Big Data Processing for Full-Text Search and Visualization with Elasticsearch

Aleksei Voit, Aleksei Stankus, Shamil Magomedov, Irina Ivanova
Moscow Technological University,
Moscow, Russia

***Abstract*—In this paper, the task of using Big Data to identify specific individuals on the indirect grounds of their interaction with information resources is considered. Possible sources of Big Data and problems related to its processing are analyzed. Existing means of data clustering are considered. Available software for full-text search and data visualization is analyzed, and a system based on Elasticsearch engine and MapReduce model is proposed for the solution of user verification problem.**

Keywords—Big Data processing; verification; elasticsearch; MapReduce; data clustering

I. Introduction

Thanks to technological advancement, Big Data has become available in various scientific and technological fields, including social sciences and management. Based on the Big Data, it is possible to carry out research and analysis to identify human intentions, feelings and thoughts that will allow us to identify not only individuals but also the intentions of communities and society as a whole [1]. However, the analysis of Big Data differs from traditional methods of generalization.

One of the main issues arising in the analysis of Big Data is the gap between the object of observation and the object of analysis. For example, if the objects of observation are user accounts, it is not always obvious whom each account represents. The account can also be used by family members, friends or outsiders. When analyzing Big Data some assumptions are usually made about the nature of the object of observation, which are often violated in practice. Verstrepen and Goethals [2] consider the challenges of recommendation system applied to shared user accounts. Another problem that leads to the difference between the object of observation and the object of analysis is the trade-off between information and confidentiality. Since confidentiality restricts access to data at the individual level, the analysis is carried out based on indirect data.

Another important issue in analyzing Big Data is a possibly false conclusion that an object of observation can be considered an average representative of a wider population than a sample actually covered. For example, you can get a false conclusion if you are analyzing data from online sources in countries where less than half of the population has access to the Internet.

An important issue that arises with collecting publicly available data from company websites or aggregators is generalization. There is not always information about which groups of people the available Big Data represents, how it was sampled, whether it was pre-processed, etc. Such data is also less likely to include detailed demographic information for confidentiality reasons. Consequently, the inference from the analysis of Big Data to wider groups than those from which this data was obtained is uncertain. Moreover, the very possibility of such an inference is called into question. For example, while Twitter is a popular source of Big Data among researchers, even if we can view Twitter data as a random sample of a larger set of tweets, Twitter users are different from the average representative of the population as a whole - they tend to be younger and have a specialist or higher education.

Generalization is also a serious problem when Big Data is obtained using web scraping. Although scraping provides more control over the collection process, there are many unknowns about the relationship between the information available on the website and information that the website owner does not provide. In addition, server problems, network load, website update policies, poor web page design, and the non-random nature of search results are also just some of the factors that lead to sampling errors when collecting Big Data (these and other issues are discussed in the paper by Jank and Shmueli [3]).

Bender [4], Hauge et al. [5] demonstrate the possibility of using Big Data to identify a specific person from indirect data publicly available in the Internet, while Narayan and Shmatikov [6] showed the possibility to identify apparent political and personal preferences. Extracting information from Big Data allows us to establish causal links that include the concepts of internal validity (the ability to draw a causal conclusion from the data), external validity (the ability to generalize the influence to other contexts) and statistical generalization.

The paper considers Big Data sources, problems of Big Data analysis, existing software for Big Data processing. A system for full-text search and visualization is proposed, which is aimed to solve the problem of anonymous user verification.

II. Big Data Sources

At present various sources are available for obtaining Big Data:

- Data from large companies which allow access to their storage through the means of direct download or specialized API's (Netflix, AOL, Twitter, Amazon, eBay and others).

- Open data by government agencies and organizations (traffic accidents, crimes, health surveys, etc.), which provide good coverage but are often not easily accessible.
- Websites that aggregate individual data sets from disparate sources (UCI Machine Learning Repositoryx and others), as well as data mining contest platforms (Kaggle.com, crowdanalytix.com and others). Such data is commonly used for research, machine learning and testing of new algorithms.
- Web scraping — methodical data collection from websites using automated programs. Some websites disallow web scraping by setting technological barriers and legal notices. But many websites do tolerate web scraping if it does not overload their servers.

