International Conference on Applied Internet and Information Technologies, 2018
DOI:10.20544/AIIT2018.P08

Visualization of Big Digital Data with Zeppelin

Snezana Savoska *, Dragan Milevski*, Andrijana Bocevska*
* Faculty of information and communication technologies, Partizanska bb, 7000 Bitola, Macedonia
{snezana.savoska, andrijana.bocevska}@fikt.edu.mk, dragan_055@hotmail.com

Abstract - The Big Digital Data (BDD) concepts deserve attention nowadays because of the need of analysis of data models they use, their usage from the customers with various levels of IT knowledge and the need for different analysis of big data in time and space. The huge amount of data and objects collected from different sources are flowing in databases with heterogeneous data with purpose of efficient visual data analysis. For this purpose, as emerging trend, visualization of big digital data is very important and has to be taken into consideration from methodology and practical aspects. The methodology used for visualization of BDD usually demands usage of a model for big data. The practical aspect demands data science knowledge and specific tools capable to deal with big data for different purposes. Zeppelin software tools for problems of visual data analysis of big digital data are used in this paper.

I. INTRODUCTION

Big Data become emerging technological trend as a part of Data Science. A big data solution is different in all aspects from a traditional business inteligence solution. Companies are deriving significant insights by analyzing big data that gives a combined view of both structured and unstructured customer data as well as seeing increased customer satisfaction, loyalty, and revenue [1]. Big data are not focused only on the unique market, but are combination of technologies for data management that are evolving in time. Big Data provide storing, managing, manipulation with a huge amount of data with a high accuracy and speed in precise time, enabling accurate and valuable information and knowledge. The key for big data understanding is the fact that big data have to be managed to meet the business requirements and design the solution [2]. As companies begin to evaluate new types of big data solutions, they are able to monitor data coming from machine sensors to predict a catastrophic event. Retailers can monitor data in real time to upsell customers' related products as they are executing a transaction. Big data solutions can also be used in healthcare to determine the cause of an illness and provide a physician with guidance on treatment options [3].

Dealing with this problems demands usage of some predefined patterns that are suitable for visualization of big digital data as Mashup View Pattern [2], Compression Pattern, Zoning Pattern, First Glimpse Pattern, Exploder Pattern, Portal Pattern and Service Facilitator Pattern [1]. For these patterns knowledge of the applied model is needed (Fig.1). Commercial tools that appear on the market promise higher productivity for big digital data analysis and provide specific business visualizations. Tools as QlikView [4], TIBCO Spotfire, SAS RA [5], Tableau and Zeppelin [6] can be mentioned. They also can be used in combination for gaining BDD visualization.

Apache Zeppelin [7, 8] is multipurpose software for data ingestion, research, analysis, visualization and collaboration. It supports more than 20 backend systems, including Apache Spark, Apache Flank, Apache Hive, Python, R, and JDBC (Java Database Connectivity) [1, 9]. It is easy to deploy, built on top of modern web technologies (provides built-in Apache Spark integration, eliminates the need to build a special module, plug, or library), incorporates visualizations and dynamic forms [10, 11]. Apache Zeppelin is flexible, allows users to mix different languages, exchange data between backends, as well as customization of appearance [10, 12]. Hardware and software components for interpreter, authentication and visualization can be included. Its advanced features provide interaction between custom visualizations and a group of resources [13].

Apache Zeppelin is open source web based software that performs interactive data analysis with possibilities of data capture, research, sharing, visualization and collaboration with Hadoop and Spark [14]. Zeppelin is a tool that enables engineers, analysts and data scientists to be more effective and productive through development, organizing, sharing and code exchange as well as data visualization, in a huge interactive work processes and projects [15, 16]. There are many available Spark software tools. Zeppelin supports Python, as well as a long list of programming languages such as Shell and Markdown [9, 10]. In fact, Zeppelin supports multiple language backends which has support for a growing ecosystem of data sources. It provides interactive "snippet-at-time" experience for Data scientists. The collaborative data analyses with Zeppelin and data visualization capability make them easy for data manipulation for research, visualization, sharing and collaboration, using Apache Flink [17], Apache Hadoop, and Apache Spark [14] as some of the Big Data platforms. With Apache Zeppelin, a wide range of users can make excellent visualizations that can be used for analysis of collaborative documents with SQL, Scala or other tools. Zeppelin is used in many companies as Amazon Web Services, Hortonworks, JuJu and Twitter [1, 4].

