داده را در بخش ۶ ارائه نموده ایم و در نهایت، در بخش ۷ به بیان خلاصه ای از آنچه در این مطالعه مورد بحث و بررسی قرار گرفت، می پردازیم.

۲- مروری بر تکنیک های کلان داده

در [6] تمرکز بر مرور و بررسی تکنیک هایی است که برای پردازش و به کارگیری، نمایش اطلاعات و تحلیل آنها بکار گرفته شده است. این تکنیک ها مجموعه ای از قواعد را در برمیگیرد که شامل داده های آماری، یادگیری ماشین، شبکه های عصبی، تحلیل شبکه های اجتماعی، پردازش سیگنال، تشخیص الگو، روش های بهینه سازی و روش های مجازی سازی می باشد که در ادامه به آنها اشاره می شود:

روش های بهینه سازی: روش هایی که برای برطرف نمودن و حل مسائل تا حدودی در زمینه هایی مانند فیزیک، زیست، مهندسی و اقتصاد بکار گرفته شده است. شبیه سازی تبرید، الگوریتم ژنتیک و حرارت دهی کوانتومی.

روش های آماری: تکنیک های آماری برای بدست آوردن همبستگی ها و روابط نسبی بین اهداف گوناگون بکار می رود. الگوریتم تقریبی برای رگرسیون یکنواخت چند متغیره در مقیاس بالا و الگوریتم های آماری موازی.

داده کاوی: تحلیل خوشه بندی، دسته بندی، رگرسیون و یادگیری قانون.

یادگیری ماشین: کشف دانش و تصمیم گیری هوشمندانه خودکار از شاخصه های این بخش می باشد. SVMموازی و شبکه های عصبی مصنوعی.

روش تجسم: این روش تکنیک های مورد استفاده برای جداول، تصاویر، نمودارها و در کل، نمایش بصری برای درک داده می باشد.

تحلیل شبکه های اجتماعی: یک تکنیک کلیدی است که در جامعه شناسی مدرن ظهور پیدا کرده است و به روابط اجتماعی بر حسب نظریه تئوری شبکه می نگرد که شامل گره ها و روابط بین آن ها است.

[1] ETL = Extract, Transfer, Load

۳- رویکرد جریان برای پردازش زبان طبیعی

در [10]، یک معماری جدید توزیعی و تکنولوژی جدید برای مقیاس پذیری تحلیل متن با حفظ سرعت نرخ رشد جریان های داده ارائه شده است و زنجیره ای کامل از پردازشگرهای متنی بر روی چندین ماشین مجازی می باشد. همچنین ماژول های پردازش زبان طبیعی که شامل ماشین های مجازی می باشند و بیشتر آنها از ابزارهای خط لوله IXA وارد می شوند، پیشنهاد شده است. جدول۱ ماژول های نصب شده در ماشین مجازی زبان انگلیسی را نشان می دهد. داخل ماشین های مجازی با استفاده از فریمورک Storm برای هر محاسبه جریان مدیریت می شوند و هر ماژول پردازش زبان از گره های bolt داخل توپولوژی Storm تشکیل شده است. هنگامی که تاپل جدیدی وارد می شود گره های bolt یک دستور خارجی را برای ارسال محتوای تاپل به یک جریان ورودی استاندارد فرا می خواند. خروجی ماژول پردازش جریان از خروجی جریان استاندارد دریافت شده به گره بعدی در توپولوژی ارسال می شود. بنابراین هر ماژوی یک سند NAF[۲] با سند و افزودن خط لوله برای هر ابزار جدید برای نوشتن و خواندن NAF را دریافت می کند. در نتیجه برای دانش ما جامع ترین خط لوله NLP موجود است. این روش می تواند برای تحلیل متون فارسی مورد بهره برداری قرار گیرد. از دو مجموعه داده اطلاعات خبری راجع به Car و Wikinews برای آزمایش ماژول ها استفاده شده است.

جدول ۱. ماژول های پردازش زبان نصب شده بر روی ماشین های مجازی
ستون آخر نشان دهنده نسخه خط لوله ای است که ماژول ها استفاده می کنند

ماژول	توضیحات	خط لوله
ixa-pipe-tak	Tokenizer, sentences splitter	(۱و۲)
ixa-pipe-pos	POS tagger	(۱و۲)
ixa-pipe-parse	Constituency parser	(۲)
TimePro	Time expression recognition	(۱و۲)
ixa-pipe-nerc	Named Entity Recognition	(۱و۲)
WSD	Word Sense Disambiguation	(۱و۲)
dbpedia-spotlight	Named Entity Disambiguation	(۱و۲)
ixa-pipe-coref	Coreference resolution	(۲)
MATE	Dependency parser and Semantic Role Labeling	(۱و۲)
opinion miner	Opinion detection and Opinion holders to targets	(۲)
Factuality	Factuality	(۱و۲)
eCoref	Event coreference	(۱و۲)

در مجموعه داده Car ماژول MATE-SRL تقریبا ۸۰درصد زمان پردازش را به خود اختصاص می دهد. در مجموعه داده Wikinews ماژول MATE-SRL ۵۶ درصد از کل زمان را می گیرد و در ادامه، ixa-pipe-coref ۲۱درصد از زمان را مصرف می کند. این نتایج پیشنهاد می دهند که هر دو مازول کاندیدهای مناسبی برای موازی سازی هستند. برای موازی سازی ماژول ها از یک خط لوله NLPکه شامل ۴ ماژول: ixa-pipe-tok، ixa-pipe-pos،ixa-pipe-nerc و UKB و ابزاری برای ابهام زدایی از حس کلمه بر اساس گراف است، استفاده شده است.