Considering the above-mentioned issues, one of the most promising ways to obtain Big Data on the behavior and activity of people on the territory of the Russian Federation is the collection and analysis of mobile traffic data and user interaction with mobile devices.

According to statistics for the year 2016 [7], 84 million people aged 16 and over are Internet users in Russia, accounting for more than 70% of the country's population in this age group (Fig. 1).

At the same time, almost half of users access the Internet from their mobile phones and tablets, and this share is steadily growing (Fig. 2).

It should be noted that in the age group of 16 to 55 years, which includes the most active segments of the population, the share of Internet users exceeds 80%, more than half of them access the Internet from mobile devices, and in the age group from 16 to 30 years, the share of mobile device users exceeds 75% (Fig. 3). At the same time, there is a steady increase in the number of Internet users and mobile device owners among the population over 55 y.o.

It should be also noted that currently active Internet users are no longer concentrated in large cities - about 2/3 of the population of small towns and villages in Russia have access to the Internet, and Internet coverage continues to increase steadily. At the same time, the relative share of mobile device users in small cities is often even higher than that for larger cities (Fig. 4).

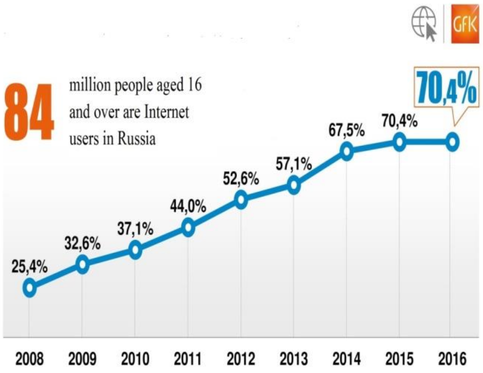

Fig. 1. Internet coverage in Russia (Source: Omnibus GfK, 2016).

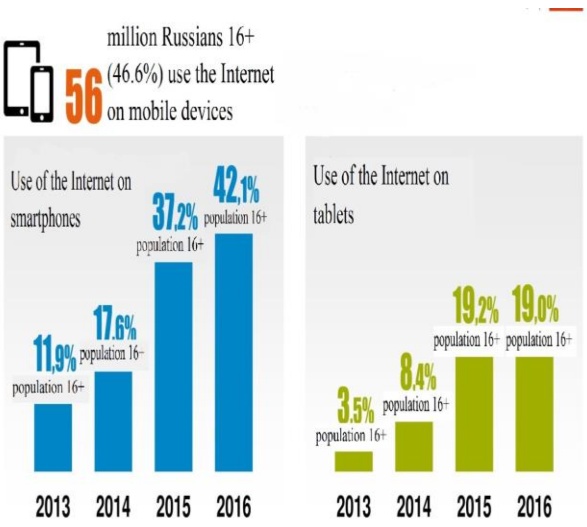

Fig. 2. Internet usage on mobile devices (*Source: Omnibus GfK, 2016*).

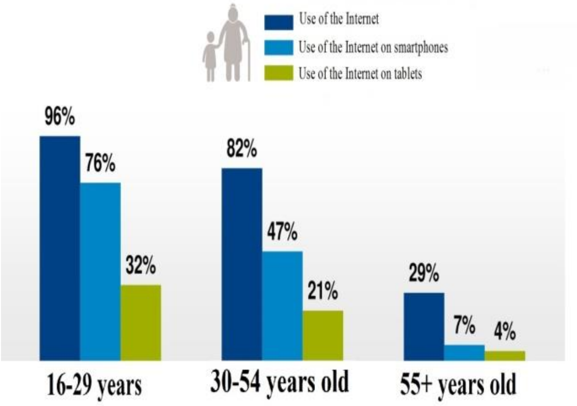

Fig. 3. Profile of Internet users in Russia (Source: Omnibus GfK, 2016).

Thus it can be argued that data on the usage of various information resources in the Internet, including mobile devices and software installed on them, is sufficiently representative for carrying out studies on behavior models using the technologies of Big Data analysis.