The paper will be organized as follows. First section describes some visualization possibilities using Zeppelin. The second section considers creating visualization with local databases and SQL statements with examples as well as examples for mathematical

function visualization. The practical usage for sharing the gained visualizations on the extended screens and copying the paragraphs with links follows. Concluding remarks highlight the main contribution of the paper and propose some logical conclusions.

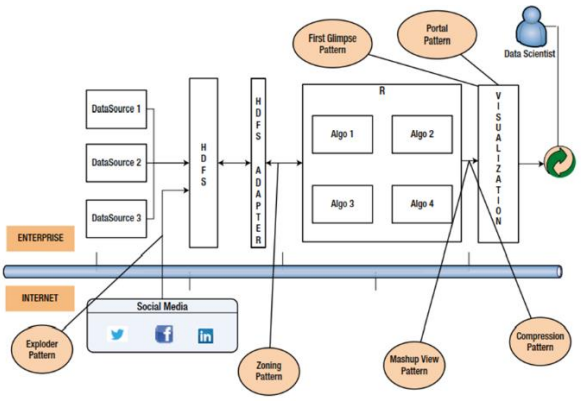

Figure 1. Big data analysis and visualization patterns [3]

II. VISUALIZATION WITH ZEPPELIN

Zeppelin can help to create big digital data (BDD) visualizations analyzing whole data with diagrams from Zeppelin's library [10]. The analyst can have the general data overview, gaining all data from big data repository in the same screen online from some cloud based platform or gain some snippet-at-time saved locally. As an example, we provide visualization of extracted data from big data repository, named cars, transformed as .csv file (Fig.3). For this purpose, we wrote the Scala code shown in Fig.2.

```
      def dataTable():String={
          var str=""
          //Load csv file
          Val
csv=scala.io.Source.fromFile("D:/Downloads/weather/cars.csv")
          for(line <- csv.getLines){
              val cols=line.split(",").map(_.trim)
              str=str+cols.mkString("\t")
              str=str+"\n"
          }
          csv.close
          return str
      }
      //To display the charts
      println("%table "+dataTable())
```

Figure 2. Scala code for creating visualization

1	mpg,cylinders,engine,horsepower,weight,acceleration,year,origin,name
2	18,8,307,130,3504,12,70,American,chevrolet chevelle malibu
3	15,8,350,165,3693,11.5,70,American,buick skylark 320
4	18,8,318,150,3436,11,70,American,plymouth satellite
5	16,8,304,150,3433,12,70,American,amc rebel sst
6	17,8,302,140,3449,10.5,70,American,ford torino
7	15,8,429,198,4341,10,70,American,ford galaxie 500
8	14,8,454,220,4354,9,70,American,chevrolet impala
9	14,8,440,215,4312,8.5,70,American,plymouth fury iii
10	14,8,455,225,4425,10,70,American,pontiac catalina
11	15,8,390,190,3850,8.5,70,American,amc ambassador dpl
12	0,8,350,165,4142,11.5,70,American,chevrolet chevelle concours (sw)
13	0,8,351,153,4034,11,70,American,ford torino (sw)
14	0,8,383,175,4166,10.5,70,American,plymouth satellite (sw)
15	0,8,360,175,3850,11,70,American,amc rebel sst (sw)
16	15,8,383,170,3563,10,70,American,dodge challenger se
17	14,8,340,160,3609,8,70,American,plymouth cuda 340
18	0,8,302,140,3353,8,70,American,ford mustang boss 302
19	15,8,400,150,3761,9.5,70,American,chevrolet monte carlo
20	14,8,455,225,3086,10,70,American,buick estate wagon (sw)
21	22,6,198,95,2833,15.5,70,American,plymouth duster
22	18,6,199,97,2774,15.5,70,American,amc hornet
23	21,6,200,85,2587,16,70,American,ford maverick

cars

Figure 3. Cars.csv database shown in Excel

Using the code from Fig.2, the database is loaded in .csv file, the table with the source of Zeppelin is created and then the data are visualized. The main comparative advantage of Zeppelin is the ability of direct instantaneous visualizations results of data with all data files from database. With a simple drag and drop of data that have to be visualized, Zeppelin aggregates all data values and shows in a graphical way, using red color for files and green for data that have to be visualized. The three areas where data can be dragged and dropped are Keys, Groups and Values. In the Key area, the key files that have to be visualized are entered. The area Groups contains the files that have to be grouped according to some criteria. The last area, Values, contains the files that have to be aggregated, necessary to create sum, count, average, min, max or other functions.

