سامانه‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند
Intelligent Multimedia Processing and Communication Systems (IMPCS)

Summer 2023, 4 (2), 43-53
DOR:

Received: 10 Apr 2023
Accepted: 23 May 2023

مقاله پژوهشی

Description of the Distributed Architecture for Processing Streaming Social Network Data

Binazir Ganji[1], Ali Rezaee[*], Sahar Adabi[3], Ali Movaghar[4]

1. Ph.D. Student, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. binazir.ganji@srbiau.ac.ir
2. Assistant Professor, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. [*] *Corresponding Author*, alirezaee@srbiau.ac.ir
3. Assistant Professor, Department of Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran. sahar_adabi@iau-tnb.ac.ir
4. Professor, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. movaghar@sharif.edu

Abstract

Introduction: To analyze big data, especially streaming social network data, we require real-time and distributed systems to process streaming data with high speed and efficiency. In this paper, a distributed architecture for collecting, ingesting, processing, storing and visualizing streaming social network data based on Kappa architecture is introduced. Also, the proposed architecture includes a component for detecting anomalous data.

Method: We utilize the 4+1 architectural view models to visually illustrate the various architectural layers, components, and their interactions.

Results: The proposed architecture serves as a distributed solution designed for processing streaming social network data. We utilized the 4+1 architectural view model and UML diagrams to outline proposed architecture. This documentation clearly outlines the data processing pipeline and specifies both functional and non-functional system requirements.

Discussion: The proposed architecture is designed to process streaming social network data, leveraging distributed and parallel solutions for improved efficiency. Anomaly detection is a pivotal component integrated within the architecture to identify outlier data, enhancing processing precision and quality. By utilizing the 4+1 architectural view model and UML diagrams, the proposed architecture is effectively outlined, ensuring a well-defined structure that aids in organizing information. This structured approach provides stakeholders with tailored architectural views that cater to their individual needs and priorities. Notable functional requirements include real-time processing, while non-functional requirements encompass scalability, interoperability, portability, usability, and efficiency.

Keywords: streaming social networks, stream data processing, distributed architecture, 4+1 architectural view model, UML diagrams.

سامانه‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند
Intelligent Multimedia Processing and Communication Systems (IMPCS)
انجمن علمی تجارت الکترونیکی ایران
واحد زنجان

توصیف معماری توزیع‌شده برای پردازش داده‌های شبکه‌های اجتماعی جریانی

تاریخ دریافت: ۲۱/ ۰۱/ ۱۴۰۲
تاریخ پذیرش: ۱۴۰۲/۰۳/۰۲
دورهٔ چهارم، تابستان ۱۴۰۲
شمارهٔ دوم، صص: ۴۳-۵۳

بی‌نظیر گنجی[۱]، علی رضایی[۲*]، سحر آدابی [۳]، علی موقر [۴]

۱. دانشجوی دکترا، گروه مهندسی کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران. binazir.ganji@srbiau.ac.ir
۲. استادیار، گروه مهندسی کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران. (نویسندهٔ مسئول)، alirezaee@srbiau.ac.ir
۳. استادیار، گروه مهندسی کامپیوتر، واحد تهران شمال، دانشگاه آزاد اسلامی، تهران، ایران. sahar_adabi@iau-tnb.ac.ir
۴. استاد، گروه مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران. movaghar@sharif.edu

چکیده: برای تجزیه و تحلیل کلان داده‌ها از جمله داده‌های شبکه‌های اجتماعی جریانی، به سیستم‌های بلادرنگ و توزیع‌شده برای دریافت، پردازش و نمایش نتایج تجزیه و تحلیل جریان داده با سرعت و کارایی بالا، ضمن حفظ توالی داده‌های ورودی نیاز داریم. در این مقاله، یک معماری توزیع‌شده برای جمع‌آوری، جذب، پردازش، ذخیره و نمایش نتایج حاصل از پردازش داده‌های شبکه اجتماعی جریانی برپایه معماری کاپا، پیشنهاد شده‌است. باتوجه به اینکه داده‌های ناهنجار، برکیفیت و دقت نتایج حاصل از پردازش داده‌ها، تأثیر منفی می‌گذارند، لذا در معماری پیشنهادی، مؤلفه تشخیص داده‌های ناهنجار نیز در نظر گرفته‌شده‌است. ما لایه‌های مختلف معماری، اجزا و تعاملات آن‌ها را با استفاده از مدل نمای معماری ۴+۱ و نمودارهای UML مربوط به هر نما، توصیف‌می‌کنیم. براساس مستندات حاصل، جنبه‌های مختلف این معماری بررسی و مهم‌ترین آن‌ها استخراج‌می‌شود. یک ساختار واضح از معماری ارائه‌شده که به سازماندهی اطلاعات کمک‌می‌کند، دید لازم از معماری، به ذی‌نفعان داده‌می‌شود و آن‌ها این فرصت را دارند که روی دیدگاهی که بیشترین ارتباط را با آن‌ها دارد، تمرکزکنند و نیازهای عملکردی و غیرعملکردی سیستم، مشخص‌می‌شود.

واژه‌های کلیدی: شبکه‌های اجتماعی جریانی، پردازش داده‌های جریانی، معماری توزیع‌شده، مدل نمای معماری ۴+۱، نمودارهای UML.

۱. مقدمه

شبکه‌های اجتماعی، متشکل از انواع ابزارهای تحت وب هستند که کاربران خود را قادرمی‌سازند تا ایده‌ها و اطلاعات جدید را در یک محیط تعاملی و مجازی به اشتراک بگذارند[۱]. تجزیه و تحلیل داده-های بزرگ، از جمله شبکه‌های اجتماعی، به یک زیرساخت سخت-افزاری مقیاس‌پذیر با قابلیت‌های پردازش موازی نیاز دارد. این سیستم باید حافظه، پهنای باند، توان عملیاتی کافی داشته‌باشد و بتواند چندین کار را همزمان انجام‌دهد و در عرض چند ثانیه، پردازش موازی الگوریتم‌های تحلیلی پیشرفته را انجام‌دهد. تکنولوژی‌های سنتی، به‌دلیل عدم مقیاس‌پذیری، داشتن ساختاری متمرکز و ساختار داده‌های یک ریخت، کارایی لازم برای پردازش کلان داده‌ها را ندارند. فضای ذخیره‌سازی مقیاس بزرگ و منابع محاسباتی قدرتمند دو چالش اصلی در پردازش این نوع کلان‌داده‌ها هستند[۲].

محاسبات کلان‌داده، از جمله تجزیه و تحلیل شبکه‌های اجتماعی، باید به‌صورت پردازش توزیع‌شده و بر روی خوشه‌ای از رایانه‌ها پیاده‌سازی‌شود. ایجاد یک سیستم ذخیره‌سازی داده‌های بزرگ که بتواند حجم زیادی از داده‌های متنوع (متن، صدا، تصویر، ویدئو...) را جذب‌کند، پرهزینه و زمان‌بر است. در این سیستم‌ها، داده-ها را می‌توان به‌صورت جریانی و دسته‌ای واردکرد. سرعت، یک چالش برای فرآیند خط لوله[۱]داده‌است. با پیچیده‌ترشدن داده‌ها در سیستم پردازش کلان‌داده، توسعه آن سیستم نیز دشوارتر می‌شود. به‌خصوص برای پردازش بلادرنگ داده‌ها، در شبکه‌های اجتماعی، جذب داده‌ها باید متناسب با سرعت ورود داده ها به سیستم باشد. در غیر این صورت ممکن است، بخشی از داده‌ها وارد خط لوله داده نشوند.

