دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

پیاده‌سازی یک ابزار داده‌کاوی مبتنی بر آپاچی اسپارک برای داده‌های جاری

سینا شیخ‌الاسلامی
sinash@aut.ac.ir

استادان راهنما:

دکتر امیرحسین پی‌براه
دکتر سید رسول موسوی

دانشگاه صنعتی امیرکبیر
۹ تیر ۱۳۹۵

دانشکده مهندسی کامپیوتر
و فناوری اطلاعات

فهرست

- داده‌های جاری: کاربردها و چالش‌ها
- بسترهای پردازش داده‌های جاری
- مروری بر رابط برنامه‌نویسی اسپارک‌استریمینگ
- الگوریتم نمونه‌برداری تصادفی توزیع‌یافته با مخزن ثابت (DRSFR)
- طراحی و پیاده‌سازی ابزار
- جمع‌بندی و کارهای آینده

داده‌های جاری: کاربردها و چالش‌ها

- داده‌های جاری

* https://www.pehub.com/2014/03/thomson-reuters-partners-with-cambridge-associates-on-benchmark-data/

داده‌های جاری: کاربردها و چالش‌ها

- برخی کاربردهای داده‌های جاری

- شناسایی الگوهای لحظه‌ای جستجو در وب
- تشخیص موضوعات داغ در شبکه‌های اجتماعی
- نظارتِ پزشکی
- کنترل ترافیک هوشمند در شبکه‌های حمل و نقل
- پایش محیط زیستی
- شبکه‌های هوشمند انرژی
- تشخیص ناهنجاری در تراکنش‌های بانکی
- تشخیص حملات به شبکه‌های کامپیوتری

داده‌های جاری: کاربردها و چالش‌ها

- چالش‌های پردازش و کاوش داده‌های جاری

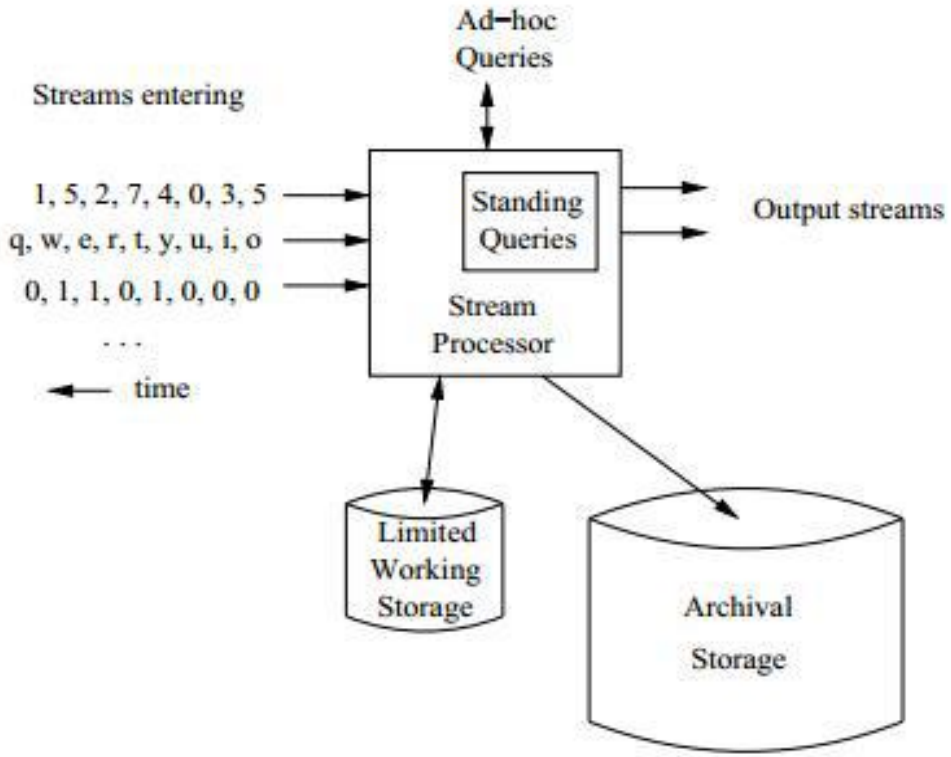
داده‌های جاری: کاربردها و چالش‌ها

- چالش‌های پردازش و کاوش داده‌های جاری

- نیاز به الگوریتم‌های تک-عبوره
- نیاز به پردازش و کاوش بهنگام یا کم‌تأخیر
- عدم امکان ذخیره‌ی همه‌ی داده‌ها برروی حافظه‌های انبوه و پایگاه‌داده‌ها
- امکان تغییر در نرخ ورود و حجم داده‌ها
- وقوع تحول در داده‌ها