First visualization shows the diagram of year of production of cars compared with each car's acceleration (0 to 100 km/h), grouped according to production country (origin by America, EU, Japan) – Fig.4. In this visualization, some results are obvious, as the year of production of cars, shown on the X axis. Y axis shows the acceleration rate in seconds (from 0 to 100 km/h) of the fastest accelerating car produced in the given years. All cars are grouped according the Origin shown on the upper right corner in Fig. 4.

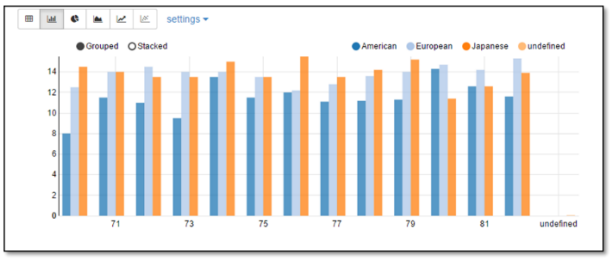

Figure 4. Visualization of the year of production related to minimal acceleration grouped by cars production origin

Fig.5 presents magnification of the Fig.4 gained with selection of the desired data column. We can see that the fastest accelerating car is produced in America in 1970 and can achieve acceleration from 0 to 100 km/h in 8 seconds.

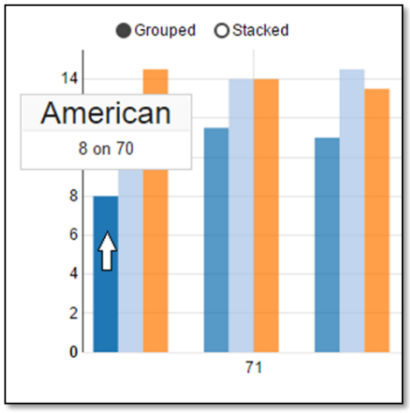

Figure 5. Details about selection in Fig.4

The second visualization shows diagram of cars' motor power in relation with minimal acceleration from

0 to 100 km/h, grouped by year of cars' production. X axis shows engine volume of old-timers in cubic inch. Y axis presents summed acceleration for the fastest cars and year of car's production (Fig.6).

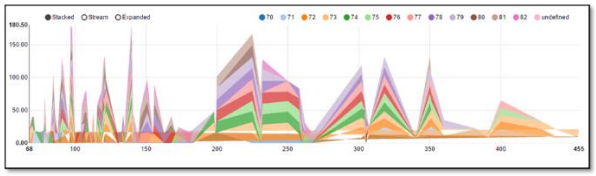

Figure 6. Visualization of the motor power related to min acceleration and year of production of the cars

Detail overview of all cars from database cars.csv with motor volume bigger than 250 cubic inches or 3.2 liters is shown on Fig.7. To obtain the details we can point with the mouse over the year of production for all years in the database. Right of the years of production (70, 71 and 72) the values for acceleration in seconds are shown, grouped by year of production.

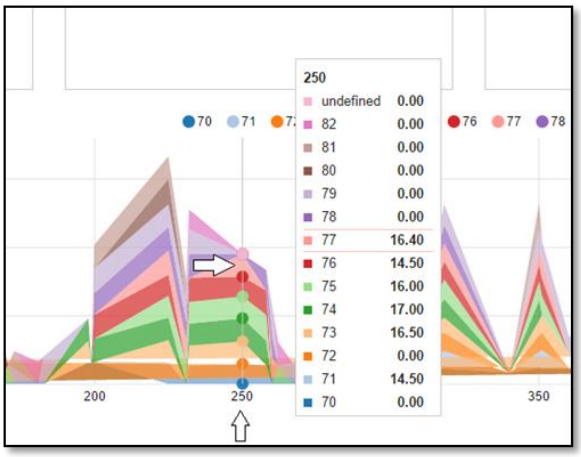

Figure 7. Detail overview of cars from cars.csv database

Zeppelin has an option to include or exclude only the data that needs to be visualized by simply clicking the buttons in the data grouping section. The example shown on Fig.8 will present only the data from 74, 78, 79, 80 and 82 (Fig.9).