هنگامی که ما با منابع داده‌های مختلف در شبکه‌های اجتماعی کارمی‌کنیم، کیفیت داده‌ها یک چالش است. فرمت‌های داده ناسازگار، داده‌های تکراری و مقادیر ازدست‌رفته و نیز داده‌های غیرعادی، تجزیه و تحلیل را غیرقابل‌اعتماد می‌کند. اطلاعات تولید-شده توسط سیستم‌هایی مانند شبکه‌های اجتماعی، دارایی عظیمی برای تبدیل داده‌ها به ارزش است، اما به‌دلیل عدم وجود فناوری‌های مناسب برای مدیریت این داده‌ها، نمی‌توان از پتانسیل این داده‌ها، به‌طور کامل استفاده‌کرد. چارچوب‌های کلان‌داده، فرصت‌هایی برای مدیریت و تجزیه و تحلیل حجم زیادی از داده‌ها ارائه‌می‌دهند. اما قابلیت‌های آن‌ها بیشتر بر روی سرعت، تحمل خطا و موازی‌سازی متمرکز است که در مرحله دریافت داده‌ها و پردازش آن‌ها، اهمیت بیشتری دارد. اما نکته مهم این است که اگر داده‌های بی‌کیفیت و غیرعادی به این سیستم پردازشی داده‌شود، نتایج تحلیل از دقت کافی برخوردار نبوده و باعث‌می‌شود تا کاربران نسبت به اطلاعات استخراج‌شده از عملیات پردازشی، اعتماد خود را ازدست‌دهند. بنابراین، تجزیه و تحلیل داده‌ها باید پس از آماده‌سازی مناسب و حذف داده‌های ناهنجار، انجام‌شود.

نوشتن کد برای دریافت، استخراج، پاک‌کردن و بارگذاری داده‌ها می‌تواند دشوار باشد. زیرا حجم داده‌های شبکه‌های اجتماعی، افزایش یافته و بسیار متنوع شده‌است. بنابراین باید حرکت به سمت پردازش خودکار داده‌ها انجام‌شود. در تحلیل شبکه‌های اجتماعی با حجم زیادی از داده‌ها مواجهیم و نیاز به معماری مقیاس‌پذیر است که در برابر تغییرات پایدار بوده و به‌راحتی قابل‌توسعه باشد. افزایش سرعت این محاسبات، نیازمند راه‌حل‌های توزیع‌شده و موازی است[3, 4].

بنابراین، برای حل بسیاری از چالش‌های فوق، در این مقاله، یک معماری توزیع‌شده برای پردازش داده‌های شبکه اجتماعی جریانی با درنظرگرفتن حذف داده‌های ناهنجار، پیشنهاد شده‌است.

قسمت‌های اصلی این مقاله، به شرح زیر است:

- ارائه یک معماری برای جمع‌آوری، جذب، پردازش، ذخیره و نمایش داده‌های شبکه‌های اجتماعی جریانی که قابلیت تشخیص داده-های غیرعادی را دارد.
- برای بهبود در دسترس بودن[۲] و عملکرد[۳]، این معماری، توزیع شده‌است و قابلیت استفاده از ابزارهایی با کارکرد توزیع‌شده و الگوریتم-های موازی را دارد. همچنین، یک مؤلفه تشخیص ناهنجاری، برای بهبود دقت و کیفیت نتایج پردازش، درنظر گرفته شده‌است.
- معماری پیشنهادی با استفاده از مدل نمای معماری۱+۴[۴] مبتنی بر نمودارهای UML، مدل‌سازی و توصیف شده‌است.

ساختار مقاله به شرح زیر است: بخش ۲ نمای کلی از کارهای مرتبط را ارائه‌می‌دهد، بخش ۳ معماری پیشنهادی و لایه‌های مختلف آن نشان‌می‌دهد. دربخش۴ نماهای ۱+۴ مبتنی برUML، ارائه‌شده است. در نهایت، بخش ۵ نتیجه‌گیری و کار آینده را تشریح می‌کند.

۲. کارهای مرتبط

در این بخش خلاصه‌ای از تحقیقات مرتبط ارائه‌شده‌است. در این تحقیقات از دو معماری برای پردازش کلان‌داده‌های جریانی استفاده شده‌است: معماری لامبدا [5] و معماری کاپا [6]. معماری لامبدا از تکنیک‌های پردازش دسته‌ای و جریانی، برای تجزیه و تحلیل کلان-داده‌ها استفاده می‌کند. در مقابل، معماری کاپا تنها بر روی جریان و پردازش بلادرنگ بدون لایه دسته‌ای، تمرکز دارد.

در [7]، یک معماری پردازش داده‌های بزرگ پنج لایه، به نام BDPA معرفی شده‌است. این معماری شامل یک لایه جمع‌آوری، لایه ذخیره‌سازی، لایه پردازش، لایه تحلیل و لایه کاربردی است تا یک رویکرد استاندارد برای جمع‌آوری، مدیریت، پردازش و تجزیه و تحلیل حجم زیادی از داده‌های ایستا یا پویا را ایجادکند. این راهکار برپایه معماری Lambda است. محدودیت‌هایی مثل در نظرنگرفتن سرعت داده‌ها، فرکانس و توالی داده‌ها پس از دریافت، ناتوانی در حذف داده‌های پرت در آن وجوددارد. چارچوبی بلادرنگ توسط Laska و همکاران [8] برای پردازش داده‌های جریان مکانی-زمانی دستگاه‌های

IoT پیشنهاد شده‌است. دراین چارچوب از Apache Kafka برای دریافت داده و از Apache Storm برای پردازش داده استفاده‌می‌شود. مزایای کلیدی این چارچوب شامل پردازش بلادرنگ، توزیع، تحمل خطا و استفاده از سیستم کارگزار پیام است.

در پایان‌نامه Fotiadis[9]، راه‌حلی برای پردازش بلادرنگ داده‌ها، بر اساس معماری کاپا و پلت فرم ابری Microsoft Azure ارائه‌شده‌است. این راه‌حل، شامل دریافت داده‌های جریانی از چندین منبع مستقل و مسیریابی آن به‌عنوان رویداد به زیرسیستم صف پیام است. یک جزء تجزیه و تحلیل جریان، به‌عنوان یک زیرسیستم مصرف‌کننده، عمل‌می‌کند و رویدادها را از صف پیام می‌خواند. رابط کاربری توییتر برای دریافت داده‌ها، استفاده‌می‌شود. در مقابل، Azure Event Hub برای دریافت داده‌ها، Azure Stream Analytics برای تجزیه و تحلیل داده‌ها و Power BI برای نمایش نتایج استفاده می‌شود. استفاده از پلتفرم ابری، در این راه‌حل، مزایای متعددی مانند قابلیت اجرای پردازش توزیع‌شده، مدیریت پردازش از طریق یک لایه پردازش واحد، مقرون‌به‌صرفه بودن، مقیاس‌پذیری و تحمل خطا را ارائه‌می‌دهد.