بسترهای پردازش داده‌های جاری

- مدل کلاسیک پردازش داده‌های جاری



شکل ۱ – یک سامانه‌ی پردازش داده‌های جاری [۲]

بسترهای پردازش داده‌های جاری

- معماری بسترهای توزیع‌یافته‌ی پردازش داده‌های جاری

شکل ۲ – معماری بسترهای توزیع‌یافته‌ی پردازش داده‌های جاری [۱۴]

بسترهای پردازش داده‌های جاری

- بسترهای توزیع‌یافته‌ی مطرح پردازش داده‌های جاری
  - Apache Flink
  - Apache Storm
  - Apache Spark
  - Heron

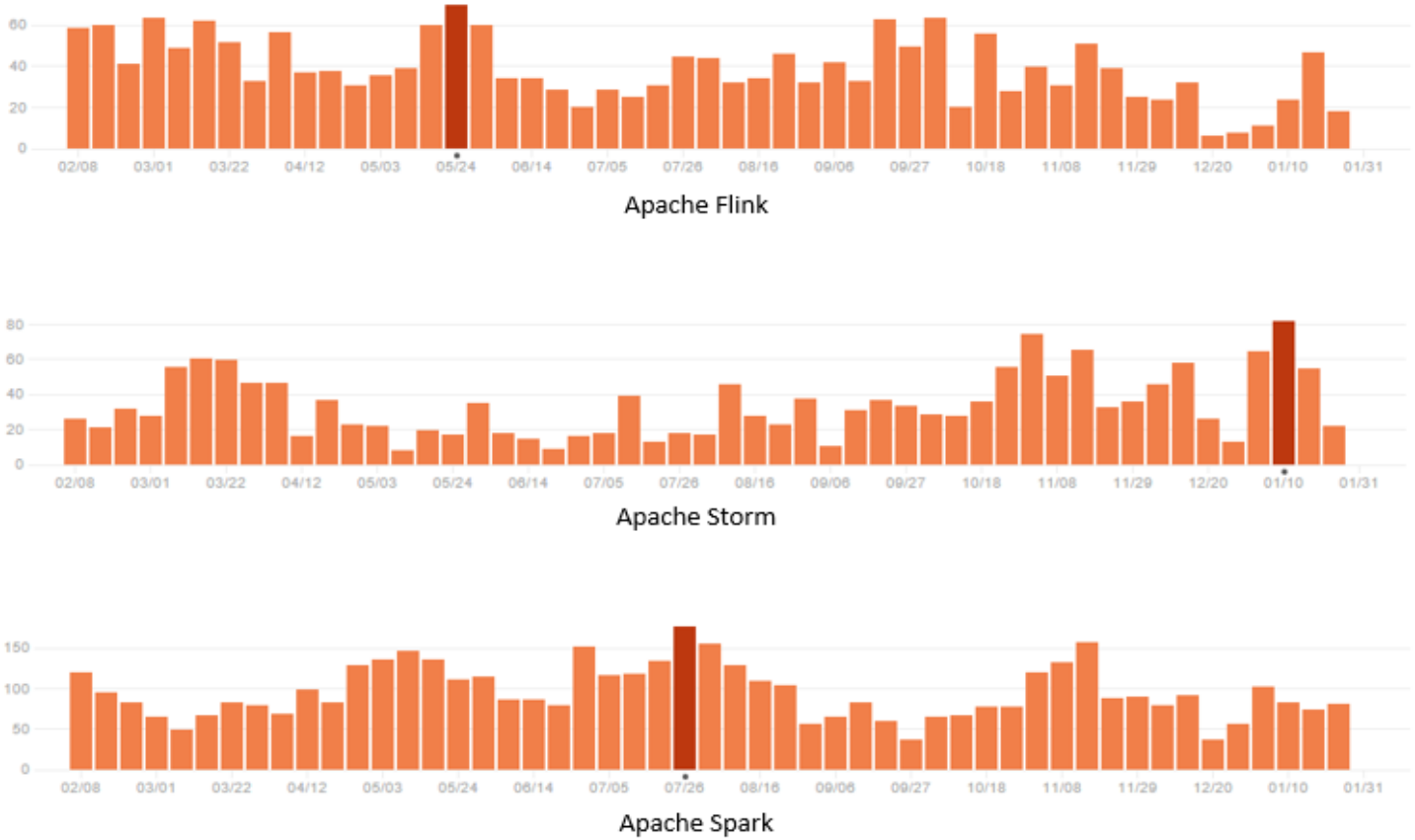
بسترهای پردازش داده‌های جاری

- انتخاب بستر پیاده‌سازی و اجرای الگوریتم‌ها
- معیارهای موردنظر:

- شرایط استفاده (آزاد بودن، متن‌باز بودن، گواهی‌های مورد استفاده)
- میزان فعال بودن جامعه‌ی توسعه‌دهندگان
- در دسترس بودن مستندات و منابع آموزشی

بسترهای پردازش داده‌های جاری

- انتخاب بستر پیاده‌سازی و اجرای الگوریتم‌ها (ادامه)



شکل ۳ – تعداد تغییرات اعمال شده در کد در هر هفته برای هر بستر در بازه‌ی فوریه ۲۰۱۵ تا ژانویه ۲۰۱۶

بسترهای پردازش داده‌های جاری

- انتخاب بستر پیاده‌سازی و اجرای الگوریتم‌ها (ادامه)

	آپاچی اسپارک	آپاچی استورم	آپاچی فلینک
توسعه‌دهندگان فعال	۷۹۷	۲۰۰	۱۵۴
سئوالات تگ‌شده در وب‌سایت stackoverflow.com	۹۹۰۰	۱۳۹۳	۲۰۸

شکل ۴ – مقایسه‌ی برخی ویژگی‌های مربوط به توسعه‌ی سه بستر (در تاریخ ۳۱ ژانویه ۲۰۱۶)

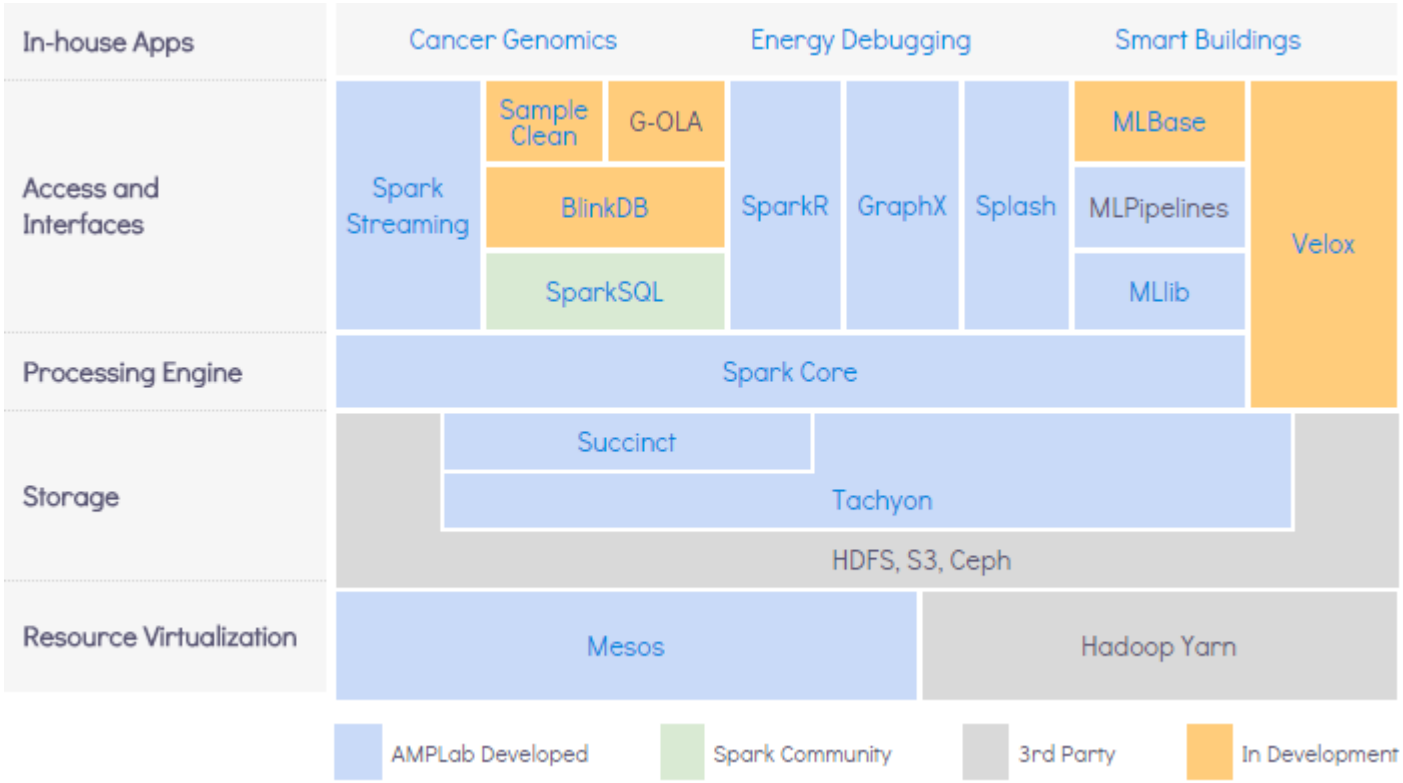
بسترهای پردازش داده‌های جاری

- انتخاب بستر پیاده‌سازی و اجرای الگوریتم‌ها (ادامه)

مروری بر رابط برنامه‌نویسی اسپارک‌استریمینگ

مروری بر رابط برنامه‌نویسی اسپارک‌استریمینگ

- جایگاه اسپارک‌استریمینگ در BDAS



شکل ۵ – استک تحلیل داده‌های برکلی (**BDAS**) [۱۰]

مروری بر رابط برنامه‌نویسی اسپارک‌استریمینگ

- مفاهیم و ساختارهای اساسی اسپارک

  - مجموعه‌داده‌ی ارتجاعی توزیع‌یافته (RDD)
  - تبدیل‌ها (Transformations)
  - اقدام‌ها (Actions)
  - ارزیابی تنبلِ تبدیل‌ها (Lazy Evaluation)
  - استفاده از زبان اسکالا (Scala)

مروری بر رابط برنامه‌نویسی اسپارک‌استریمینگ

- جریانِ ورودی و خروجی در اسپارک‌استریمینگ

شکل ۶ – جریان ورودی و خروجی اسپارک‌استریمینگ [۱۱]