جدول ۳. عملکرد خط لوله پردازش زبان طبیعی با تنظیمات متفاوت
Storm دارای یک نمونه و $Storm_{۲}$ دارای دو نمونه و $Storm_{۴}$ دارای چهار نمونه از ماژول WSD است

	زمان کل	کلمات(ثانیه)	فرستاده شده(ثانیه)	بهره (٪)
۱۰۰ documents				
Pipeline	۲۱m ۱۶s	۱۰۸,۸	٤,۲	-
Storm	۱۸m ٤۳s	۱۲۳,٥	٤,۸	۱۲,۰
$Storm_{۲}$	۱۰m ٤۸s	۲۱٤,۳	۸,٤	٤۹,۳
$Storm_{٤}$	۷m ٤٦s	۲۹۷,٦	۱۱,٦	٦۳,٥
$Storm_{٥}$	۷m ٤٤s	۲۹۹,۱	۱۱,۷	٦۳,۷
$Storm_{٦}$	۷m٤۸s	۲۹٦,۱	۱۱,٦	٦۳,۳
۱۰۰۰ documents				
Pipeline	۳h ۱٥m ۱٦s	۱۰۱,۲	٤,۲	-
Storm	۲h ٥۰m ۲۱s	۱۱٦,۰	٤,۸	۱۲,۸
$Storm_{۲}$	۱h ٤۰m ۳۷s	۱۹٦,٥	۸,۱	٤۸,٥
$Storm_{٤}$	۱h ۱٤m ۲٥s	۲٦٥,٦	۱۰,۹	٦۱,۹
$Storm_{٥}$	۱h ۱۰m ٤٥s	۲۷۹,۳	۱۱,٥	٦۳,۸
$Storm_{٦}$	۱h ۱۱m ۳۷s	۲۷٦,۰	۱۱,۳	٦۳,۳

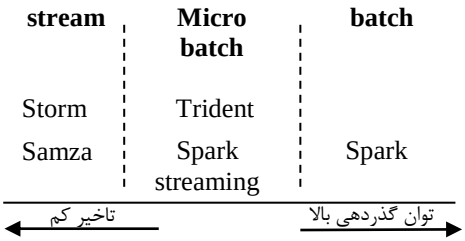
جدول ۳ نشان دهنده اجرای خط لوله NLP در تنظیمات مختلف است: ۱۰۰ سند و ۱۰۰۰ سند. با اجرای چندین نمونه از UKB WSD عملکرد کلی افزایش معناداری پیدا می کند. بالاترین بهره برای WSDدر ۵ نمونه است که برابر 63.7 درصد است و با افزایش نمونه ها، بهره افزایش نمی کند. نتایج نشان دهنده فضای بزرگی برای بهبود پردازش NLP است.

---

[۲] فرمت حاشیه نویسی NLP

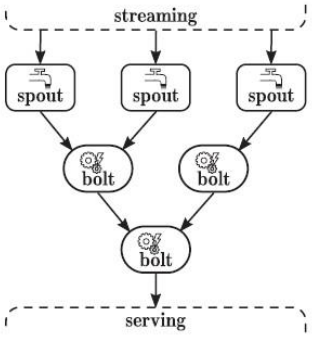
۴- پردازشگرهای بلادرنگ

در [11] اشاره می شود که پردازشگرهای جریان با وجود مشترک بودن مفاهیم و اصل کار خود تفاوت اساسی بین سیستم هایی که به سرعت قابل دستیابی پردازش تبدیل می شوند مانند تاخیر وجود دارد که در شکل ۲ نشان داده شده است. پردازش سریع داده ها به محض ورود، منجر به کاهش تاخیر خواهد شد در حالیکه بافر کردن و پردازش دسته ای داده ها با وجود آنکه کارایی را افزایش می دهد اما زمان سپری شده داده در خط لوله را افزایش می دهد. سیستم های جریان محور مانند Storm و Samza تاخیر بسیار کم و هزینه نسبتا بالا را ایجاد می کنند در حالیکه سیستم های دسته‌ای به منابع کارای غیرموازی دست پیدا می کنند که هزینه تاخیر آنها بصورت غیرقابل اجتنابی برای برنامه های بلادرنگ، بالا می باشد. فاصله میان تاخیر کم و گذردهی بالا بسیار زیاد است و بعضی از سیستم‌ها مانند Storm Trident و Spark Streaming استراتژی های دسته‌های کوچک[3] دارند که برای معاوضه مابین تاخیر و گذردهی بکار می روند: در Storm Trident، تاپل ها به هدف ماندن در یک مدل پردازش زمانی به سود افزایش گذردهی، به دسته‌هایی تبدیل می‌شوند، درحالیکه در Spark Streaming اندازه دسته‌های داده با هدف ایجاد تاخیر کم، محدود می‌باشد.



شکل ۲. انتخاب یک مدل پردازش، موازنه ای بین گذردهی و تاخیر

Storm: اولین سیستم پردازش جریان توزیع شده که معروف به هادوپ بلادرنگ[4] می باشد. یک خط لوله داده در Storm، توپولوژی نامیده می شود که همانطور که در شکل ۳ مشخص می باشد گراف جهت دار جریان داده می باشد و لبه های جهت دار میان گره ها مراحل پردازش را نشان می دهند: گره‌های spout، گره‌هایی هستند که داده از آنجا وارد می‌شود و تاپل های ورودی را به گره‌های جریان‌های پایین‌تر که bolt نامیده می‌شوند و عملیات پردازش روی تاپل‌های ورودی را انجام می‌دهند، منتشر می‌کنند. Storm ضمانتی برای ترتیب پردازش تاپل‌ها نمی‌دهد و همچنین حداقل یکبار پردازش را از طریق ردیابی تاپل‌های وارد شده به توپولوژی فراهم می‌کند. یک مکانیسم فشار برگشتی[5] برای داده وارد شده معرفی نمود که داده‌هایی که سرعت ورود آنها بیشتر از آن است که Storm بتواند آنها را پردازش کند، داده مجدد وارد گره‌های توپولوژی به هدف پردازش گردد. بدون چنین مکانیسمی، پردازش می تواند تبدیل به گلوگاهی شود که موجب کاهش توان گذردهی خواهد شد.