در مقاله [10]، نویسندگان یک چارچوب بلادرنگ برای پردازش، تبدیل، ذخیره و نمایش داده‌های جریانی توییتر پیشنهادکردند. این چارچوب از فناوری‌های منبع باز مانند آپاچی کافکا برای دریافت توییت‌ها، Spark Streaming برای پردازش جریان‌های داده و پایگاه داده Cassandra برای ذخیره توییت‌ها استفاده‌می‌کند. هدف این کار، استفاده از چارچوب یکپارچه پیشنهادی، برای انجام تحلیل و پردازش بلادرنگ توییت‌ها است. معماری ارائه‌شده در این مقاله، می‌تواند برای انواع داده‌های جریانی مختلف استفاده‌شود و دارای تحمل خطای بالا، قابلیت‌های پردازش بلادرنگ، مقیاس‌پذیری و استفاده از پایگاه داده برای ذخیره توییت‌ها برای پردازش مجدد است. چارچوب جدیدی برای پیش‌بینی کلان‌داده‌های بزرگ، در شبکه‌های IoT بر اساس Apache Spark در[۱۱] معرفی شده‌است. این چارچوب از چهار جزء اصلی تشکیل شده‌است: معماری شبکه اینترنت اشیا، معماری جریان داده‌های بزرگ، مدل پیش‌بینی سری زمانی برای جریان داده‌ها و یک روش پیش‌بینی جریان داده‌های بزرگ. معماری پیشنهادی، از معماری کاپا برای پردازش داده‌های جریانی استفاده‌می‌کند. با کمک Apache Spark و کتابخانه Mlib آن، الگوریتم‌های مختلف یادگیری ماشین بر روی داده‌ها به‌صورت توزیع‌شده و موازی با سرعت بالا اجرامی‌شوند. الگوریتم رگرسیون افزایشی خودکار، برای مدل‌های پیش‌بینی، استفاده‌شده‌است. معماری پیشنهادی بلادرنگ، قابل انطباق و با کارایی بالا است و قادر به مدیریت طیف وسیعی از الگوریتم‌های طبقه‌بندی و خوشه‌بندی است. شایان ذکراست که جذب داده در این معماری انجام‌نمی‌شود و عدم وجود کارگزار پیام، منجر به ازدست‌رفتن داده‌ها می‌شود. علاوه بر این، تشخیص نقاط پرت در نظر گرفته‌نمی‌شود.

Khriji و همکاران [12]یک معماری شبکه بی‌سیم متصل به ابر، به نام REDA را ارائه‌کرده‌اند که برای پردازش بلادرنگ داده‌های شبکه حسگر بی‌سیم جهت کنترل رطوبت خاک استفاده‌می‌شود. معماری شامل چندین مؤلفه است که با واحد سنجش شروع‌می‌شود و داده‌ها را جمع آوری‌کرده و از طریق واحد دروازه به سرویس ابری AWS ارسال‌می‌کند. پردازش جریان رویداد، با استفاده از آپاچی کافکا انجام‌می‌شود و نتایج با استفاده از پایگاه داده MongoDB ذخیره‌می‌شوند. معماری پیشنهادی، انعطاف‌پذیراست و می‌تواند حداکثر ۸۰۰۰ پیام در ثانیه را با تأخیر کم، ذخیره‌کند. این مطالعه اثربخشی معماری REDA را برای پردازش بی‌درنگ داده‌های شبکه حسگر بی‌سیم، در کنترل رطوبت خاک نشان‌می‌دهد. Folino و همکاران [13]، چارچوبی را برای پردازش بلادرنگ و تشخیص موارد پرت در داده‌های گزارش کاربر، بر اساس پشته ELK و Kubernetes ارائه‌می‌کنند. پشته ELK از ابزارهای منبع باز مانند Elasticsearch، Logstash و Kibana برای جستجو، تجزیه و تحلیل و نمایش داده‌ها استفاده‌می‌کند. در چارچوب Kubernetes، داده‌ها باید در سیستم فایل توزیع‌شده، ذخیره‌شوند. برای تشخیص ناهنجاری‌ها در رفتار کاربران، یک الگوریتم طبقه‌بندی بر اساس سناریوی تشخیص بالماسکه پیشنهاد شده‌است. این چارچوب از چندین ابزار پردازش داده‌های بزرگ برای پخش داده‌ها و تشخیص ناهنجاری استفاده می‌کند. با این حال، هیچ راه‌حلی برای پردازش توزیع‌شده و دریافت جریان داده ارائه‌نشده‌است. نویسندگان مقاله [14]، یک سیستم توزیع‌شده برای پردازش داده‌های جریان مربوط به Covid-19 با استفاده از داده‌های شبکه اجتماعی توییتر پیشنهاد کرده‌اند. روش پیشنهادی از Apache Pulsar به‌عنوان یک سیستم پیام‌رسانی توزیع‌شده، استفاده‌می‌کند. راه‌حل ارائه‌شده در این مقاله شامل تجزیه و تحلیل فضایی مجموعه داده است و برای پردازش داده‌های بزرگ جریانی، توزیع‌شده و مقیاس‌پذیر است. در جدول ۱، مقایسه بین ویژگی‌های معماری کارهای مرتبط با معماری پیشنهادی انجام شده‌است.

۳. معماری پیشنهادی

دراین بخش، معماری مرجع سیستم پیشنهادی، معرفی‌می‌شود و به دنبال آن یک تصویر دقیق از لایه‌ها، اجزاء و تعاملات معماری پیشنهادی با استفاده از نمودارهای UML ارائه‌می‌شود. ما از مدل نمای معماری 4+1 برای به‌تصویرکشیدن معماری پیشنهادی خود استفاده‌کرده‌ایم.

جدول ۱: مقایسه ویژگی‌های معماری کارهای مرتبط با معماری پیشنهادی

مقاله	معماری مرجع	توزیع شده	کارگزار پیام	پردازش موازی	تشخیص ناهنجاری	توصیف معماری
[7] (2019)	Lambda	✓	-	✓	-	-
[۹] (2021)	Kappa	✓	✓	✓	-	-
[۸] (2018)	Kappa	✓	✓	✓	-	-
[۱۰] (2021)	Kappa	✓	✓	✓	-	-
[۱۵] (2022)	Lambda/ Kappa	✓	✓	✓	-	-
[۱۲] (2022)	Kappa	✓	✓	✓	-	-
[13] (2023)	Kappa	✓	-	✓	✓	-
[14] (2023)	Kappa	✓	✓	✓	-	-
معماری پیشنهادی	Kappa	✓	✓	✓	✓	✓

۱،۳. نمای کلی معماری پیشنهادی

معماری Lambda[3] یا معماری Kappa[6] را می‌توان برای پردازش جریان داده‌های بزرگ استفاده‌کرد. معماری لامبدا، یک چارچوب قوی برای پردازش مقادیر زیادی داده با استفاده از روش‌های پردازش دسته‌ای و جریانی است. این معماری از سه لایه مجزا تشکیل‌شده‌است: لایه دسته‌ای، لایه سرعت و لایه سرویس. لایه دسته‌ای، مسئول ذخیره مجموعه داده‌های اصلی و محاسبه منظم توابع دلخواه در آن مجموعه داده است.