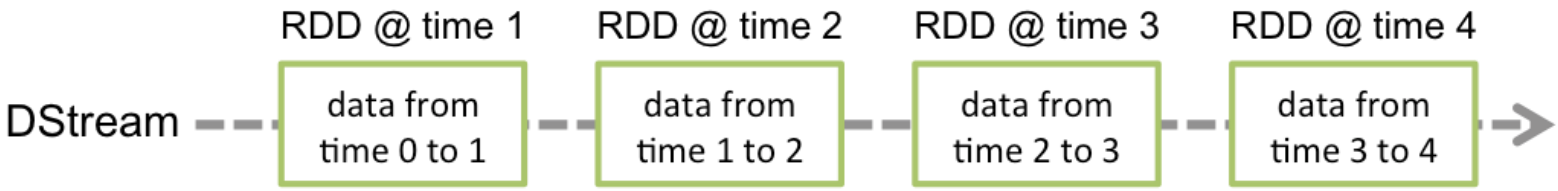
مروری بر رابط برنامه‌نویسی اسپارک‌استریمینگ

- تقسیم جریان‌داده به دسته‌های داده

شکل ۷ – تقسیم جریان‌داده‌ی ورودی به دسته‌های داده برای پردازش

مروری بر رابط برنامه‌نویسی اسپارک‌استریمینگ

- جریانِ گسسته‌شده (DStream)



شکل ۸ – جریانِ گسسته‌شده و RDDهای موجود در آن [۱۱]

خلاصه‌ی مباحث مطرح شده تا اینجا

- داده‌های جاری، کاربردها و چالش‌ها
- بسترهای پردازش داده‌های جاری
  - مدل کلاسیک پردازش داده‌های جاری
  - بسترهای توزیع‌یافته
  - انتخاب بستر پیاده‌سازی
- مروری بر رابط برنامه‌نویسی اسپارک‌استریمینگ

الگوریتم نمونه‌برداری تصادفی توزیع‌یافته با مخزن ثابت (DRSFR)

- خلاصه‌سازی، نمونه‌برداری و کاربردهای آن
- الگوریتم نمونه‌برداری تصادفی با مخزن ثابت (RSFR)
- الگوریتم نمونه‌برداری تصادفی توزیع‌یافته با مخزن ثابت (DRSFR)
  - برای داده‌های شماره‌گذاری شده
  - برای داده‌های بدون شماره

الگوریتم نمونه‌برداری تصادفی توزیع‌یافته با مخزن ثابت (DRSFR)

- کاربردهای خلاصه‌سازی (Synopsis Construction)
- برخی روش‌های خلاصه‌سازی
  - نمونه‌برداری
  - نمونه‌برداری تصادفی (Random Sampling)
  - نمونه‌برداری مختصر (Concise Sampling)
  - ساخت هیستوگرام (Histogram)
  - موجک‌ها (Wavelets)

الگوریتم نمونه‌برداری تصادفی توزیع‌یافته با مخزن ثابت (DRSFR)

- الگوریتم نمونه‌برداری تصادفی با مخزن ثابت

Random Sampling with a Fixed Reservoir (RSFR)

We have a reservoir of size **n**.
Input data comes in form of a stream of elements.