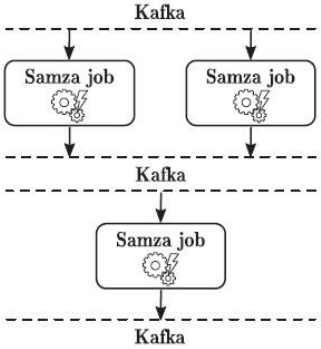
شکل۳. جریان داده در توپولوژی Storm: داده از لایه جریان وارد و پس از عبور از مولفه های Storm درخروجی به لایه خدمت می رسد

Storm Trident: توپولوژی ها، گراف های جهت دار بدون چرخه[6]می باشند و از چرخه پشتیبانی نمی کنند که مناسب پیاده سازی الگوریتم های تکرارپذیر نمی باشد. Trident با تاپل های مجزا کار نمی کند بلکه بصورت دسته های کوچک با معرفی اندازه دسته ها به عنوان پارامتری برای افزایش گذردهی و کاهش تاخیر می باشد (تاخیرهایی در حد چند میلی ثانیه). همه دسته ها با ترتیب و بصورت

[3] Micro-batching
[4] Hadoop of real-time
[5] Backpressure mechanism
[6] DAG = Directed Acyclic Graph

متوالی، یکی بعد از دیگری، پردازش می شوند. همینطور امکان پردازش چندین دسته بصورت موازی نیز وجود دارد. در بالای Storm، Trident برنامه ای برای مدیریت خطاپذیری با دقیقا یکبار پردازش معنا فراهم می کند.

Samza: یک پردازشگر جریان با مدل پردازش در زمان و حداقل یکبار پردازش معنا می باشد. جریان ها تقسیم می شوند و پیام ها(مانند آیتم های داده) در هر قسمتی، مرتب می شوند در حالیکه هیچ ترتیبی بین پیام ها در قسمت های مختلف وجود ندارد. برخلاف Strom، Samza فقط به ZooKeeper وابسته نیست بلکه در بالای Hadoop YARN برای خطاپذیری اجرا می شود.



شکل۴. جریان داده در یک خط لوله تجزیه و تحلیل معمولی Samza

همانطور که در شکل ۴ مشخص می باشد، یک کار Samza مراحل پردازش را در خط لوله تجزیه و تحلیل نشان می دهد که بطور تقریبی با یک گره bolt در توپولوژی Storm مطابقت دارد. خروجی های تولید شده در هر کار Samza همیشه در کافکا نوشته می شود زیرا ممکن است مورد نیاز یک کار Samza دیگر باشد که ورودی خود را از کافکا دریافت می کند. این طراحی مراحل پردازش را جدا می کند و در نتیجه توسعه به سادگی صورت می پذیرد. از دیگر مزایا این پردازشگر جریان، بافر کردن داده بین گام های پردازش است که می تواند توسط تیم های دیگر همان سازمان مورد استفاده قرار گیرد و در نهایت، حذف نیاز به الگوریتم فشار پشتی را منجر خواهد شد زیرا مشکلی در پرشدن موقتی یک کار خاص بدلیل اندازه منطقی توسعه با کافکا، وجود ندارد. چون Samza پیام ها را به ترتیب پردازش می کند و نتایج را بصورت ماندگار نگهداری می نماید از گم شدن داده جلوگیری می نماید.

Spark Streaming: یک فریمورک پردازش جریان که به عنوان جانشین غیررسمی هادوپ ذکر شده است چون در مقایسه، مزایای متعددی را ارائه می دهد که مهمترین آن، بهبود عملکرد قابل توجه از طریق ذخیره سازی در حافظه می باشد. به ویژه الگوریتم های تکراری(الگوریتم های یادگیری ماشین مانند k-means یا رگرسیون لجستیک) به ترتیب اندازه، شتاب گرفته‌اند زیرا لزومی به نوشتن و بارگذاری داده از دیسک در هر مرحله از پردازش نیست. باتوجه به مزایای این عملکرد، Spark الگوریتم های متنوع یادگیری ماشین را از طریق کتابخانه های MLlib فراهم می نماید. مانند Storm و Samza خطاپذیر عمل می کند و از مقیاس پذیری منابع اختصاص داده شده بصورت پویا پشتیبانی می کند. برای جلوگیری از دست دادن داده حتی برای منابع داده ای غیر قابل اطمینان، Spark Streaming اجازه پیش نویس داده[۷] از جایی که داده می تواند مجدد بعد از خطا اجرا شود را می دهد. جدول ۴ مقایسه ای بین فریمورک های اشاره شده را بیان می کند.

جدول۴. مقایسه Apache Storm/Trident، LinkedIn`s Samza و Apache Spark Streaming

	Storm	Trident	Samza	Spark Streaming
Strictest guarantee	At-least-once	Exactly-once	At-least-once	Exactly-once
Achievable latency	<< ۱۰۰ ms	< ۱۰۰ms	< ۱۰۰ms	< ۱s
State management	Yes	Yes (small state)	Yes	Yes
Processing model	One-at-a-time	Micro-batch	One-at-a-time	Micro-batch
Backpressure mechanism	Yes	Yes	Not required(buffering)	Yes
Ordering guarantees	No	Between batches	Within stream partitions	Between batches
Elasticity	Yes	Yes	No	Yes

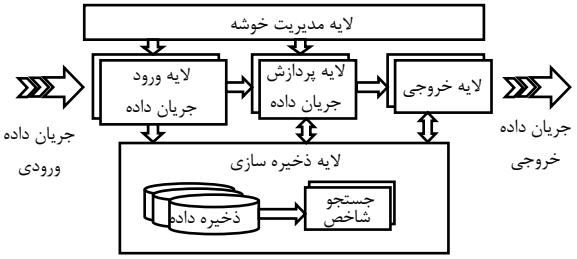
[7] WAL = Write-Ahead Log

۵- فریمورک های توزیع شده پردازش جریان داده

در [7]، ۲ محور اصلی برای پردازش جریان ذکر شده است: سیستم های مدرن پردازش جریان[8] و موتورهای پردازش جریان داده[9] که به توضیح آن در ادامه خواهیم پرداخت.