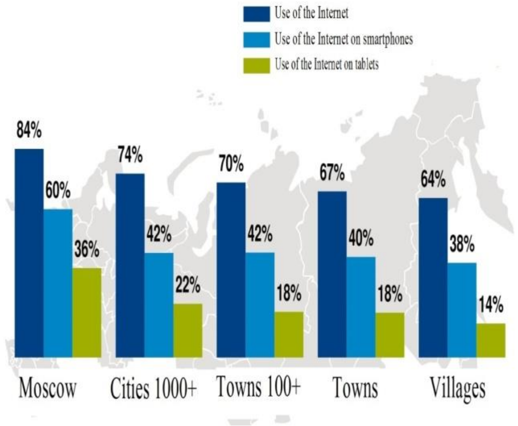

Fig. 4. Geography of the Internet in Russia (Source: Omnibus GfK, 2016).

The above-mentioned data on the usage of information resources includes the content that is created by users through information technologies by interaction with various social media platforms. This category includes data on the digital representation of the user, technological data associated with the digital interaction process, and digital relationships.

By downloading software to the mobile device or using integrated services, the user gives his consent to the processing of his personal data by agreeing to the data use policy. Accordingly, some account information can be accessed from the user's mobile device without identifying it, in particular:

- nickname,
- gender,
- age,
- place of residence,
- education,
- etc.

Technical data about the device and its usage, in particular:

- model and technical characteristics of the mobile device,
- data about installed and deleted applications,
- device location data,
- date and time,
- Internet access data,
- etc.

Data about digital relationships of the user, in particular:

- membership of a particular social group,
- social activity,
- types of created and consumed digital content,
- involvement in marketing programs of various services,
- etc.

III. Problems of Big Data Analysis

After determining the source of Big Data it is necessary to point out a number of problems connected directly to its analysis. The most common methods of statistical analysis for determining and quantifying causality from experimental data are analysis of variance and regression models. Regression models are also extremely popular in observational studies that test causal hypotheses. Nevertheless, approaches to behavioral conclusions that are effective in small samples face problems in analyzing Big Data. The advantage of Big Data lies in its richness in terms of diversity: it is more likely to contain information on rare minorities than small samples or lower dimensional data. However, when applying these statistical models to Big Data, rare minorities are either filtered out (for example, they are considered to be emissions), or their influence is leveled by averaging with the majority. For example, for very large samples, the impact of small minorities and emissions on regression coefficients and statistical tests is very small.

Another problem related to aggregation and heterogeneity is the Simpson’s paradox: the phenomenon in statistics, where in the presence of two groups of data, in each of which there is an identically directed dependence, the direction of the dependence is reversed when the groups are combined. It is important to determine whether the Simpson’s paradox is manifested in the data set used to make decisions. Given the size and diversity of Big Data, the probability of the Simpson’s paradox arises when analyzing is significantly higher than when working with small samples. Shmueli and Yahav [8] introduced a method that uses classification and regression trees for automated detection of potential Simpson’s paradoxes in data sets with few or many potentially confounding variables, which scales to large samples.

If the researcher is interested in analyzing the behavior of subgroups or individuals in addition to the average estimate, then the approaches of predictive modeling and validation can be useful. For example, a causal statistical model can be used to generate predictions for a limited set of observations. Then the predictions and their errors can be compared in different subgroups or sorted to identify subgroups for which the effects differ significantly from the average majority. Examples of the usage of predictive testing to improve causal studies are given in [9]-[12].

Thus researchers should understand that classical statistical causal modeling and inference, based on experimental data or observational data, are aimed at revealing the general causal effect within the analyzed sample. Therefore, it is necessary to interpret the results obtained with the help of these methods with care, so as not to mistake the conclusions received for the sample as a whole as applicable at the individual level.

Another problem arising when analyzing Big Data is that of the utility of statistical significance and inference due to large samples and multiple testing. In particular, in very large samples even minor effects are statistically significant [13] and therefore testing hypotheses using P-value is difficult. Alternatives to the usage of P-values include such methods as using estimation in place of testing or adopting a Bayesian approach [14], [15].

The problem of multiple testing is summarized in the works of Agarwal and Chen [16], devoted to the development of algorithms for computational advertising and content recommendations. It is based on the multivariate nature of outcomes in a variety of different contexts with multiple objectives. Such multiplicity, when using statistical inference, leads to the testing of many hypotheses that run the risk of false conclusions. For example, when testing multiple independent hypotheses, if each hypothesis is checked at a given significance level, then the probability of a false conclusion at least on one of the tests increases exponentially with respect to their number.