---

1 Add the first ***n*** elements of the data stream to the reservoir for initialization.

2 When the ***k***th element arrives, it is placed in the reservoir with a probability of ***n/k***.

3 If ***k***th element has to be added to the reservoir, an existing element of the reservoir with equal probability of 1/***n*** will be selected and removed from the stream, and the ***k***th element of input will replace it.

شکل ۹ – گام‌های اجرای الگوریتم RSFR

الگوریتم نمونه‌برداری تصادفی توزیع‌یافته با مخزن ثابت (DRSFR)

- الگوریتم نمونه‌برداری تصادفی با مخزن ثابت (ادامه)
  - اثبات بدون تبعیض (Unbiased) بودن
  - ماهیت غیر توزیع‌یافته
  - حساس بودن به ورود عضو جدید و تضاد با مدل برنامه‌نویسی اسپارک‌استریمینگ

الگوریتم نمونه‌برداری تصادفی توزیع‌یافته با مخزن ثابت (DRSFR)

- الگوریتم نمونه‌برداری تصادفی توزیع‌یافته با مخزن ثابت
Distributed Random Sampling with a Fixed Reservoir
  - موازی سازی
  - بازطراحی طراحی با توجه به مدل برنامه‌نویسی اسپارک‌استریمینگ
  - پیاده‌سازی
    - داده‌های شماره‌گذاری شده (Pre-Indexed)
    - داده‌های بدون شماره
  - عدم تغییر در منطق کلی و صدق‌کردن اثبات صورت گرفته برای بدون تبعیض بودن

Distributed Random Sampling with a Fixed Reservoir (DRSFR)

For Pre-Indexed Data Streams and Apache SparkStreaming

We have a reservoir of size **n**.

random is a floating point number between 0 and 1, randomly generated on each occurrence.

Input data comes in form of a stream of pre-indexed elements. Indices are based on the order of production of data elements.

Input elements are in the form of: **"value, index"**

---

foreach batch interval ***interval***, do the following:

1 **transform** the ***interval.DStream*** into another DStream by applying a **map** operation to the containing RDD, resulting in a new DStream containing a RDD with its elements in the form of (value, index), and name the new DStream as ***indexedDStream***.

2 **filter** ***indexedDStream*** **foreach** element ***e*** in ***indexedDStream*** that satisfies the following predicate, and name the filtered DStream as ***filteredDStream***.

$$(n \,/\, e.\text{index}) > random$$

3 **foreach** RDD ***r*** in ***filteredDStream***, do the following:

3.1 **collect** ***r*** inside an array named ***updateSet***.

3.2 **sort** ***updateSet*** in ascending order based on indices of elements of ***r***; call the new array as ***sortedUpdateSet***.

3.3 **foreach** element ***el*** in ***sortedUpdateSet***, do the following:

3.3.1 ***replaceIndex*** = a random integer between 0 and ***n***-1.

3.3.2 replace the reservoir element with the index of ***replaceIndex*** with ***el***.

شکل ۹ – گام‌های اجرای الگوریتم DRSFR برای داده‌های شماره‌گذاری شده

طراحی و پیاده‌سازی ابزار

- معماری کلی و پیکرپاره‌های ابزار SDMiner
  - هدف از ایجاد ابزار: ساده‌تر کردن تعریف و اجرای وظایف داده‌کاوی بر بستر آپاچی اسپارک
- مدل فرآیندی آبشاری
- برخی از مستندات تحلیل و طراحی