۵-۱- سیستم های پردازش جریان داده

این سیستم ها سعی در ترکیب قابلیت های پردازش دسته ای و جریان در یک یا چند خط لوله پردازش موازی داده را دارند. رویکرد سنتی پردازش داده، پردازش دسته ای می باشد، مثلا داده های تراکنش های مالی که در یک انباره داده تجمیع و ذخیره می گردد و سپس، داده های ذخیره شده مورد پردازش و تحلیل قرار می گیرند و یا سیستم جستجوی وب که با جستجوی مداوم در وب به دنبال استخراج دانش می باشد و پردازش آن بصورت دسته ای است تا شاخص های جستجو را در هر ساعت بروزرسانی نماید. مشکل هر دو سناریو، تاخیر زیاد بین جمع آوری داده و استفاده از آن ها را نشان می دهد. هدف این سیستم ها غلبه بر تاخیر با پردازش بلافاصله داده ها در زمان رسیدن با در نظر گرفتن مقیاس پذیری و خطاپذیری است. ورودی DSPS ها، جریان های پیوسته داده هستند و هر DSPS علارغم تفاوت در منطق کسب و کار و دامنه تحت پوشش، دارای یک خط لوله پردازش داده عمومی می باشد که از ۵ لایه شامل: لایه دریافت داده، لایه پردازش جریان داده، لایه ذخیره سازی، لایه مدیریت منبع و لایه خروجی تشکیل شده است که در شکل ۵ قابل مشاهده می باشد.



شکل۵. معماری سیستم پردازش جریان داده

لایه دریافت داده: استفاده از سرویس های پیام رسان از مدل های سنتی مانند MQTT، RabbitMQ و ActiveMQ تا محصولات جدیدتر مانند ZeroMQ و NSQ می باشد.

لایه پردازش جریان داده: محلی که برنامه های پردازش جریان یا کارها اجرا می شوند. موتورهای پیشرفته مدیریت پردازش[10] شامل Samza SQl، KSQL و SQL stream Blaze می باشند.

لایه ذخیره سازی: راه حل های ذخیره داده برای پشتیبانی از معماری DSPS از سیستم فایل های سنتی مانند HDFS و BFS[11] تا دیتابیس های رابطه ای فایل توزیع شده مانند PostgreSQL، دیتابیس های درون حافظه ای مانند VoltDB، مبتنی بر سند مانند MongoDB، سیستم های ذخیره مبتنی بر گراف مانند Neo4j، دیتابیس های NoSQL مانند Cassandra و NewSQl مانند CockroachDB می باشند.

لایه مدیریت منبع: فرایندهای موجود در یک سیستم توزیع شده می تواند از طریق شبکه یا برخی از فضاهای ذخیره سازی مشترک خواندن/نوشتن، پیام را مستقیما تبادل نماید. مدیریت اعضای گروه و مدیریت فراداده هنگام ساختن یک خوشه توزیع شده از بخش های مهم این لایه می باشد. مدیریت منابع پیشرفته مانند ZooKeeper، Mesos، Tupperware و سرویس مدیریت شده ای مانند Borg می باشند.

لایه خروجی: نتایج حاصل شده از خط لوله پردازش جریان می تواند به برنامه های دیگری هدایت شوند یا وارد ابزار تجسم سازی گردند و یا یک داشبورد نظارتی/هشدار را ایجاد نمایند (شکل ۵).

ابزارهای تجسم سازی به چهار دسته زیر تقسیم می شوند:

- تصویرسازی گراف با استفاده از ابزاری مانند TreeNetViz.
- تصویرسازی متنی که شامل تصویرسازی متن ایستا با ابزاری مانند WordClouds یا PhraseNet و یا متن پویا با ابزاری مانند TextFlow یا SparkClouds.
- تصویرسازی نقشه برای داده های جغرافیایی با ابزاری مانند BirdVis
- تصویرسازی چندمتغیره داده برای انواع داده های عمومی با استفاده از ابزارهای مقیاس پذیری چند بعدی.

---

[8] DSPS = Data Stream Processing System
[9] DSPE = Data Stream Processing Engine
[10] DSME = Data Stream Management Engine
[11] Baido File System

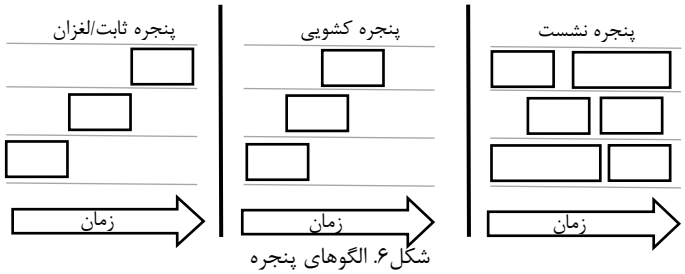
۵-۲- موتور پردازش جریان داده

موتورهای پردازش جریان شامل خطوط پردازش جریان داده را به عنوان یک DAG[۱۲] منطقی متصل شده به کارهای پردازش جریان می باشد که DAG را اجرا می کند بدین صورت که آیتم های جریان را از یک منبع دریافت می کند، روی آن پردازش انجام می دهد و نتایج را بر می گرداند و از چندین مولفه تشکیل شده است که شامل منابع داده و سینک، درایور برنامه، مدیر جریان و پردازشگرهای جریان است. موتورهای پردازش جریان توزیع شده با تمرکز بر مقیاس پذیری حجم کلان داده بصورت پیوسته طراحی شده اند[۱۸].