With respect to the analysis of Big Data, which are usually heterogeneous and of high dimensionality, the above problems force us to seek a compromise between post-testing effects for different subgroups to identify heterogeneous effects (for example, by gender, age, place of residence and other variables and their combinations) and the risk of false conclusions due to multiple testing.

Taking the described problems into account, it can be concluded that to perform the verification of individuals based on analysis of Big Data on their interaction with various information resources, the task of optimal segmentation (clustering) of the analyzed data is of primary importance.

IV. Research

The clustering procedure is aimed at dividing the data into groups of similar objects in accordance with the criterion of maximizing the similarity between objects in the same group and minimizing the similarity between objects in different groups [17]. With a continuous increase in the amount of data, traditional clustering methods have reached their limits, which have led to the development of methods for parallel clustering.

The most well-known model for Big Data processing is MapReduce. This model was proposed by Dean and Ghemawat [18] at Google, where it was successfully used for various purposes. The strengths of this model correlate to the fact that it allows automatic parallelism and distribution. In addition to a fault-tolerant mechanism that helps to overcome failures, it also provides tools for state management, monitoring and load balancing. Optimization of data distribution is provided by storing them on local disks to avoid excessive consumption of network bandwidth.

Cluster analysis with MapReduce consists of two steps: “Map” and “Reduce”. At the “Map” step data is filtered and sorted, while at the “Reduce” step the results of the previous step are summarized. The Map function takes records from the input files as key-value pairs and creates intermediate key-value pairs. The Reduce function works with the values of a certain intermediate key and produces one final value for the same key.

There are several software implementations of the MapReduce model. The most popular framework is Hadoop, implemented in the Java language. Developed by the Apache Software Foundation, this project includes a set of open source modules that enables reliable and scalable distributed computing. Thanks to Hadoop’s features, such as its organized architecture, scalability, cost-effectiveness, flexibility and resilience, the Hadoop MapReduce framework is the most preferred platform for solving the problem in question.

Some of the most common clustering algorithms based on MapReduce are the following:

PKMeans [19] is a MapReduce-based implementation of the k-means algorithm. It is designed with a single MapReduce job, in which the Map function is responsible for the assignment of each sample to the nearest center, and the Reduce function is responsible for updating the new centers.

MR-DBSCAN [20] is a MapReduce-based implementation of the well-known DBSCAN algorithm. Its parallel method consists of four steps. In the first step, the size and the general spatial distribution of all the records are summarized, and then a list of dimensional indices indicating an approximate grid partitioning is generated for the next step. The second step performs the main DBSCAN process for each subspace divided by the partition profile. The third step handles the cross border issues when merging the subspaces. At the end, a cluster ID mapping, from local clusters to global one, is built for the entire data set based on pairs lists collected from the previous step. Finally, the local ID’s are changed by the global ones for points from all partitions in order to produce a unified output.

DBCURE-MR [21] is the parallel version of a new density-based clustering algorithm, called DBCURE, which is implemented using the MapReduce programming model. DBCURE acts similar to DBSCAN by reiterating two steps. In the first step an unvisited point in the data set is selected, which is considered a seed and is inserted into the seed set. In the second step, all points that are density-reachable from the seed set are retrieved. This process produces clusters one at a time and stops when the seed set becomes empty, contrary to its parallel version, which finds several clusters at the same time by treating each core point in parallel through four steps. The first step is responsible for the estimation of the neighborhood covariance matrices and it is performed using two MapReduce algorithms. The second step performs the computation of ellipsoidal τ-neighborhoods and it is performed using two other MapReduce algorithms. The third step discovers core clusters, which is done by a single MapReduce algorithm. Finally, the last step is responsible for the merge of core clusters and it is performed with a single MapReduce algorithm.

PMR-Transitive [22] is a new parallel heuristic based on the MapReduce programming model of a recently appeared method, namely, Transitive heuristic [23]. In this heuristic, clusters are obtained by partitioning categorical large data sets according to the relational analysis approach. The relational analysis approach provides a mathematical formalism where the problem of clustering takes the form of a linear program with n^2 integer attributes (with n being the number of instances). Heuristics are the most convenient solution to produce satisfactory clustering results in the fastest time, particularly in the context of Big Data, where the number of instances is large and the response time is a critical factor. Since the original heuristic is sequential, it needs to be adjusted to the MapReduce model. This paper provides a detailed description of the new design based on the key methods of the MapReduce model, namely, Map and Reduce. And advantageously, most steps which produce high computational costs involved in Transitive heuristic can be processed in parallel.