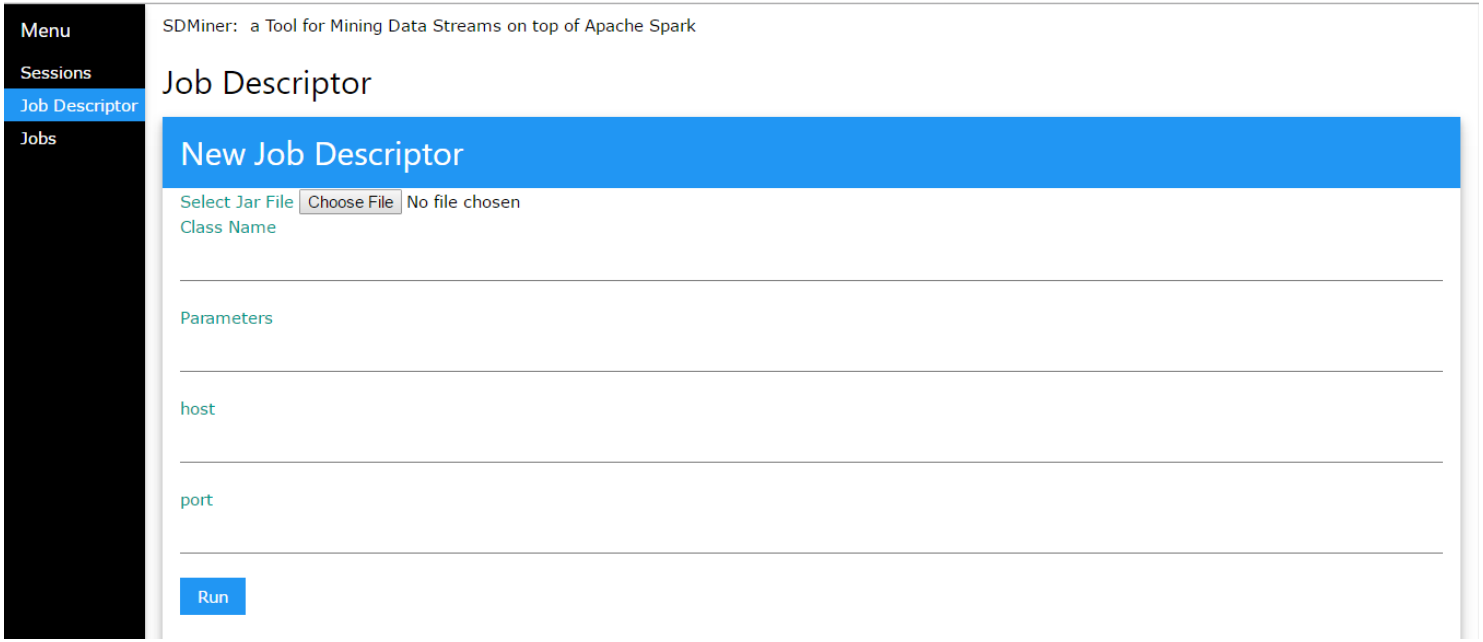
طراحی و پیاده‌سازی ابزار

- معماری و پیکرپاره‌های ابزار SDMiner

شکل ۱۰ – معماری لایه‌ای SDMiner

طراحی و پیاده‌سازی ابزار

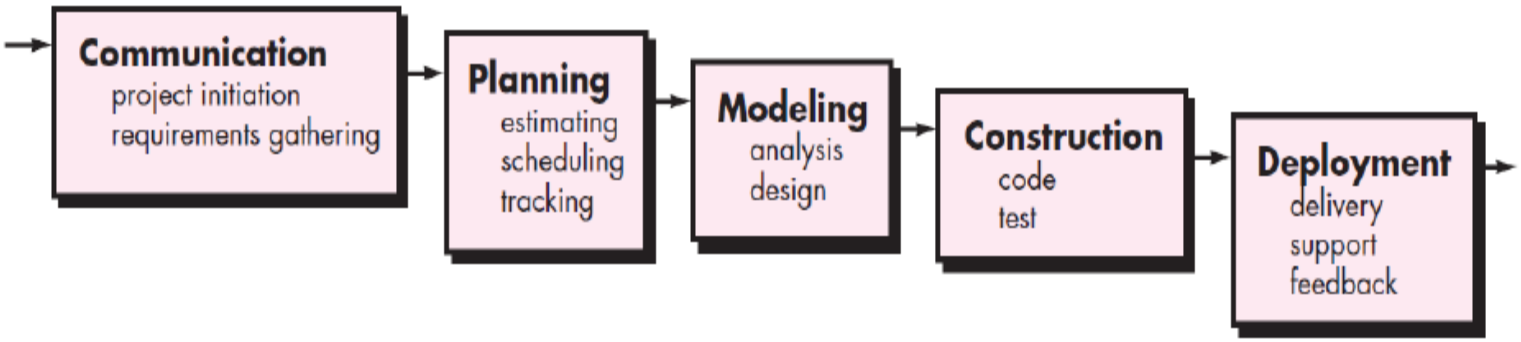
- رابط کاربری تحت وب



شکل ۱۱ – نمای تعریف وظایف داده‌کاوی در SDMiner