جدول۵. مقایسه موتورهای پردازش جریان داده

معیار	Storm	Spark Streaming	Flink	Kafka Streams	IBM Streams
Version	2.0	2.4.3	1.8	2.3.0	4.3.0
Model	Native	Micro batch, continuous processing	Native	Native	Native
Partitioning	Shuffle, field, partial key, all, global, none, direct, local, custom	Hash, range	Same, hash, random	Same, hash, round-robin	Same, hash, random, round-robin, entire, modulus, range, DB2, auto
State management	Key value, redis	Write-ahead and other state stores	Key-value store	Key-value, hash map	Tuple history, operator windows, operator custom logic, primitive operators
Message delivery	Atleast-once, Exactly-once through Trident	Exactly- and At-least-once	Exactly-once	Exactly-once	Exactly-once
Fault tolerance	Checkpoint, Stream replay	Checkpoint	Stream replay, checkpoint	Stream replay	Checkpoint, consistent region, replay

داشتن یک دید کلی از جریان داده های ورودی، امکان پذیر نیست لذا داده ها بصورت قطعه هایی با پنجره های زمانی محدود در بالای یک جریان داده بی انتها، پردازش می شوند. پنجره های مبتنی بر زمان به سه گروه تقسیم می شوند(شکل ۶): پنجره های ثابت: جریان داده ورودی را به سگمنت هایی با اندازه ثابت که هر کدام یک طول پنجره، زمان شروع و زمان پایان دارند، مشخص می شود. پنجره های کشویی: جریان داده ورودی را به سگمنت هایی با اندازه ثابت تقسیم می کند که هر کدام یک طول پنجره، زمان شروع و زمان پایان دارند. اگر بازه کشویی به اندازه طول پنجره باشد، تبدیل به پنجره ثابت می شود در نتیجه یک یا چند بخش از داده شامل بیش از یک پنجره کشویی می گردد. پنجره های نشست: بر خلاف دو پنجره دیگر، طول از پیش تعریف شده ای ندارد و معمولا با فاصله عدم تحرک که بزرگتر از حد آستانه است تعریف می شود.



شکل۶. الگوهای پنجره

این موتورها به چهار دسته زیر تقسیم می شوند:

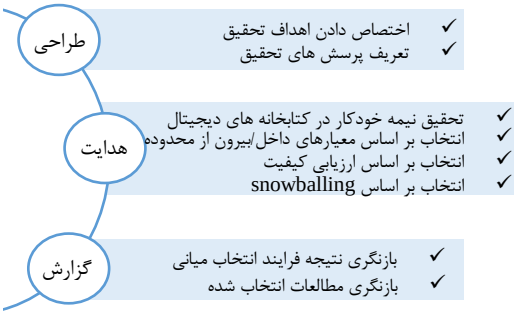
۱- سیستم های بر اساس پرس و جو مانند NiagaraCQ و AsterixDB
۲- سیستم های یادگیری ماشین توزیع شده آنلاین مانند SAMOA[۱۳]
۳- سیستم های تجزیه و تحلیل گراف جریان مانند GraphJat
۴- فریمورک های همه منظوره پردازش جریان داده مانند Flink و Spark Streaming

۶- چالش ها و راهکارها برای پردازش بلادرنگ جریان کلان داده

در [6]، به بررسی سیستماتیک ادبیات پردازش جریان که به گردآوری مطالعات مرتبط با پردازش جریان بلادرنگ و چالش ها و تحولات انجام شده است، می پردازد. از میان ۶۷۹،۵۴۷ مقاله انتخاب شده برای استخراج داده، ۷۴ مقاله بعد از ارزیابی کیفی برای تحقیق باقی ماندند که در این بررسی

[12] Directed Acyclic Graph
[13] Scalable Advanced Massive Online Analysis

به پیاده سازی چالش ها، همراه با رویکردهای توسعه ای انباره داده های بلادرنگ و سیستم های پردازش جریان کلان داده و مقایسه میان آن ها را فراهم نموده است. در متدولوژی تحقیق شامل سه فاز: برنامه، هدایت و بررسی گزارش می باشد. شکل ۷ متدولوژی تحقیق و فرایندهای جستجو برای فعالیت های تحقیق را نشان می دهد.



شکل۷. متدولوژی تحقیق

یک فرایند کاملا ساختیافته که شامل موارد زیر می باشند، دنبال شده است: اهداف تحقیق، مشخص نمودن سوالات تحقیق، سازماندهی جستجوهای پایگاه داده، مطالعه انتخاب ها، غربالگری مطالعات مرتبط، استخراج داده، سنتز نتایج و نهایی نمودن گزارشات مورد بررسی.

۶-۱- اهداف تحقیق

شامل : - توسعه دادن عناوین مرتبط کتابخانه به پردازش جریان بلادرنگ در طول فاز ETL یا سایر فازها و ایجاد یک مجموعه داده.
- مشخص نمودن کارهای مرتبط که به چالش های پردازش جریان بلادرنگ، ETL و انباره داده بلادرنگ جهت می دهد.
- تشخیص شکاف های تحقیق.
- مشخص کردن رویکردها و راهکارهای موجود برای این چالش ها در زمان پیاده سازی پردازش جریان بلادرنگ برای داده های ناهمگون، ساختاریافته/غیرساختاریافته و روشن نمودن شباهت ها و تفاوت های میان آنها.

آنچه اهمیت دارد آن است که بتوان سوالات تحقیق را فرموله نمود، پرسش هایی که در جدول ۲ قرار گرفته است.