The task of user verification is solved in two stages. Segmentation (clustering) allows grouping the indirect data about the behavior of unauthorized users on the network in such a way that each group (segment or cluster) corresponds to a specific individual. After that, the data in these segments can be searched, retrieved on demand, analyzed and visualized. Special tools are used for these tasks; most common of them are the following.

Sphinx is a full-text search engine with a distinctive feature of high indexing and searching speed, as well as integration with existing database management systems (MySQL, PostgreSQL) and API for common web programming languages (officially supports PHP, Python, Java; there are community-implemented API's for Perl, Ruby, .NET, and C++). Supports advanced search capabilities, including ranking and stemming for Russian and English languages, distributed search and clustering support. For large volumes of data the Delta index scheme can be used to speed up indexing. In addition, Sphinx supports Real Time indexes, filtering and sorting of search results and searching for wildcard conditions.

Apache Solr is an extensible search engine for full-text search with open source, based on the Apache Lucene project. Its peculiarity is that it is not just a technical solution for searching, but a platform which can easily be expanded, changed and customized for various needs — from the usual full-text search on a website to a distributed system for storing, receiving and analyzing text and other data with a powerful query language. Unlike Sphinx, documents are saved entirely and do not need to be duplicated in the database. Main features of Solr — full-text search, highlighting of results, facet search, dynamic clustering, integration with databases, processing of documents with complex format (for example, Word, PDF). Since Solr has the capability of distributed search and replication it is highly scalable.

Xapian is a search engine library. Packages are available for Ubuntu and Red Hat, can be compiled for OSX, and can also run under Windows via CygWin. Xapian is less common and flexible than the above mentioned search engines. It has no morphology, but there is stemming for a number of languages (including Russian). Other implemented features include spell check in search queries, incremental index, updated in parallel with the search, operating with several indexes and in-memory indexes for small databases.

Elasticsearch was initially developed as a system for full-text search in large volumes of unstructured data. At present, Elasticsearch is a full-fledged analytical system with various capabilities. Data in Elasticsearch is stored in an inverted index format based on Apache Lucene. Apache Lucene is the most famous search engine, originally focused specifically on embedding in other programs. Lucene is a library for high-speed full-text search, written in Java. It provides advanced search capabilities, a good index building and storage system that can simultaneously add, delete documents and perform optimization along with the search, as well as parallel search on a set of indexes combining the results. The downside is comparatively low indexing speed (especially in comparison with Sphinx), as well as lack of API (which is taken care of by Elasticsearch).

Elasticsearch allows dividing the data between several machines, which makes it possible to support high-performance operations. The parts between which data is divided are called shards. Shards come in two types — master and replica. The master allows both read and write operations, while the replica is read only, and is an exact copy of the master. Such a structure ensures the stability of the system, since in the event of a master failure, the replica becomes a master. Because the replicas are exact copies of the master, different queries can be processed at the same time from both the master and the replica. Thus, customer requests for the index are executed in parallel on all shards, after which the results of each shard are collected and sent back to the client. This greatly increases system performance.

There are many other libraries for full-text search, such as MySQL fulltext, PostgreSQL Textsearch, CLucene, Lucene++ and others, but most of them are applicable only in systems with a specific database or programming language and are not suitable for general solution of the task in question.

Comparative analysis of the above mentioned search engines [24] are presented in Table I.

Sphinx provides a very fast search and indexing, but is slow to update due to the fact that there is no mechanism to automatically update the index. A significant disadvantage is that it only works with MySql and Postgres. It is not suitable for the solution of the task in question, because it can not update or delete documents in the index (only the addition works).

Apache Solr provides very high indexing and searching speed, its index size is one of the smallest, and it has high extensibility. It can also act as a repository. Solr includes many additional functions, such as inaccurate search and the ability to scale out of the box. The downside is that it is a Java-server in a servlet container, implemented as a web service with XML / JSON / CSV interfaces.