طراحی و پیاده‌سازی ابزار

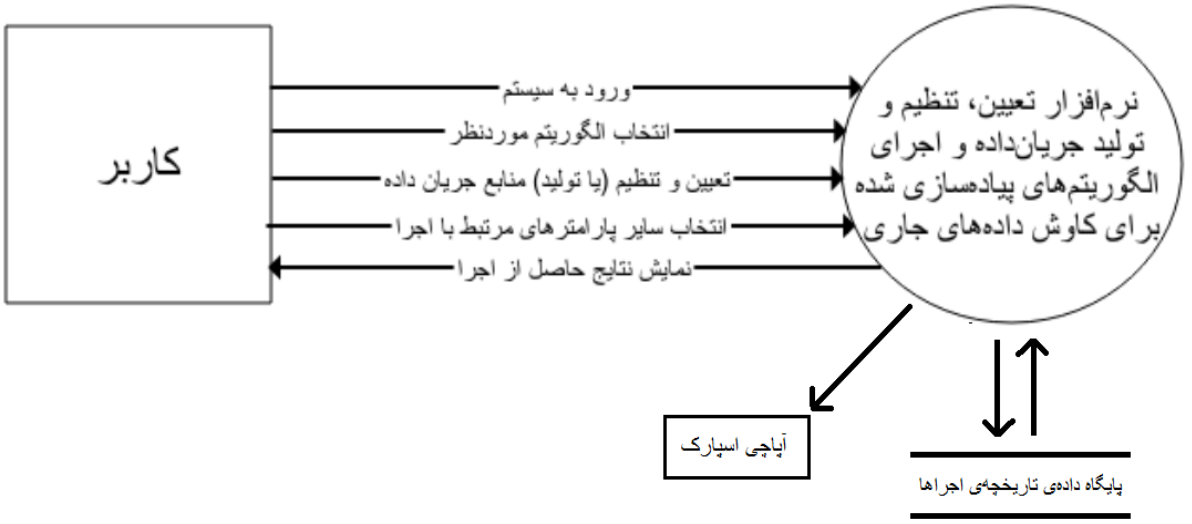
- مدل فرآیندی آبشاری
  - ارتباط
  - برنامه‌ریزی
  - مدل‌سازی
  - ساخت
  - استقرار



شکل ۱۲ – مدل فرآیندی آبشاری [۱۳]

طراحی و پیاده‌سازی ابزار

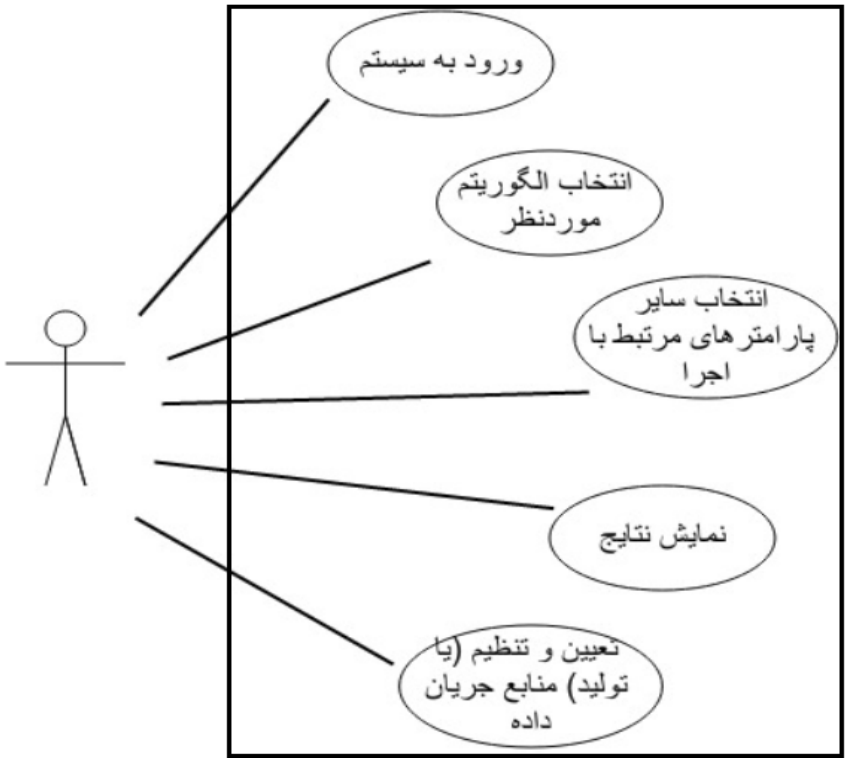
- مستندات تحلیل و طراحی: نمودار مفهومی سطح صفر