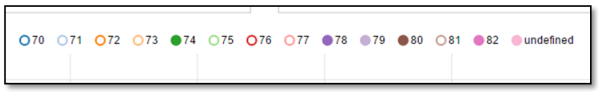

Figure 8. Controlling the grouped data that have to be shown

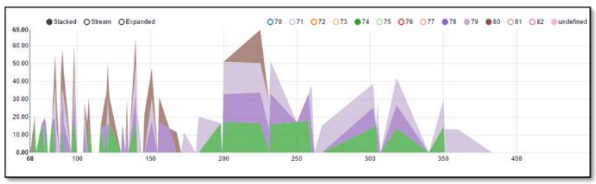

Figure 9. Filtered data for year of production - selected in Fig.8

Visualization on Fig.10 in which the number of cylinders, the number of horsepower and motor volume, is related with the cars' weight aims to show how we can put more data in the Keys, Groups and Values areas.

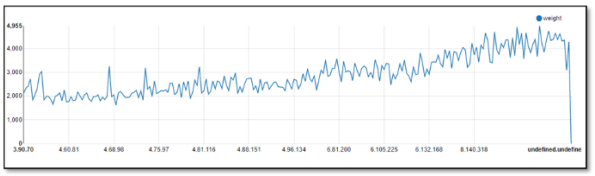

Figure 10 Visualization of numbers of cylinders, horsepower and motor volume related with cars' weight

Fig. 11 shows an example of detailed overview of all cars in cars.csv. X axis represents the three values (separated with commas): numbers of cylinders, horsepower and motor volume, sorted by the number of cylinders for each car. Y axis shows the car's weight. Pointing the value, we can see toolip (Number of cylinders is 4, horsepower 70 hp, motor volume 79 cubic inches and car's weight 2074 lbs or 940 kg).

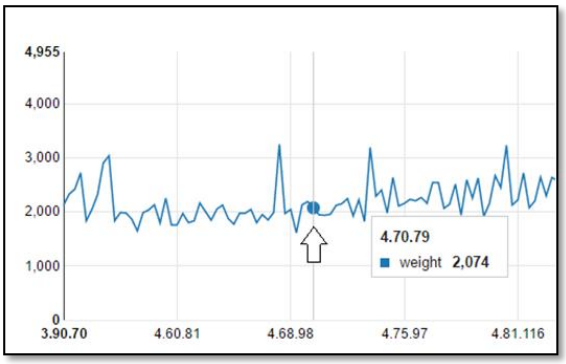

Figure 11. Detail overview of all cars with tooltip for the selected cars

III. Creating Visualization on Local Databases with SQL Statements

In order to visualize data with Zeppelin, extracted from a huge big data set and store in local database, containing 45 212 data rows, with a huge amount of columns with variate data is used, named bank.csv. It is also snippet-at-time of big data repository. The data have to be transformed from CSV file to RDD (Resilient Distributed Datasets that is the main Spark structure) and for that reason it is necessary to create the script file, shown on Fig.12. When the transformation is done, header will be removed from file by means of filter option.

```
val bankText = sc.textFile("D:/Downloads/bank/bank-full.csv")

case class Bank(age:Integer, job:String, marital : String, education : String,
balance : Integer)

// split each line, filter out header (starts with "age"), and map it into Bank
case class
val bank = bankText.map(s=>s.split(";")).filter(s=>s(0)!="\"age\"").map(
    s=>Bank(s(0).toInt,
            s(1).replaceAll("\"", ""),
            s(2).replaceAll("\"", ""),
            s(3).replaceAll("\"", ""),
            s(5).replace All("\"", "").toInt
        )
)

// convert to DataFrame and create temporal table
bank.toDF().registerTempTable("bank")
```

Figure 12. Script file - create RDD from bank.csv

After loading the database, it is converted into a table which can be used to perform certain analyzes and visualizations by writing SQL commands. In this example, they are used to filter the column "age", to display values that are less than 30. (Fig. 13).

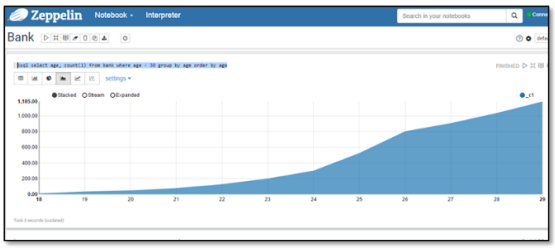

Figure 13. Area chart with condition "age" column < 30

If the data that have to be analyzed and visualized are variable and it is necessary to obtain different visualizations depending on the values, a field, in which the user inputs a value himself is used by implementing SQL code. The values that are entered in the field are executed instantaneously, so that immediately new visualization is obtained in relation to the entered value in the field. This is one of the main advantages of Zeppelin compared with other big data analysis tools. There is also possibility to index data from different columns and make new insight with some optimization tools.