Elasticsearch (based on Apache Lucene) has slightly lower indexing and searching speed compared to Sphinx, but it offers not only search and storage, but also contains other tools (visualization, log collector, encryption system, etc.). It is able to scale and enables sampling of very complex shapes, which makes it a good choice for the analytical platform. This engine is not the easiest to use, but it contains a lot of extra features. The big advantage is that this engine uses very little memory, and incremental indexing is as fast as indexing multiple documents at once.

TABLE I. COMPARATIVE ANALYSIS OF SEARCH ENGINES

	Sphinx	**Solr**	**Elasticsearch**	**Xapian**
Indexing speed (Mb/s)	4.5	2.75	3.8	1.36
Search speed (ms)	7/75	25/212	10/212	14/135
Index size (%)	30	20	20	200
Realization	Server	Server	Library	Library
Interface	API, SQL	Web-service	API	API
Search operators	Boolean, prefix search, exact phrase, words near, ranges, word order, zones	Boolean, prefix search (+ wildcards), exact phrase, words near, ranges, approximate search	Boolean, prefix search (+ wildcards), exact phrase, words near, ranges, approximate search	Boolean, prefix search, exact phrase, words near, ranges, approximate search

Xapian provides relatively fast searching, but significantly slower indexing. It also has a very large index size. As an advantage it can also be highlighted that it has many interfaces for different languages (C ++, Java, Perl, Python, PHP, Tcl, C #, Ruby, Lua). Nevertheless, for the task in question Xapian is completely inappropriate due to the large amount of data and frequent indexing.

Thus, the search engine and the full-text search system Elasticsearch is best suited for the task of search and visualization in large sets of clustered data corresponding to users' interaction with various information resources.

V. Full-Text Search and Visualization System

The proposed system for full-text search and visualization has the following composition:

- Software module responsible for receiving collected data, integration of third-party services and clients and analyzing the data received.
- Software module responsible for structuring of received data, further data processing and preparation for upload or visual display.
- Software module responsible for storage of the received information in the database.
- Software responsible for data backup, which is a part of the data storage tool — the non-relational database Elasticsearch [25]. All collected data are stored in a failproof cluster, where data storage and accessibility are provided by the built-in Elasticsearch mechanisms. The Elasticsearch system also provides data scaling. When a new Elasticsearch database server appears, its internal mechanisms ensure the organization of data storage on it automatically.
- Software module responsible for transmission of data collected from integrated third-party services and clients. Encryption of the transmitted data is carried out using the standard SSL Internet security technology. Encryption on the user's side is performed by the browser and data are transmitted via the https protocol. Data encryption on the system side is performed using the system's web server [26], [27].
- Auxiliary software modules — the transaction log module, the data protection module and load balancers. The transaction log is maintained using the system log, as well as logging of operations performed in each module by means of the module itself. The load balancing for the data is realized by means of Elasticsearch database. Balancing the flow of input requests is carried out by increasing the number of servers in the front-end layer (the number of web servers used). Each web server has its own ip-address. Whenever a website is accessed by its name, it is assigned to the next server of the front-end layer, according to ip-address order. This way the load balancing is performed, while also automatically maintaining the working capacity of the front-end server layer in case one of them suffers a failure.

The full-text search and visualization system is developed using the mvc model (models, controllers and views):

- Model class describes the access to the data necessary for the operation of the application website pages.
- Controller class describes the logic for the management of the application website pages.
- View class describes the user interface.

The main advantage of using the mvc model is the freedom of combining its components. Each part of the application can be changed independently of other software modules.

Input data for the full-text search and visualization system include:

- Data received from Elasticsearch database — in JSON format.
- Data from the service database with information about users and workplaces — in SQL format.

Output data of the system include:

- Data for visual display — in HTML format.
- Data for printing — in csv format.

The data are stored in the non-relational Elasticsearch database, the main purpose of which is to provide fast search in large data sets.

The internal representation of the Elasticsearch data used by the system has the following hierarchical structure:

- Object index.
- Document type.
- Set of data values.

Service data associated with user management and user workstations are stored in an auxiliary MySQL database.

Hardware of the proposed full-text search and visualization system includes a set of servers (front-end and back-end server layers), workstations of administrators and system developers and peripheral equipment, including external backup and archive drives (Fig. 5).