لایه سرعت، عملکردهای محاسباتی را بر روی داده‌ها در زمان واقعی انجام‌می‌دهد. در نهایت، لایه سرویس نتایج پردازش در دو لایه قبلی را در قالب گزارش‌های گرافیکی در اختیار کاربران قرارمی‌دهد. این ابزارها درک مجموعه داده‌های پیچیده و استخراج بینش‌های معنی‌دار را آسان‌می‌کنند [5]. معماری کاپا، شبیه معماری لامبدا است. با این تفاوت که شامل لایه دسته‌ای نیست، بنابراین داده‌ها را در زمان واقعی پردازش‌می‌کند. معماری کاپا ساده، انعطاف‌پذیر و مناسب برای پردازش آنلاین جریان‌های داده است[6, 16]. از معماری کاپا، به‌عنوان معماری مرجع برای معماری پیشنهادی این تحقیق، استفاده‌شده‌است. شکل۱، با الهام از معماری کلان داده Microsoft Azure [17] قسمت‌های مختلف معماری پیشنهادی را نشان‌می‌دهد. در این شکل، پردازش دسته‌ای از معماری کلان داده Azure درنظرگرفته‌نشده‌است و تمرکز بر پردازش جریانی است. از طرفی مؤلفه‌ای برای تشخیص داده‌های ناهنجار درنظرگرفته‌شده‌است.

لایه‌های مختلف معماری پیشنهادی شامل موارد زیر است:

- **لایه جمع‌آوری داده[۵]:**

لایه جمع‌آوری داده‌ها ازAPIهای مرتبط برای جمع‌آوری داده‌های ساختاریافته، بدون ساختار و نیمه ساختاریافته از شبکه‌های اجتماعی استفاده‌می‌کند.

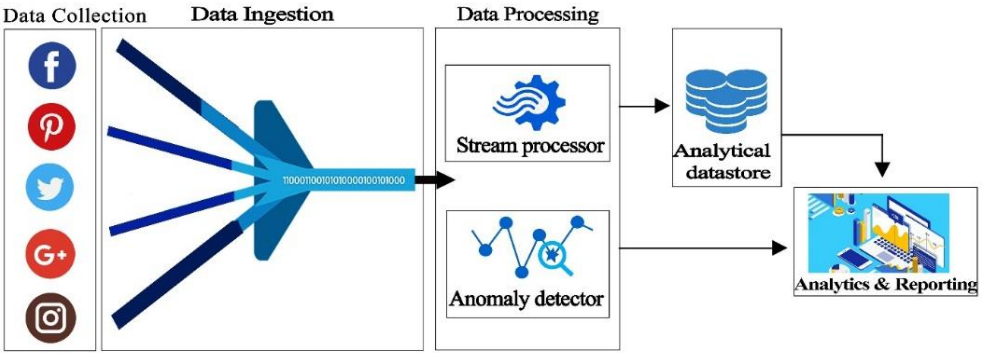
- **لایه جذب داده[۶]:**

دریافت داده، شامل انتقال انواع داده‌های مختلف به اجزای سیستم برای پردازش است. سیستم‌های پردازش جریان توزیع‌شده در زمان واقعی، معمولاً از سیستم‌های واسطه پیام برای ذخیره داده‌ها قبل از پردازش استفاده‌می‌کنند. مکانیسم‌های دریافت پیام، می‌توانند تعادل بار و تحمل خطا را بر اساس اندازه، فرمت، فرکانس و سرعت داده‌ها بهبود بخشند[18]. یکی از ابزارهایی که می‌توان در این لایه استفاده کرد، آپاچی کافکا [19] است که به‌طور گسترده‌ای به‌عنوان یک سیستم کارگزار پیام با توان عملیاتی بالا، شناخته‌می‌شود.

- **لایه پردازش داده[۷]:**

در این لایه، عملیات‌های متعددی از جمله پردازش، پیش پردازش، ادغام، نمایه‌سازی و تشخیص ناهنجاری انجام‌می‌شود [۲۰]. در معماری پیشنهادی، این لایه شامل دو جزء است:

۱. مؤلفه پردازش: این مؤلفه، می‌تواند عملیات پردازشی را به صورت توزیع شده و موازی با سرعت بالا، انجام‌دهد. یکی از ابزارهای قابل استفاده در این لایه، Apache Spark است که یک موتور پردازش توزیع‌شده است و با استفاده از روش نگاشت-کاهش[۸] الگوریتم‌های پردازشی را به‌صورت بلادرنگ، انجام‌می‌دهد.



شکل ۱: معماری پیشنهادی برای پردازش داده‌های شبکه‌های اجتماعی جریانی [۱۷]

۲. مؤلفه تشخیص ناهنجاری: نقاط داده‌ای که به‌طور قابل‌توجهی با سایر داده‌های مجموعه داده متفاوت هستند، داده‌های ناهنجار می‌باشند[20]. این ناهنجاری‌ها، بر کیفیت و دقت پردازش تأثیر می‌گذارند. مؤلفه تشخیص ناهنجاری، داده‌های ناهنجار را با کمک الگوریتم‌های یادگیری ماشین تشخیص‌می‌دهد و دقت و کیفیت تحلیل را افزایش می‌دهد.

- **لایه پایگاه داده تحلیلی[۹]:**

پایگاه‌های داده تحلیلی، مقادیر زیادی از داده‌های یک سازمان را ذخیره‌می‌کنند که برای به‌دست‌آوردن بینش در مورد تجارت، مشتریان و موارد دیگر استفاده‌می‌شود. در این پایگاه‌ها، اطلاعات مرتباً به‌روز می‌شوند. انواع پرس‌وجوها، کاربران را قادر می‌سازد تا داده‌ها را از پایگاه داده بازیابی‌کنند، بدون اینکه ساختار داخلی پایگاه داده را بدانند[21].

- **لایه تجزیه، تحلیل و گزارش[۱۰]:**

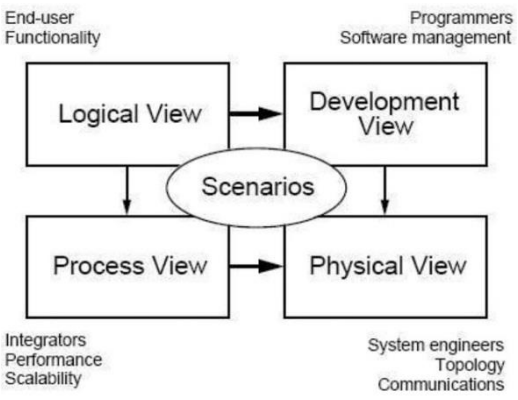
در این لایه، نتایج تحلیل داده‌ها در اختیار کاربران قرارمی‌گیرد. برای این کار، ابزارهای مربوط باید از برنامه‌های تحلیلی استاندارد و تعریف‌شده توسط کاربر و مدل‌های داده‌کاوی تطبیقی با قابلیت اتصال به زبان‌های سطح بالا پشتیبانی‌کنند[7]. معماری پیشنهادی، امکان جمع‌آوری، جذب، پردازش، ذخیره‌سازی و نمایش داده‌های شبکه اجتماعی را برای تجزیه و تحلیل فراهم‌می‌کند. این امر شامل، استفاده از تکنیک‌های نمایش داده‌های بزرگ مانند نمودارها، نقشه‌ها، نمودارهای جعبه و ...است. با این تکنیک‌ها، کاربران می‌توانند بینش‌های ارزشمندی را در مورد فرآیندهای اساسی به‌دست‌آورند.