جدول۶. پرسش های تحقیق

پرسشهای تحقیق	عبارات پرسش های تحقیق	انگیزه
RQ 1	کدامیک انتشارات مرتبط برای تحقیق پردازش جریان بلادرنگ می باشند؟	مشخص نمودن آنکه تحقیق پردازش جریان بلادرنگ می تواند هدف مناسبی برای مطالعات آینده باشد با ارزیابی کیفیت کانال های هدف بر اساس رتبه بندی آن ها و دسته بندی انتشارات برای دوره زمانی انتخاب شده، نوع و مناطق جغرافیایی تحقیقات.
RQ 1.1	چه تعداد عنوان در سال های بین ۲۰۱۱ تا ۲۰۲۰ منتشر شده است؟	
RQ 1.2	کدام کانال ها و مناطق جغرافیایی تحقیقات پردازش جریان بلادرنگ را هدف قرار می دهند؟	
RQ 1.3	چه نوع تحقیقاتی، رویکردهایی و برنامه هایی برای مطالعه انتخاب می شوند؟	
RQ 1.4	چه تعداد از رویکردهای مطالعات بصورت تجربی تایید شده است؟	
RQ 2	با کدام چالش ها در طول پیاده سازی پردازش جریان داده ها مواجه شده اند؟	با ارزیابی و ساده سازی پردازش جریان بلادرنگ با نیازمندی های متفاوت و به منظور تشویق نمودن برای جنبه های مختلف با هدف تحقیقات در آینده. شناسایی برنامه هایی که پردازش جریان بلادرنگ انجام می دهند و به چیزی بجز انباره داده نیاز دارند
RQ 2.1	کدام ابزارها/تکنولوژی ها/ رویکردها برای حل این چالش ها توسعه داده شده است؟	
RQ 2.2	کدام پیاده سازی در حوزه اینترنت اشیا و رسانه های اجتماعی با پردازش بلادرنگ جریان داده گزارش شده است؟	
RQ 3	در زمان پردازش جریان بلادرنگ برای انباره داده بلادرنگ، چه رویکردها/ابزارهایی برای حل چالش ها در مرحله ETL گزارش می شود؟	برای تشخیص رویکردهای متصل گزارش شده داده جریان/نیمه جریان، ساختاریافته/نیمه ساختاریافته موجود در ادبیات پردازش جریان بلادرنگ، چالش هایی در پیاده سازی انباره داده بلادرنگ شناسایی می شود.
RQ 3.1	کدام شکل و ساختار داده برای پیاده سازی رویکردها استفاده می شود؟	
RQ 4	در حل چالش های پردازش جریان بلادرنگ چه اتفاقاتی گزارش می شوند؟	ارزیابی اتفاقات فراهم شده بوسیله این مطالعات، منجر به رویکردهای دقیقتری می شود و نیز تشخیص مجموعه داده های استفاده شده در هنگام آزمایش رویکرد پیشنهاد شده(مصنوعی یا واقعی). تشخیص اثر رویکردها بر روی عملکرد و هزینه و ارزیابی فرکانس تطبیق معیار عملکردی خاص برای اعتبارسنجی.
RQ 4.1	چه مجموعه داده هایی برای آزمایش بوسیله این رویکردها گزارش شده است؟	
RQ 4.2	چه معیار عملکردی برای آزمایش این رویکردها با مجموعه داده های واقعی سازگار می باشد؟	

فرایند هدایت نمودن این بررسی شامل چهار مرحله می باشد که به این ترتیب می باشند:

۱- مطالعات اصلی مرتبط از رایج ترین کتابخانه های دیجیتال جستجو شده اند.

۲- انتخاب مطالعات بر اساس معیارهای استخراج/شامل بودن انجام شده است.

۳- سپس، معیار ارزیابی کیفیت برای بدست آوردن کیفیت بیشتر صورت می پذیرد.

۴- جستجوهای مرتبط برای استخراج مقالات کاندید در طول این گام انجام می شود.

۶-۲- تحقیق نیمه خودکار

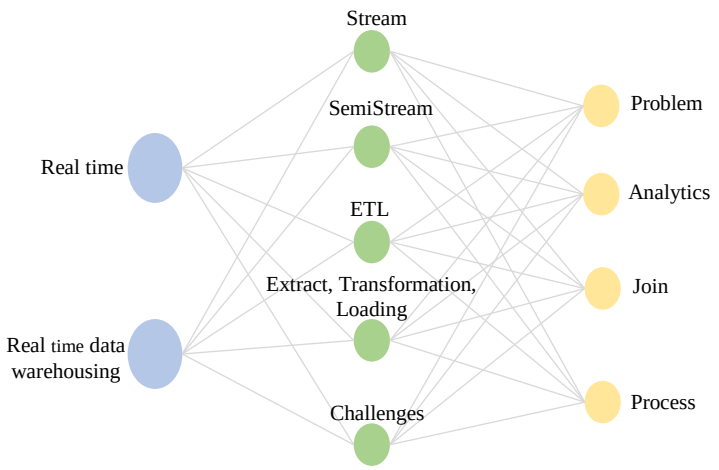
تحقیق نیمه خودکار در ۷ کتابخانه دیجیتال زیر انجام پذیرفته که در جدول ۷ آمده است.