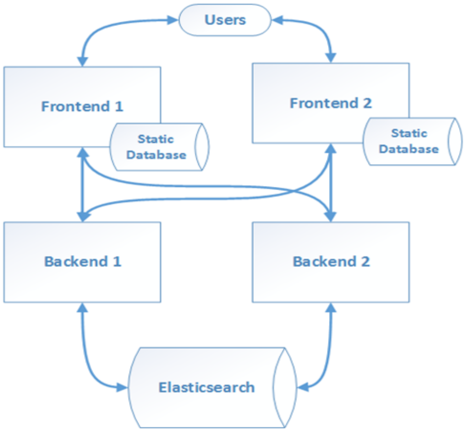

Fig. 5. Hardware structure of the full-text search and visualization system.

The front-end servers receive requests from users. After receiving the request, they submit corresponding processing request to the back-end layer.

All static information used to display website data is stored and delivered by the web server. Such static information includes images, media files, etc. All these data are processed by the OS file system. Delivery of these static data to the servers is carried out during system software deployment or update. This ensures that all static data is identical on all servers in the front-end layer.

Expected performance of data processing by the full-text search and visualization system is at least 10,000 requests per second in total with response delay time under 1 second.

VI. Conclusion

It can be concluded that the clustering methods under consideration are well suited for preliminary processing of Big Data for the purpose of classifying unauthorized Internet users on the basis of indirect data and behavioral characteristics obtained as a result of analyzing mobile traffic and interaction of users with mobile devices. The results of data processing can be analyzed using various software tools that are most suitable for the solution of a particular problem.

The Elasticsearch system is overall the most suitable fir the tasks of full-text search and data visualization (free, open source, simple interface, web-based data processing).

It is proposed to use the capabilities of Elasticsearch to organize the interface to work with Big Data (search and visualization), while for the preliminary processing and the tasks of data segmentation and user verification based on indirect data the MapReduce model can be used, and in particular the new PMR-Transitive approach.

The proposed full-text search and visualization system can cover the demand for a modern innovative software user verification platform that can perform user deanonymization tasks and increase the involvement of users in online economic model. Available methods of authorization and user identification do not cope with the task of obtaining up-to-date and reliable information about users, and authorization methods using such key parameters as alphanumeric login or e-mail address are not sufficient.

Different sectors of economy, such as banking, e-commerce and related Internet services, face problems of fraud and false data.

Implementation of the proposed system can reduce and minimize the risks of real sectors of economy dealing with anonymous service users, which will also have a favorable impact on information security and the state as a whole.

Processing large amounts of data related to the user will allow identifying the user as accurately as possible on the basis of indirect data obtained during the analysis of online behavior, traffic and other user activities.

This work was partially supported by motivational payments system faculty MIREA [28].

Acknowledgment

The work was financially supported by the Ministry of Education and Science of the Russian Federation (State Contract No. 03.G25.31.0280, May 30, 2017).

References

[1] Shmueli G. Research Dilemmas with Behavioral Big Data. Big Data. 2017 Jun;5(2):98-119. doi: 10.1089/big.2016.0043.

[2] Verstrepen K, Goethals B. Top-n recommendation for shared accounts. In: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys'15, New York, NY, 2015, ACM, pp. 59–66.

[3] Jank W, Shmueli G. Modeling online auctions. Hoboken, NJ: John Wiley and Sons, 2010.

[4] Bender S, Jarmin R, Kreuter F, Lane J. Privacy and confidentiality. In: Big Data and Social Science Research: Theory and Practical Approaches. CRC Press, 2016.

[5] Hauge M, Stevenson M, Rossmo D, Le Comber S. Tagging banksy: Using geographic profiling to investigate a modern art mystery. J Spat Sci. 2016;61:185–190.

[6] Narayan A, Shmatikov V. Robust de-anonymization of large sparse datasets. In: Proceedings of 29th IEEE Symposium on Security and Privacy, 2008.