از ویژگی‌های اساسی معماری پیشنهادی، می‌توان به موارد زیر اشاره‌کرد:

۱. این معماری، براساس معماری کاپا برای پردازش داده‌های جریانی از جمله داده‌های شبکه‌های اجتماعی جریانی، پیشنهادشده‌است.
۲. درمعماری پیشنهادی، حجم زیادی ازجریان‌های داده، بدون توجه به سرعت یا فرمت آن‌ها، به‌طور مؤثر پردازش می‌شوند. این معماری، می تواند داده های ساختاریافته، نیمه ساختاریافته و بدون ساختار را در اشکال و ساختارهای مختلف، مدیریت‌کند.
۳. پردازش را می‌توان بدون نیاز به ذخیره کل داده‌ها انجام‌داد.
۴. در این معماری، با استفاده از مؤلفه جذب داده‌ها، سرعت ورود داده‌ها با سرعت پردازش سیستم متناسب است. این امر، تضمین می‌کند که هیچ داده‌ای در طول فرآیند پردازش از بین نمی‌رود و تمام داده‌های دریافتی به‌طور مؤثر، پردازش و ذخیره‌می‌شوند.
۵. این معماری بر روش‌های توزیع‌شده و الگوریتم‌های موازی، برای پردازش کارآمد داده‌ها متکی است. ابزارهای محبوب مانند Apache Spark و Apache Kafka را می‌توان به‌راحتی در سیستم، برای بهبود تحمل خطا و توزیع وظایف پردازش، استفاده‌کرد. نتایج حاصل از پردازش، کاربران را قادرمی‌سازد تا تصمیمات سریع و آگاهانه اتخاذکنند.
۶. در معماری پیشنهادی، یک مؤلفه برای تشخیص داده‌های ناهنجار در نظر گرفته شده‌است. با حذف داده‌های پرت، سیستم می‌تواند نتایج دقیق و قابل‌اعتمادتری تولیدکند و کیفیت کلی فرآیند تجزیه و تحلیل داده‌ها، افزایش‌می‌یابد.

۴. نماهای معماری پیشنهادی

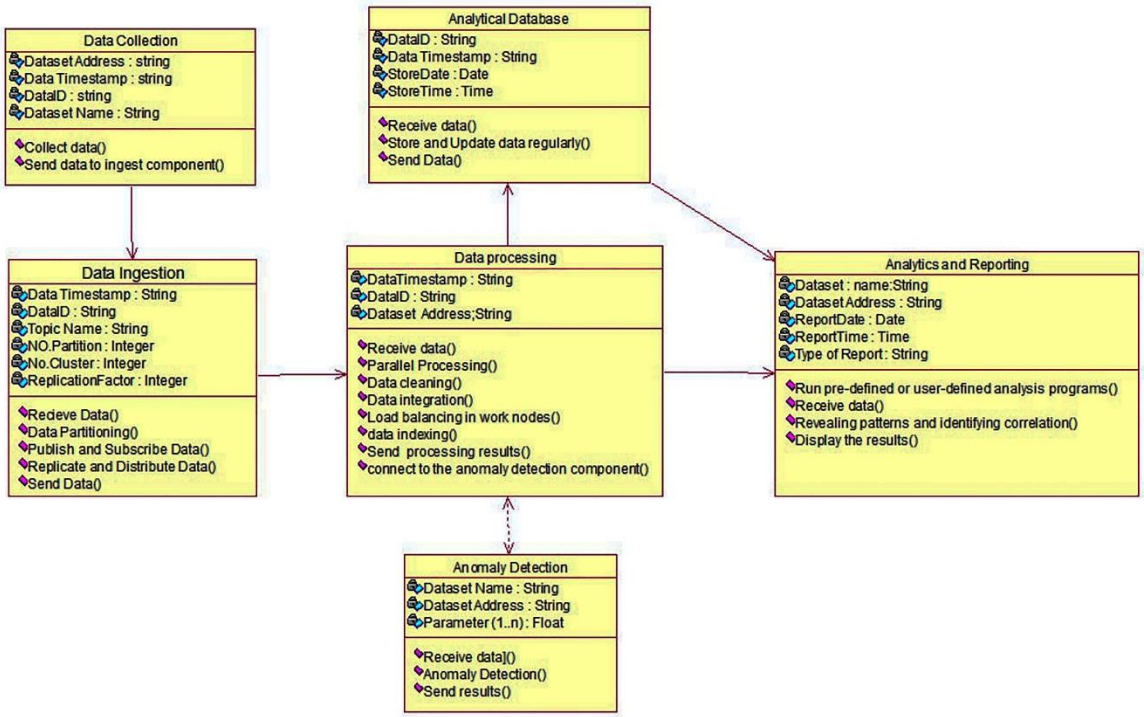
دراین بخش، نماهای معماری پیشنهادی با استفاده از نمودارهای UML و مدل نمای معماری ۴+۱ ارائه شده‌است. ۴+۱ یک مدل نما، برای توصیف معماری سیستم‌های نرم‌افزاری، با استفاده از نماهای متعدد و هم‌روند است. نماها برای توصیف سیستم، از دیدگاه ذینفعان مختلف، مانند کاربران نهایی، توسعه‌دهندگان، مهندسین سیستم و مدیران پروژه استفاده‌می‌شوند. در مدل ۴+۱ نماهای مورد نظر شامل نمای منطقی، نمای توسعه، نمای سناریو، نمای فرایند و نمای فیزیکی می باشند [22] [23]. در شکل ۲، مدل نمای معماری ۴+۱ آمده‌است.



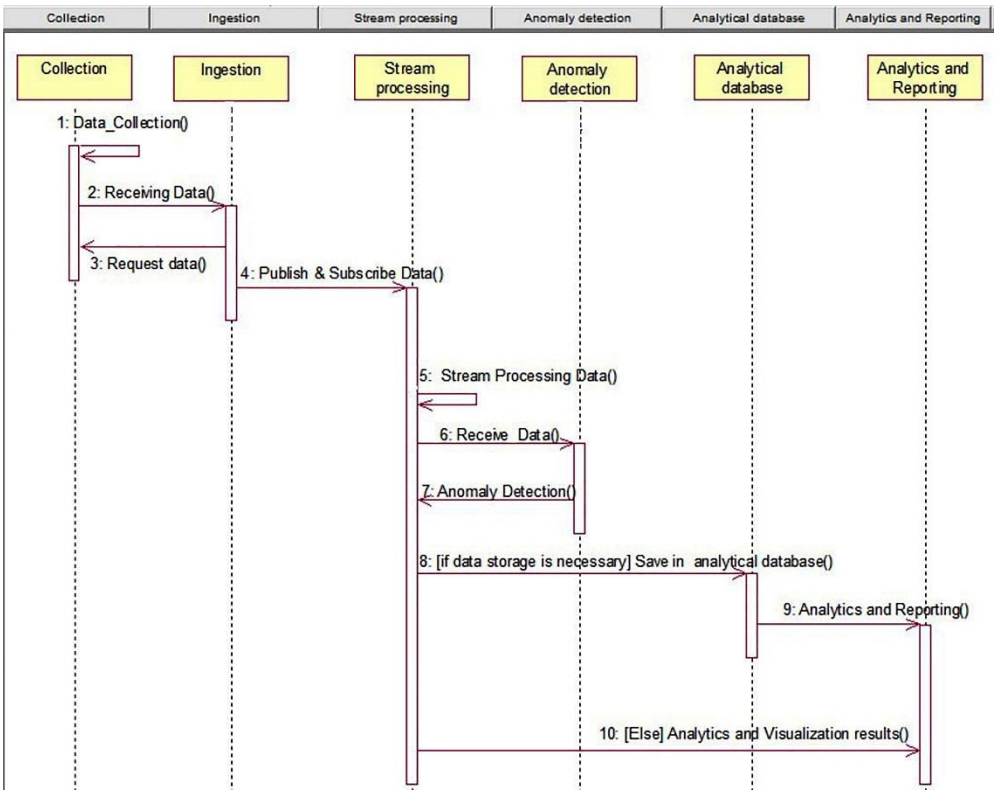
شکل ۲: مدل نمای معماری۴+۱ [22]

۱،۴. نمای منطقی

دیدگاه منطقی، مربوط به عملکردی است که سیستم در اختیار کاربران نهایی قرارمی‌دهد. برای نمایش نمای منطقی، از نمودارهای UML استفاده‌می‌شود و شامل نمودارهای کلاس و نمودارهای توالی است[۲۴]. نمودارهای کلاس، یک نمای کلی ساختاری از سیستم را ارائه‌می‌دهند و هر کلاس، روابط مربوطه را برجسته‌می‌کند. نمودارهای توالی، تعاملات بین مؤلفه‌ها را در طول زمان و جریان فعالیت را در هر مرحله نشان‌می‌دهند. این نمودارها، به درک عملکرد سیستم و نحوه پردازش داده‌ها کمک‌می‌کنند. نمودار کلاس در شکل ۳ و نمودار توالی در شکل۴ مربوط به معماری پیشنهادی آمده‌است.