Mathematical data visualization with Zeppelin can be created using Scala programing language. Fig.14 shows example of visualization for the function y=sin(x), using Scala code. First X and Y axes are created, then the cycle from 0 to 360 degree is run and finally sin(x) is calculated. Visualization of data is shown on Fig.15.

```
println("%table\nx\ty")
(1 to 360).map(i=>i.toDouble / 50).map(x=>(x,
Math.sin(x))).foreach{case (x,y) => println(x + "\t" + y)}
```

Figure 14. Scala code for y=sin(x) function visualization

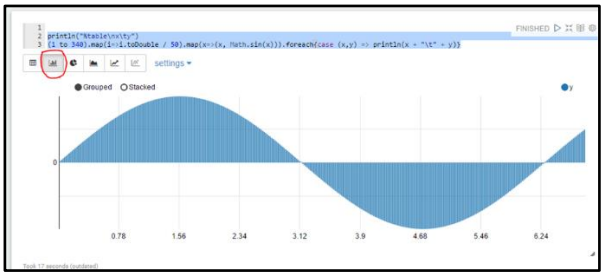

Figure 15. Mathematical visualization of sin(x) using Zeppelin

IV. Practical Usage of Sharing the Gained Visualization on the Extended Screens and Copying the Paragraphs with Links

Apache Zeppelin has the capability to share the working screen of data visualization with web socket or URL address. Visualizations can be changed interactively enabling all working addresses instantly to see the changed data. To share the screen the teams and users working on Zeppelin, just have to copy the URL address that is in the upper part of the browser (Fig. 16).

Figure 16. URL address shared with Zeppelin

The option for screen sharing is shown on Fig.17. The advantage of this option provides the easiest access to codes and visualizations on the distance locations between teams and users online.

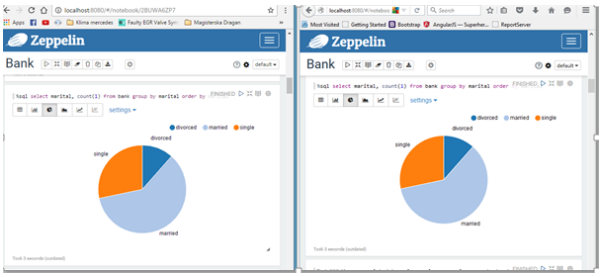

Figure 17. Sharing screen with Zeppelin

Zeppelin also gives the possibility to copy paragraphs with links. In this case, the link from the paragraph is copied and the result is published in a new window in the browser, as shown in Fig.18. If that paragraph needs to be opened in another browser, it is necessary to copy the URL and use it to access.

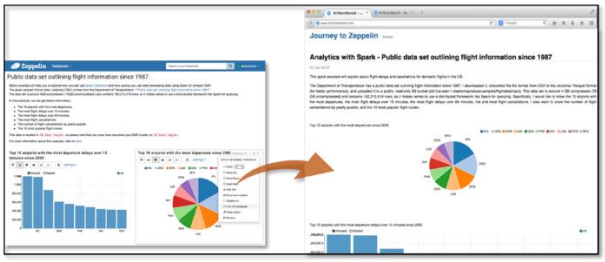

Figure 18. Copying the paragraph in visualization

As conclusion, Apache Zeppelin has many advantages and possibilities which simplify the team work, offering a wide range of programming languages with embedded possibilities using Interpreter menu. It can work online on cloud and with databases extracted from big digital data unlike some other tools. Also variable database formats with a wide range of visualization possibilities, enabling easy data analysis, taking into consideration the huge amount of data that can be supported. The SQL statements which Zeppelin has in disposal can help in decision-making processes with easy sorting, extracting and data filtering.