[7] Research GfK: Trends in the development of the Internet audience in Russia. Moscow, 26.01.2017 http://www.gfk.com/fileadmin/user_upload/dyna_content/RU/Documents/Press_Releases/2017/Internet_Usage_Russia_2016.pdf

[8] Shmueli G., Yahav I., 2014, "Tackling Simpson's paradox in Big Data with classification and regression trees", Proceedings of the European Conference on Information Systems (ECIS) 2014, Tel Aviv, Israel, June 9-11, 2014, ISBN 978-0-9915567-0-0

[9] Shmueli G, Koppius O. Predictive analytics in information systems research. MIS Q. 2011;35:553–572.

[10] Demidova L., Nikulchev E., Sokolova Yu. (2016). BIG DATA classification using the svm classifiers with the modified particle swarm optimization and the svm ensembles. international journal of advanced computer science and applications T.7, №5, P. 294-312.

[11] Lo V. The true lift model. ACM SIGKDD ExplorNewslett. 2002;4:78–86.

[12] Shmueli G, Bruce PC, Patel NR. Data mining for business analytics: Concepts, techniques, and applications with XLMiner, 3rd ed. John Wiley and Sons, 2016

[13] Lin M, Lucas H Jr., Shmueli G. Too big to fail: Large samples and the p-value problem. InfSyst Res. 2013;24:906–917.

[14] Trafimow D, Marks M. editorial. Basic and Applied Social Psychology 2015;37:1–2.

[15] Burnham KP, Anderson DR. Model selection and multimodel inference: A practical information-theoretic approach. Springer Science & Business Media, 2003.

[16] Agarwal DK, Chen B-C. Statistical methods for recommender systems. Cambridge University Press, 2016.

[17] Berkhin P. A survey of clustering data mining techniques. In: Kogan J, Nicholas C, Teboulle M, editors. Grouping multidimensional data. Heidelberg: Springer; 2006. p. 25–71.

[18] Dean J, Ghemawat S. Mapreduce: simplified data processing on large clusters. In: Proceedings of the 6th conference on symposium on operating systems design and implementation: 06–08 December 2004; San Francisco. Berkeley: USENIX Association. 2004. p. 137–50.

[19] Zhao W, Ma H, He Q. Parallel k-means clustering based on mapreduce. In: Proceedings of the first international conference on Cloud Computing. 1–4 December 2009; Beijing. Heidelberg: Springer-Verlag. 2009. p. 674–79.

[20] He Y, Tan H, Luo W, Mao H, Ma D, Feng S, Fan J. Mr-dbscan: an efficient parallel density-based clustering algorithm using mapreduce. In: Proceedings of the 17th international conference on parallel and distributed systems: 7–9 December 2011; Tainan. Washington: IEEE Computer Society. 2011. p. 473–80.

[21] Kim Y, Shim K, Kim M-S, Lee JS. Dbcure-MR: an efficient density-based clustering algorithm for large data using mapreduce. Inf Syst. 2014;42:15–35.

[22] Lamari Y., Slaoui SC Clustering categorical data based on the relational analysis approach and MapReduce, Journal of Big Data. 2017. https://doi.org/10.1186/s40537-017-0090-7

[23] Slaoui SC, Lamari Y. Clustering of large data based on the relational analysis. In: Proceedings of the international conference on intelligent systems and computer vision. 25–26 March 2015; Fez. Washington: IEEE Computer Society. 2015. p. 1–7.

[24] Sharkou D.S. A QUICK SEARCH ON THE PROJECTS WITH A HIGH LOADS AND A LARGE AMOUNT OF DATA. Modern technologies: Current issues, achievements and innovations — collection of articles III International Scientific conference / under the general editorship of G. Yu Gulyaev — Penza MCNS « Science and Education » - 2016. P. 23-32

[25] Elasticsearch Available from < https://www.elastic.co/products/elasticsearch>

[26] *Popov G., Magomedov Sh.* Comparative analysis of varios methods treatment expert assessments. International Journal of Advanced Computer Science and Applications. 2017. Т. 8. № 5. С. 35-39. DOI: 10.14569/IJACSA.2017.080505

[27] Magomedov Sh. *Organization of secured data transfer in computers using sign-value notation. ITM Web of Conferences. 2017. Т. 10. DOI: 10.1051/itmconf/20171004004*

[28] V. Pankov.. “The effectiveness of incentive mechanism, and the potential level of satisfaction of the needs of the employee”.Russian Journal of Technology. №4. Pp 288-291. (2015).