جدول۷. کتابخانه های دیجیتال مورد استفاده

کتابخانه های دیجیتال
ACM Digital Library [http://dl.acm.org]
IEEE eXplore [http://ieeexplore.ieee.org]
ScienceDirect [https://www.sciencedirect.com]
IGI Global [https://www.igi-global.com/search/]
Inderscience Online [https://www.inderscienceonline.com/]
Hindawi [https://www.hindawi.com/]
MDPI (Multidisciplinary Digital Publishing Institute) [https://www.mdpi.com/]
arXiv [https://arxiv.org/search/cs]
Taylor&Francis Online [https://www.tandfonline.com/]

علاوه بر موارد بیان شده در جدول ۷، چند کتابخانه دیجیتال دیگر نیز که شامل محدودیت های دسترسی بودند،بررسی شدند. مرتب سازی های ممکن از جمله تحقیق را می توان در شکل 8 مورد توجه قرار داد. کلیدهای اصلی به عنوان کلید شناسایی برای تحقیق پردازش جریان بلادرنگ انتخاب شدند و کلیدهای اصلی انتخاب شده با کلیدهای ثانویه یا کایدهای تجمیعی، مرتبط هستند. ترکیب این کلیدها، عملگرهای بولین و کارکترهای جایگزین[۱۴]، رشته جستجوی نهایی ایجاد می شود:

(real time OR real tiem data warehous*) AND
(stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND
(join OR process* OR analy* OR problem*)



شکل۸. جستجوی کلمات کلیدی با استفاده از پایگاه دانش برای شناسایی آثار

جدول ۸، جملات تحقیق نهایی را نشان می دهد که برای جستجو در کتابخانه های دیجیتال بیان شده مورد استفاده قرار گرفته اند. انتخاب بر اساس شامل بودن در محدوده و خارج بودن از محدوده تحقیق می باشد. شامل بودن را مواردی مانند بودن در محدوده پردازش بلادرنگ جریان داده، هدف

[14] Wildcard

پرسش های تحقیق، نتایج شامل مقالات ترویجی باشند و می بایست درمورد توسعه و برنامه های پردازش بلادرنگ بحث و بررسی نموده باشد می توان نام برد. همچنین انگلیسی نبودن و مرتبط با مباحث پردازش بلادرنگ جریان در انباره داده یا کلان داده ها، خارج از محدوده تحقیق می باشند. همچنین پرسش های تحقیق مورد ارزیابی قرار گرفته اند. برای مثال، در پاسخ به پرسش اول در جدول ۲ که شامل : کدام انتشارات مرتبط با تحقیق پردازش جریان بلادرنگ می باشند؟، باید گفت تحلیل وجود توسعه و چالش ها برای پردازش بلادرنگ جریان داده، چالش کلیدی برای توسعه تکنولوژی های هوش تجاری است. تشخیص و شناسایی انتشارات با کیفیت بالا و تحلیل علم سنجی بر اساس فرااطلاعات در ناحیه پردازش بلادرنگ جریان مورد نیاز می باشد.

جدول۸. استراتژی های جستجو برای کتابخانه های دیجیتال

کتابخانه دیجیتال	جمله جستجو	فیلترهای اعمال شده
ACM digital library	[[All: real time] OR [All: real time data warehous*]] AND [[All: stream] OR[All: etl] OR [All: challenges] OR [All: extract*,transform*,load*]] AND [[All: join] OR [All: process*] OR [All: analy*] OR [All: problem*]]	[(01/01/2011) TO 01/31/2020]
IEEEXplore	(real time OR real time data warehouse*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analytics OR problems)	2011- Jan2020
ScienceDirect	(real time OR real time data warehouse) AND (stream OR semi-stream OR ETL OR challenges OR (extraction,transformation,loading)) AND (join OR process OR analysis OR problems)	2011- Jan2020
SpringerLink	`(real time OR real time data warehouse*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analy* OR problem*)`	Computer Science, (2011- Jan2020)
IGI Global	(real time) AND (data warehousing or stream processing)	Individual Journal Articles (2011- Jan2020)
Inderscience Online	(real time OR real time data warehouse*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analytics OR problems)	2011- Jan2020
Hindawi	“real time stream processing OR real time data warehouse”	2011- Jan2020
MDPI	“real time stream processing”	2011- Jan2020
arXiv	“real time data warehouse OR real-time stream processing”	2011- Jan2020
Taylor & Francis Online	(real time OR real time data warehouse*) AND (stream OR semi-stream OR ETL OR challenges OR (extract*,transform*,load*)) AND (join OR process* OR analy* OR problem*)	2011- Jan2020

۷- نتیجه گیری

سیستم های دسته محور، برای دهه ها، وزنه های سنگینی بودند اما برای داده هایی با ماهیت پیوسته و نامحدود که در بسیاری از کاربردهای دنیای واقعی تولید می شوند، کاربردی ندارد. سیستم های جریان محور یا به عبارتی دیگر، سیستم هایی که به محض دریافت داده آنها را پردازش می کنند از تناسب طبیعی بیشتری با داده های کلانی که همه روزه در حال تولید هستند، برخوردار هستند[11] . در این میان نیاز به پلت فرم هایی برای مدیریت مقدار زیاد داده در زمینه های مشخص شده در یک روش کاملا توزیع شده می باشد، رویکردی که داده کاوی سنتی قادر به انجام آن نبود و به یک گلوگاه عملکردی غیرقابل قبول تبدیل شده بود[۲۰]. در این بررسی به اهمیت پردازش جریان داده و حفظ ارزش های پنهانی آن پرداخته شد. مروری بر تکنیک های کلان داده داشتیم، در نتیجه تقاضای قابل توجهی برای برنامه های کلان وجود دارد[۱۶]. روش هایی برای داده ها با حجم زیاد و قواعدی که بتوان با داده های آماری و یادگیری ماشین و تجزیه و تحلیل الگوها به کشف دانش پرداخت. سپس به نقش پردازش جریان داده در بازاریابی اشاره شد و اینکه تا چه حد ابزارهایی مانند شبکه های اجتماعی و داده های تولید شده توسط آنها می تواند منجر به سودآوری و افزایش وفاداری مشتریان در کسب و کارهای مختلف گردد. نقش جریان داده برای پردازش زبان طبیعی مورد بررسی قرار گرفت. با استفاده از خط لوله پردازش می توان به تحلیل متون همچون -فارسی پرداخت. پردازشگرهای بلادرنگ و جریان محوری مانند Storm، Samza، Spark Streaming و Strom Trident بررسی شدند و قابلیت ها و امکانات و نحوه پردازش جریان داده توسط آنها بیان شد. در ادامه، دو مفهوم اصلی سیستم های مدرن پردازش جریان و موتورهای پردازش جریان داده ارائه گردید. مولفه های شکل دهنده آنها و نحوه دریافت داده در طول زمان بررسی شدند و در نهایت یک تحقیق ساختارمند که بصورت سیستماتیک به جمع آوری مقالات در حوزه پردازش جریان داده ها و انبار داده بلادرنگ می پردازد، بیان شد. حرکت از اهداف تحقیق آغاز گردید، پرسش هایی مرتبط با حوزه مورد نظر برای رسیدن به مقالات و تحقیقات مرتبط ایجاد گردید. با بررسی مقالات موجود در چندین کتابخانه دیجیتال و فیلتر نمودن موارد خارج از محدودی تحقیق به ۷۲ مقاله مرتبط در فاصله بین سال های ۲۰۱۱ تا ۲۰۲۰ با