شکل ۱۳ – نمودار مفهومی سطح صفر

طراحی و پیاده‌سازی ابزار

- مستندات تحلیل و طراحی: نمودار مورد کاربرد



شکل ۱۴ – نمودار مفهومی سطح صفر

جمع‌بندی و کارهای آینده

- مرور مباحث مطرح شده
- کارهای قابل انجام برای توسعه‌ی این پروژه:
  - غِنابخشی به کتابخانه‌ی الگوریتم‌ها
  - بازطراحی الگوریتم‌ها به صورت توزیع‌یافته و مبتنی بر مدل برنامه‌نویسی اسپارک
  - بهبود رابط کاربری
    - استفاده از روش‌های مصورسازی (Visualization)
  - افزودن قابلیت نصب، تنظیم و راه‌اندازی اسپارک و لیوی بر روی خوشه‌های موردنظر
    - افزودن امکان میزان‌سازی (Tuning)
  - پشتیبانی از دریافت و پردازش هم‌زمان جریان‌داده‌های مختلف
  - پیشتیبانی از بسترهای توزیع‌یافته‌ی دیگر

منابع و مراجع

[۱] Aggarwal, Charu C. Data streams: models and algorithms. Vol. 31. Springer Science & Business Media, 2007.

[۲] Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. Mining of massive datasets. Cambridge University Press, 2014.

[۳] Andrade, Henrique CM, Buğra Gedik, and Deepak S. Turaga. Fundamentals of Stream Processing: Application Design, Systems, and Analytics. Cambridge University Press, 2014.

[۴] Han, Jiawei, Micheline Kamber, and Jian Pei. Data mining: concepts and techniques. Elsevier, 2011.

[۵] "Apache Flink: Scalable Batch and Stream Data Processing." Web. 31 Jan. 2016. <http://flink.apache.org/>.

[۶] "Apache Storm." Web. 31 Jan. 2016. <http://storm.apache.org/>.

[۷] Zaharia, Matei, et al. "Discretized streams: Fault-tolerant streaming computation at scale." Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles. ACM, 2013.

[۸] Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.

[۹]"Apache Spark™ - Lightning-Fast Cluster Computing." Web. 31 Jan. 2016. <http://spark.apache.org/>.

[۱۰]"BDAS, the Berkeley Data Analytics Stack." AMPLab UC Berkeley. Web. 31 Jan. 2016. <http://amplab.cs.berkeley.edu/software/>.

منابع و مراجع

[۱۱] “Spark Streaming | Apache Spark." Web. 31 Jan. 2016. <http://spark.apache.org/streaming/>.

[۱۲] Kreps, Jay, Neha Narkhede, and Jun Rao. "Kafka: A distributed messaging system for log processing." NetDB, 2011.

[۱۳] Pressman, Roger S. Software engineering: a practitioner's approach., 7th Edition, McGraw-Hill, 2009.

[۱۴] Kamburugamuve, Supun, and Geoffrey Fox. "Survey of Distributed Stream Processing.", 2015.

[۱۵] Bifet, Albert, et al. "StreamDM: Advanced Data Mining in Spark Streaming." 2015 IEEE International Conference on Data Mining Workshop (ICDMW). IEEE, 2015.

[۱۶] “Spark Streaming Programming Guide." Spark Streaming. Web. 06 Apr. 2016. <http://spark.apache.org/docs/latest/streaming-programming-guide.html>.

[۱۷]Das, Tathagata, Matei Zaharia, and Patrick Wendell. "Diving into Spark Streaming's Execution Model." Databricks. 2015. Web. 06 Apr. 2016. <https://databricks.com/blog/2015/07/30/diving-into-spark-streamings-execution-model.html>.

[۱۸] “MLlib | Apache Spark." Web. 06 Apr. 2016. <http://spark.apache.org/mllib/>.

[۱۹] “Livy, an Open Source REST Service for Apache Spark" Web. 21 June. 2016. <http://livy.io/>.

با تشکر از توجه شما ☺