V. Concluding Remarks

The paper considers patterns and concept for dialing with BDD visualization. Some general problems with BDD visualization are considered in order to highlight some patterns and methods for BDD visualization [18]. The huge amount of unstructured, variable data and their dynamic nature bring unique challenges for data scientists to create specific valuable and imaginative visualizations online or prepared locally [1]. For this reason, the paper considers software solution for solving some of these challenges about BDD. Some basic concepts help to create the best data visualizations and practices offline. But, first of all, the data that need to be visualized have to be understood, regarding their volume, variety, variability and velocity [18]. Next aspect is how the data have to be presented and what information they contain [19]. The knowledge of the audience is very important in order to have a clear

understanding what they want from the data and in what form [20, 21].

For effective BDD visualization, Zeppelin is used as an open-source, web-based tool that enables interactive data analytics and collaborative documents. It allows making meaningful, data-driven, interactive documents with SQL, Scala, R, or Python right in the browser. Zeppelin Interpreter is the plug-in which enables users to use a specific language/data-processing-backend [12]. Currently Zeppelin supports many interpreters such as Scala (with Apache Spark), Python (with Apache Spark), SparkSQL, Hive, Markdown and Shell. Interactive interface allows users to instantly see the results of the analytics and have an immediate connection with their creation.

Apache Zeppelin is used by organizations as Amazon Web Services, Hortonworks, JuJu and Twitter. They analyze their BDD with some advanced visualization methods in Zeppelin, gaining more insights in data and making decisions with diagrams created with Zeppelin.

In this paper our practical experience with Zeppelin was related to gaining visual BDD analysis, starting from extracting data from big digital data repository, data preparation and data visualization. With the created visualizations, we can highlight that the whole BDD concept of data analysis is based on visualization techniques and models, with software tools which bring competitive advantages to users and companies. For online usage, they demand a huge hardware resources, high transmission speed and time.

REFERENCES

[1] N. Sawant and H. Shah, Big data application architecture Q&A, 2013, pp. 13,14,19, 79-89, 123-125.

[2] E. Stubbs Big Data, Big Innovation: Enabling Competitive Differentiation through Business Analytics, Copyright © 2014, SAS Institute Inc., Cary, North Carolina, USA. A

[3] J. Hurwitz, A. Nugent, Dr. F. Halper and M. Kaufman, Big data for dummies, 2013, pp. 39-44, 45-47.

[4] Gartner, Inc. and/or its Affiliates, 2016

https://www.gartner.com/en/about.

[5] S. Grover, Big Digital Data, Analytic Visualization and the Opportunity of Digital Intelligence, SAS Institute Inc., Washington D.C, 2014.

[6] Zeppelin overview,

https://www.youtube.com/watch?v=_PQbVH_aO5E.

[7] Explore Apache Zeppelin UI, http://zeppelin.apache.org/docs/latest/quickstart/explorezeppelinui.html#explore-apache-zeppelin-ui.

[8] ZeppelinHub Viewer, https://www.zeppelinhub.com/viewer/notebooks/aHR0cHM6Ly9yYXcuZ2l0aHVidXNlcmNvbnRlbnQuY29tL0xlZW1vb25zb28vemVwcGVsaW4tZXhhbXBsZXMvbWFzdGVyLzJCMlhLRkNETS9ub3RlLmpzb24.

[9] Publish paragraph, https://zeppelin.apache.org/docs/latest/manual/publish.html.

[10] Apache Zeppelin, http://hortonworks.com/apache/zeppelin/.

[11] Apache Zeppelin: Big data prototyping and visualization in no-time, https://dataminded.be/blog/apache-zeppelin-big-data prototyping-and-visualization-no-time.

[12] Interpreters in Apache Zeppelin, https://zeppelin.apache.org/docs/latest/manual/interpreters.html.

[13] S. Mohanty, M. Jagadeesh, H. Srivatsa, Big Data Imperatives, 2013, pp. 142-144, 271-273.

[14] Sparklet User Guide, http://mund-consulting.com/Products/Sparklet-User-Guide.pdf.

[15] Honeywell Users Group Americas, Driving Digital Intelligence through Unified Data, Analytics, and Visualization, 2015.

[16] D. Adkison, IBM Cognos Business Intelligence, 2013.

[17] Apache Flink, https://flink.apache.org/.

[18] S. Murray, Interactive Data Visualization for the Web, 2013.

[19] Michael Colman, https://www.digitalthing.com.au/how-big-data-is-shaping-online-marketing-in-2015/.

[20] John Wiley & Sons, Inc. Actionable Intelligence, A Guide to Delivering Business Results with Big Data Fast, 2014.

[21] IBM Bluemix Data & Analytics, https://console.ng.bluemix.net/