موضوع تحقیق رسیدیم. با رشد نمایی داده های جریان و نیاز به پردازش های لحظه ای و ارسال بازخورد به کاربران، مطمئنا با چالش هایی در بخش منابع همچنان مواجه هستیم که این موضوع نیاز به فراهم کردن راه های نو و تازه برای استفاده از جریان پیوسته که بصورت لحظه ای تولید می شود، در آینده ای نزدیک خواهد شد.

۸- منابع

۱. کلکلی, منصور و رجایی, امیر ,پردازش داده های کلان(Big data) ۱۳۹۶, کنفرانس سالانه پارادایم های نوین مدیریت در حوزه هوشمندی,تهران.

۲. رضایی, اکرم و حق جو, مصطفی, الگوریتمی برای پردازش سریع جریان کلان داده, ۱۳۹۷, کنفرانس ملی پیشرفت های اخیر در مهندسی و علوم نوین,قرچک.

۳. زاهد, عطیه و سخی, محمدرضا, مروری بر تکنیک های کلان داده (Big Data), ۱۳۹۵, سومین کنفرانس ملی مهندسی برق و کامپیوتر سیستمهای توزیع شده و شبکه های هوشمند,کاشان.

4. Bahri, M., Bifet, A., Gama, J., and Gomes, H. M., and Maniu, S. 2021. Data stream analysis: Foundations, major tasks and tools. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11(3): e1405.
5. Mittal, V., Gangodkar, D., & Pant, B. 2021. Big Data Technologies: A Comprehensive Survey. Inventive Communication and Computational Technologies, 103-114.
6. Mehmood, E., & Anees, T. 2020. Challenges and solutions for processing real-time big data stream: A systematic literature review. IEEE Access, 8, 119123-119143.
7. Isah, H., Abughofa, T., Mahfuz, S., Ajerla, D., Zulkernine, F., & Khan, S. 2019. A survey of distributed data stream processing frameworks. IEEE Access, 7, pp: 154300-154316.
8. Alghushairy, O. S. 2021. A Genetic Algorithm-Based Local Outlier Factor for Efficient Big Data Stream Processing (Doctoral dissertation, University of Idaho).
9. MODI, J. Big Data Stream Processing in Marketing Analytics.
10. Agerri, R., Artola, X., Beloki, Z., Rigau, G., & Soroa, A. 2015. Big data for Natural Language Processing: A streaming approach. Knowledge-Based Systems, 79, 36-42.
11. Wingerath, W., Gessert, F., Friedrich, S., & Ritter, N. 2016. Real-time stream processing for Big Data. it-Information Technology, ٥٨(4), 186-194.
12. Dubey, A. K., Gupta, R., & Mishra, S. 2021. Data Stream Clustering for Big Data Sets: A comparative Analysis. In IOP Conference Series: Materials Science and Engineering (Vol. 1099, No. 1, 012030).
13. Fan, W., & Bifet, A. 2013. Mining big data: current status, and forecast to the future. ACM SIGKDD explorations newsletter, 14(2), 1-5.
14. Abualigah, L., Gandomi, A. H., Elaziz, M. A., Hamad, H. A., Omari, M., Alshinwan, M., & Khasawneh, A. M. 2021. Advances in meta-heuristic optimization algorithms in big data text clustering. Electronics, *10*(2), 101.
15. Affetti, L., Margara, A., & Cugola, G. 2020. TSpoon: Transactions on a stream processor. Journal of Parallel and Distributed Computing, ١٤٠, 65-79.
16. Alwasel, K., Calheiros, R. N., Garg, S., Buyya, R., Pathan, M., Georgakopoulos, D., & Ranjan, R. 2021. BigDataSDNSim: A simulator for analyzing big data applications in software- defined cloud data centers. Software: Practice and Experience, ٥١(5), 893-920.
17. Sahal, R., Breslin, J. G., & Ali, M. I. 2020. Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. Journal of Manufacturing Systems, ٥٤, 138-151.

18. Henning, S., & Hasselbring, W. 2021. Theodolite: Scalability benchmarking of distributed stream processing engines in microservice architectures. Big Data Research, ۲۵, 100209.
19. Pishgoo, B., Azirani, A. A., & Raahemi, B. 2021. A hybrid distributed batch-stream processing approach for anomaly detection. Information Sciences, ۵۴۳, 309-327.
20. Esposito, C., Ficco, M., Palmieri, F., & Castiglione, A. 2015. A knowledge-based platform for big data analytics based on publish/subscribe services and stream processing. Knowledge-Based Systems, ۷۹, 3-17.