شکل ۳. نمودار کلاس معماری پیشنهادی



شکل ۴. نمودار توالی معماری پیشنهادی

۲،۴. نمای توسعه

این دیدگاه، به توسعه‌دهندگان و مدیران نرم‌افزار یک دید کلی از وضعیت سیستم از منظر پیاده‌سازی ارائه‌می‌دهد که معمولاً نمای پیاده‌سازی نامیده‌می‌شود. برای توصیف اجزای سیستم، UML از یک نمودار مؤلفه استفاده‌می‌کند. این سیستم، شامل چندین مؤلفه ضروری

است که هر کدام ممکن است در برخی موارد دارای مؤلفه‌های فرعی باشند. انواع مؤلفه های اصلی مورد استفاده در معماری پیشنهادی، در جدول ۲ و نمودار مؤلفه‌ها در شکل ۵، ارائه شده‌است.

۳,۴. نمای سناریو

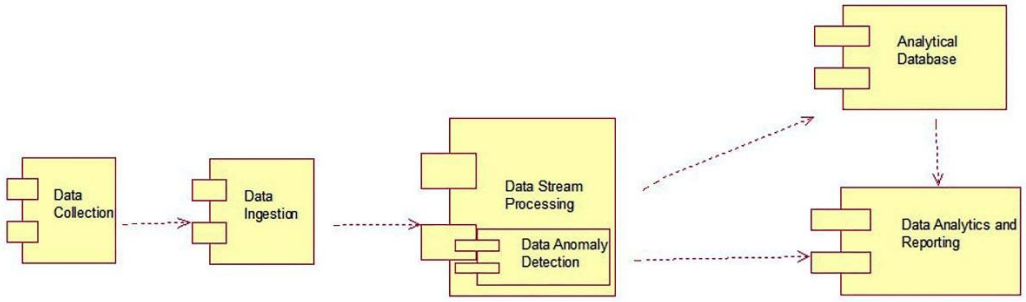
این دیدگاه، معماری را از طریق مجموعه‌ای از سناریوها توصیف‌می‌کند که هر کدام دنباله‌ای از تراکنش‌های وابسته به یکدیگر را نشان‌می‌دهد که هم توسط سیستم و هم توسط کنشگران، انجام‌می‌شوند و یک دیدگاه سیستمی سطح بالا، را از دیدگاه کنشگران ارائه‌می‌دهد و در تجزیه و تحلیل سیستم، برای استخراج نیازمندی‌ها و نشان‌دادن عملکرد سیستم مفید هستند[۲۴]. نمودار use case مربوط به نمای سناریو، در شکل ۶ نشان‌داده‌شده‌است.

۴,۴ . نمای فرآیند

نمای فرآیند با جنبه‌های دینامیکی سیستم، سروکار دارد. فرآیندهای سیستم و ارتباطات بین آن‌ها را توصیف‌می‌کند. همچنین بر رفتار سیستم در زمان اجرا تمرکز دارد[۲۴]. شکل ۷، نمودار فعالیت مربوط به نمای مذکور، را نشان‌می‌دهد.

۵,۴. نمای فیزیکی

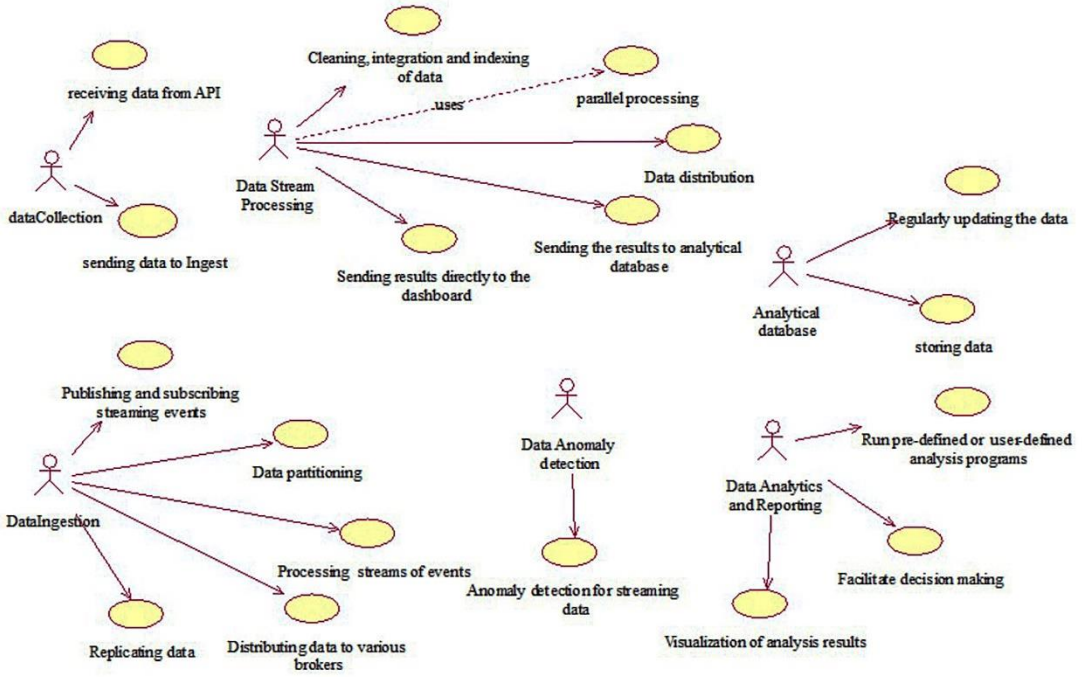
نمای فیزیکی، آرایش ساختاری لایه فیزیکی نرم افزار را نشان‌می‌دهد که شامل فعل و انفعالات فیزیکی بین اجزای آن می‌شود. نمودار استقرار، برای نشان‌دادن این دیدگاه استفاده‌می‌شود[۲۴]. در شکل ۸، نمودار استقرار معماری پیشنهادی، نشان‌داده‌شده‌است.



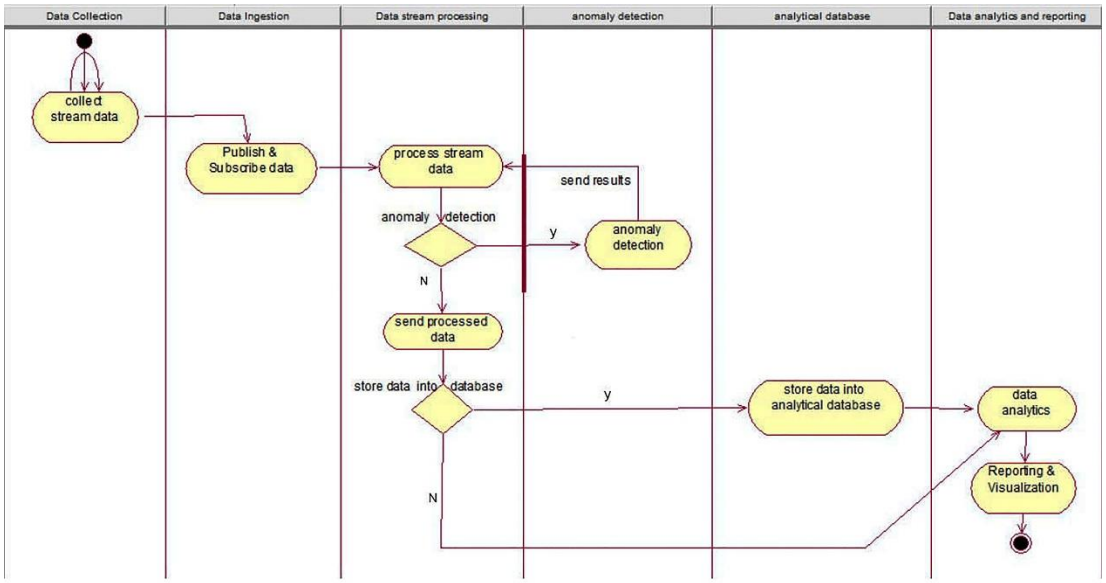
شکل۵. نمودار مؤلفه سیستم پیشنهادی

جدول ۲: مؤلفه‌های معماری پیشنهادی

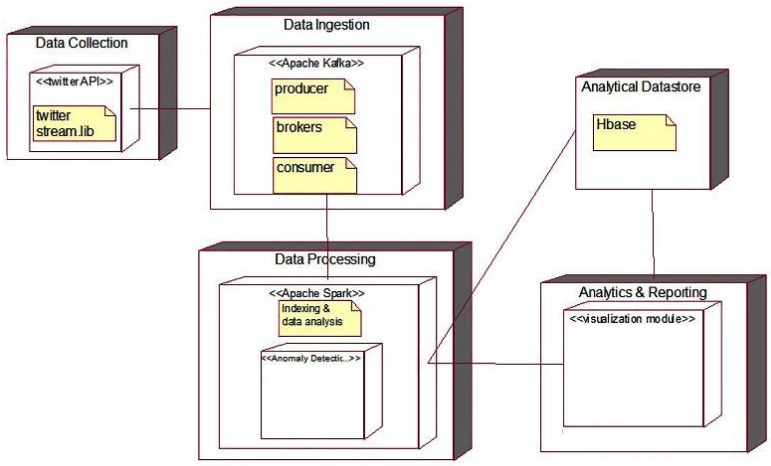
نام مؤلفه	وظیفه
جمع آوری داده	دریافت داده‌ها از API، ارسال داده به مؤلفه جذب داده
جذب داده	تکرار داده‌ها برای تحمل خطا و در دسترس بودن، انتشار و اشتراک رویدادهای جریان، پارتیشن‌بندی داده‌ها بر اساس مهر زمانی، تامین در دسترس بودن داده‌ه ، پردازش رویدادهای جریان توزیع داده‌ها به کارگزاران مختلف، تعادل بار در خوشه‌های مختلف
پردازش داده‌های جریانی	پردازش موازی داده‌های جریانی، پاکسازی، یکپارچه‌سازی، و نمایه‌سازی داده‌ها، توزیع داده‌ها، افزایش تحمل خطا، تأمین در دسترس بودن داده‌ها، تعادل بار، ارسال نتایج پردازش به پایگاه داده تحلیلی یا مستقیماً به مؤلفه تحلیل و گزارش‌گیری
تشخیص ناهنجاری	تشخیص ناهنجاری با سرعت بالا، افزایش دقت و کیفیت نتایج تجزیه و تحلیل
پایگاه داده تحلیلی	ذخیره حجم عظیمی از داده‌ها، به‌روزرسانی منظم داده‌ها، تامین در دسترس بودن داده‌ها، ارسال داده‌ها یه مؤلفه تحلیل و گزارش‌گیری
تحلیل و گزارش‌گیری	مصورسازی نتایج تجزیه و تحلیل داده‌های ذخیره‌شده یا نمایش خروجی به‌صورت بلادرنگ، تسهیل و تسریع در تصمیم‌گیری کاربران، امکان استفاده از برنامه‌های تحلیل از پیش تعریف‌شده و همچنین برنامه‌های تحلیل تعریف‌شده توسط کاربر.



شکل۶. نمودار use case **درمعماری پیشنهادی**



شکل۷. نمودار فعالیت درسیستم پیشنهادی



شکل۸. نمودار استقرار سیستم پیشنهادی

۵. نتیجه‌گیری و کارهای آتی

در این مقاله، معماری پیشنهادی می‌تواند برای پردازش توزیع‌شده داده‌های شبکه‌های اجتماعی جریانی، استفاده‌شود. معماری پیشنهادی، بر اساس معماری کاپا است و می‌تواند داده‌های جریانی از جمله داده‌های شبکه‌های اجتماعی را جمع‌آوری، جذب، پردازش، ذخیره و نمایش‌دهد. مؤلفه تشخیص ناهنجاری، برای شناسایی داده‌های پرت در معماری پیشنهادی در نظرگرفته‌شده‌است که عملکرد آن دقت و کیفیت نتایج پردازش را افزایش‌می‌دهد. ما از مدل نمای ۴+۱ و نمودارهای UML مربوط به آن نما، برای به‌تصویرکشیدن معماری پیشنهادی خود، استفاده‌کرده‌ایم. معماری از نماهای مختلف مثل منطقی، توسعه، فرایند، سناریو و فیزیکی به کمک نمودارهای UML مربوطه، به‌صورت دقیق و از جنبه‌های مختلف توصیف‌شده‌است. بنابراین، ذی‌نفعان شناخت کافی از سیستم را به‌دست‌می‌آورند و ارتباط و همکاری بین ذی‌نفعان تسهیل‌می‌شود، چراکه آن‌ها برای بحث درباره سیستم، به مستندات یکسانی دسترسی دارند. براساس مستندات حاصل، خط لوله پردازش داده، مشخص‌شده‌است. مؤلفه‌ها و نحوه تعامل آن‌ها در نمودارهای مختلف ارائه‌شده‌اند. در برخی نمودارها، جزئیات عملکرد مؤلفه‌ها مورد توجه نیست، صرفاً چیدمان و نحوه پیاده‌سازی آن‌ها مدّنظر است، مثل نمودار مؤلفه یا استقرار. از طرفی برخی نمودارها، جزئیات مؤلفه‌ها و نحوه تعامل را نمایش‌می‌دهند، مثل نمودار کلاس یا فعالیت. باتوجه به این مستندات و ویژگی‌های معماری پیشنهادی، می‌توان نیازهای عملکردی و غیرعملکردی سیستم را مشخص‌نمود. از جمله نیازهای عملکردی این سیستم، پردازش بلادرنگ است. چراکه در این سیستم، باتوجه به استفاده از داده‌های جریانی و ماهیت متغیر آن‌ها و عدم امکان پردازش دسته‌ای، پردازش بلادرنگ داده‌ها ضروری است. که مؤلفه‌های جذب، پردازش و تحلیل و گزارش‌گیری، با استفاده از راهکارهای توزیع‌شده و الگوریتم‌های موازی، این نیاز را می‌توانند به‌خوبی فراهم‌کنند.

نیازهای غیرعملکردی می‌توانند شامل مقیاس‌پذیری، قابلیت همکاری بین مؤلفه‌ها، دسترس پذیری، قابل‌حمل بودن، سهولت در استفاده و کارایی باشند. این سیستم، با حجم متغیر و معمولاً رو به افزایش داده، روبرو است، که تغییر در حجم داده‌ها، نباید تأثیری در پیکربندی سیستم داشته‌باشد. این امر لزوم مقیاس‌پذیر بودن معماری را تأکیدمی‌کند. به‌دلیل حجم بالای داده‌های ورودی و نیاز به سرعت بالای پردازش، این معماری باید کارایی بالایی داشته‌باشد که قابلیت استفاده از راهکارهای توزیع‌شده و موازی، افزایش کارایی و نیز دسترس‌پذیری را میسر می‌کند. از طرفی به‌دلیل پردازش خودکار داده‌ها و عدم دخالت کاربردر خط لوله پردازش، مؤلفه‌ها باید قابلیت همکاری بالایی داشته‌باشند و به سکوی نرم‌افزاری خاصی وابسته نباشد و به‌سادگی قابل‌استفاده و قابل‌حمل باشند.

درتحقیقات آتی، می‌توان معماری پیشنهادی را به کمک آزمون‌های رسمی بدون درگیر شدن با پیچیدگی پیاده‌سازی معماری، بررسی کرده و آن را از دیدگاه‌های مختلف، ارزیابی‌کرد. همچنین می‌توان یک الگوریتم مؤثر و بلادرنگ، برای تشخیص ناهنجاری داده‌های جریانی پیشنهادداد و یا معماری را برای پردازش بلادرنگ داده‌ها و روش‌های کاهش زمان اجرا، مورد ارزیابی قرارداد.

مراجع

[۱] Bakeshlu, M. and M. Tahghighi Sharbyan (2023), A NewApproach Based on Deep Learning Algorithms to Study Effective Factors of Using Social Networks on Students' Performance, Intelligent Multimedia Processing and Communication Systems (IMPCS), 3(4)[Persian].

[۲] Jafari, M. and A. Kalbasi (2021), A Comparative Study of Open-Source Software for Deployment and Management of Cloud Computing Utilizing a Big Data Processing Quality Mode, Intelligent Multimedia Processing and Communication Systems (IMPCS), 2(4)[Persian].

[۳] B. Yadranjiaghdam, S. Yasrobi, and N. Tabrizi, "Developing a real-time data analytics framework for twitter streaming data". IEEE International Congress on Big Data (BigData Congress): p. 329-336, 2017.

[۴] M. Y. Amare and S. Simonova, "Learning analytics for higher education: proposal of big data ingestion architecture". SHS Web of Conferences. 92, 2021,DOI: https://doi.org/۱۰/۱۰۵۱/shsconf/.۲۰۲۱۹۲۰۲۰۰۲

[۵] J. Warren and N. Marz, *Big Data: Principles and best practices of scalable realtime data systems*. Simon and Schuster, 2015.

[۶] J. Kreps, *Questioning the Lambda Architecture*. 2014[cited 2024, Available from: https://www.oreilly.com/radar/questioning-the-lambda-architecture/.

[۷] J. Y. Zhu, B. Tang, and V. O. Li, "A five-layer architecture for big data processing and analytics". International Journal of Big Data Intelligence. 6(1): p. 38-49, 2019, DOI: ۱۰/۱۵۰۴/IJBDI..۲۰۱۹/۰۹۷۳۹۹

[۸] M. Laska, et al., "A scalable architecture for real-time stream processing of spatiotemporal IoT stream data—Performance analysis on the example of map matching". ISPRS International Journal of Geo-Information. **7**(7): p.238 , 2018.

[۹] O. Fotiadis , *Distributed stream and event processing pipeline in serverless architecture*. University of Piraeus (Greece), 2021.

[۱۰] S. Arora and R. Rani, "A Novel Framework for Distributed Stream Processing and Analysis of Twitter Data". International Conference on Innovative Computing and Communications: p. 147-161, 2021.

[۱۱] J. Rosandic, *Real-Time Streaming Data Management, Processing, Analysis and Visualisation*. University of Zagreb, 2022.

[۱۲] S. Khriji, et al., "Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks". The Journal of upercomputing. **78**(3): p.3374-3401 ,2022, DOI: ۱۰/۱۰۰۷/s.۶-۰۳۹۵۵-۰۲۱-۱۱۲۲۷

[۱۳] G. Folino, C. Otranto Godano, and F. S. Pisani, "An ensemble-based framework for user behaviour anomaly detection and classification for cybersecurity". The Journal of Supercomputing, 2023, DOI: ۱۰/۱۰۰۷/s-۰۲۳-۱۱۲۲۷ -۰۵۰۴۹x.

[۱۴] Y. M. Özgüven and S. Eken, "Distributed messaging and light streaming system for combating pandemics: A case study on spatial analysis of COVID- ۱۹Geo-tagged Twitter dataset". Journal of Ambient Intelligence and Humanized Computing., 14(2), p. 773-778, 2023, DOI:۱۰/۱۰۰۷/s.۰-۰۳۳۲۸-۰۲۱-۱۲۶۵۲

[۱۵] J. Rosandic, "Real-Time Streaming Data Management, Processing, Analysis and Visualization", 2022.

[۱۶] A. Barradas, A. Tejeda-Gil, and R.-M. Cantón-Croda, "Real-Time Big Data Architecture for Processing Cryptocurrency and Social Media Data: A Clustering Approach Based on k-Means". Algorithms. **15(5)**: p. 140, 2022, DOI: https://doi.org/۱۰/۳۳۹۰/a.۱۵۰۵۰۱۴۰.

[۱۷] *Big data architecture style*, [cited 2024, Available from: https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/big-data.

[۱۸] J., Alwidian, et al., "Big Data Ingestion and Preparation Tools". Modern Applied Science. **14**(9), 2020, DOI:۱۰/۵۵۳۹/mas.v۱۴n۹p.۱۲

[۱۹] Sax, M. J., *Apache Kafka*. Encyclopedia of Big Data Technologies, ed. S. Sakr and A. Zomaya. Cham: Springer International Publishing, 1-8, 2018.

[۲۰] M. R. Basirati, et al., "Understanding changes in use cases: A case study". IEEE 23rd International Requirements Engineering Conference (RE): p. 352-361, 2015.

[۲۱] D. Hawkins, "Identification of Outliers, Springer Netherlands",1980.

[۲۲] T. Contributor, *analytic-database*. 2024[cited 2024, Available from: https://www.techtarget.com/searchbusinessanalytics/definition/analytic-database.

[۲۳] P. B. Kruchten, "The 4+1 view model of architecture". IEEE software. **12**(6): p. 42-50, 1995.

[۲۴] M. Kontio, "Architectural Manifesto: Designing Software Architectures. Part 5. Introducing the 4+1 View Model". IBM developerWorks, 2005.

پی‌نوشت

[1] Pipeline
[2] Availability
[3] Performance
[4] 4+1 architectural view model
[5] Data collection layer
[6] Data ingestion layer
[7] Data processing layer
[8] Map-reduce
[9] Analytical data store layer
[10] Analytics and reporting layer