Big data analysis for studying spatiotemporal trends in the sustainable development of large cities

Kseniya Mulyukova[1,*] and *Viktor* Kurejchik

[1]Institutes of Academy for Engineering and Technologies of the Southern Federal University, 44, Nekrasovsky lane, 347922, Taganrog, Russia

Abstract. The article covers the analysis of big data in urban planning. The purpose of this work is to study modern problems of processing big data containing information about real estate objects and prospects for solving these problems, as well as the possibility of practical implementation of the methodology for processing such data sets by designing and filling a special graphic abstraction "metahouse" using a practical example. The relevance of the study lies in identifying a number of advantages in the presentation of data in graphical form. The mathematical basis of the technique is the use of multidimensional spaces, where measurements are the characteristics of individual objects. In the course of the work, the specifics of big data sets, consisting of information about real estate in a large city, were described. methods of effective solution of the set practical problem of processing and searching for patterns in a large data array were proposed: abstraction "metahouse", data aggregator. In the course of the study, it was revealed that the presentation of groups of the obtained data in a graphical form has a number of advantages over the tabular presentation of data. The obtained results can be used both for the primary study of big data processing technologies, and as a basis for the development of real applications in the following areas: analysis of changes in the area of houses over time, analysis of changes in the number of storeys in urban development, etc.

1 Introduction

Data processing by means of computer technology is one of the main tasks of most information systems. Any information structured in a certain way can be processed both to obtain direct results of calculations and to prepare for transmission via communication channels or further processing. With the development of means of storage and communication, the amount of information increases non-linearly. In accordance with the laws of dialectics, a quantitative change in the array of processed information goes into a qualitative new state - big data.

Big data is usually understood as such data arrays that cannot be efficiently processed by standard means of computing [1] and storage due to the qualitatively different size of

Corresponding author: mu.ksusha@yandex.ru

information blocks. This forces us to search for and implement fundamentally new methods of working with such data.

Until 2005, most of the data was local and concentrated. In 2005, the concept of Web 2.0 appeared, which radically changes the approach to designing websites. Websites are becoming a dynamic web system with an ever-growing database where information is distributed among various hosts. Also, 2005 was marked by the emergence of mobile and cloud computing. These factors served as the beginning of the era of big data.

In 2008, Clifford Lynch introduced the concept of "big data" in his article. This was the impetus for the formation of big data as a scientific area. Already in 2010, Martz Nathan and Warren James give recommendations for practical work with big data in their book [2].

Nowadays, big data is used in many fields: banking, medicine, e-commerce, web analytics, and urban planning.

Big data in urban planning [3] allows seeing the city from a different angle, makes it possible to effectively solve the problems of the availability of high-quality housing, social comfort, optimization of urban traffic, territorial zoning [4] and many other issues.

One of the famous researchers in the field of big data in urban planning is Professor Rob Kitchin. He is one of the founders of the Programmable City project. The leading area of the project is working with big data concerning people, real estate, transactions, and territories [5].

In 2014, the European Innovative Partnership of Smart Cities and Communities (EIP-SCC) was created [6]. And since 2018, the "Smart City" digitalization project has been implemented in Russia, where one of the aspects of the project is intelligent analytics of big data.

It follows from this that working with big data in urban planning is in demand and relevant today.

The subject of the research is a set of big data consisting of information about real estate in a large city, obtained from open sources, having a certain structure and a sufficiently large volume, making the usual methods of data analysis imprecise or useless [7].

The main differences between such data and linear record sets are:

1. Volume;
2. Independent formation.

To build reliable analytics (for a period of 2 - 3 years) for a large city, which contains information about sales, construction and rent, etc., the data volume will exceed tens of millions of records. We are talking about volumes of data that are significantly higher than the typical volume of a database of an individual company, a registry or a sample of a market analyst site. Traditional methods of obtaining new data, such as sampling averages, median, or grouping by characteristics, give either imprecise or meaningless results.

The aim of the work is to study modern problems of processing big data containing information about real estate objects and prospects for solving these problems, as well as the practical implementation of methods for processing such data arrays by designing and filling a special graphic abstraction "metahouse".

The relevance of the chosen topic is caused by the fact that working with large and inaccurate data containing information about real estate objects changes for the better when presenting big data in graphical form instead of tabular and numerical representations. To do this, it is necessary to obtain data sets from open sources, present them in a traditional form, and then in a graphical form based on the developed abstraction.

2 Materials and methods

Big data usually means generalized data sets that include structured and unstructured data, significant in volume and diverse in structure [8].

There are many characteristics for big data. But we will formulate the main characteristic features that would not quantitatively but qualitatively distinguish them from ordinary, traditional data sets. Like any fuzzy concept, which is most often determined by the fact, it can be described by several features that complement each other. The more features correspond to the studied dataset, the more likely that the array refers to big data. Let's highlight the main features in more detail:

Amount/Volume - big data includes large-scale records that contain online ordering data [9] or clinical data and medical images [10].

Diversity - big data includes thousands of different structures [11]: structured, unstructured, quasi-structured, semi-structured information of various formats. Some of this data does not even have a description of the data at the storage level, which requires rebuilding data processing as new structures are identified.

Distribution - big data is located on many clusters and media.

Time interval - big data contains information over a huge period of time, which requires the introduction of special calculation methods if the calculations relate to temporal dynamics.

Here is a comparative table, where the difference between big data and ordinary information arrays is pointed out. For a more accessible perception of this information, the following table 1 was created.

Table 1. Difference between big data and conventional information arrays.

Feature	Conventional data	Big data
Amount/ Volume	In most cases, data is localized as private databases.	The size of the dataset goes beyond the capabilities of the classical DBMS software.
Distribution	Traditionally, specific data is contained on one or two media, processed in a single application address space and does not exceed the size of the file system.	Distributed on clusters of hundreds and thousands of hosts.
Diversity	Classical data structures are formatted as tuples “field-records” in the case of relational tables or in the form of a repository of standard documents in document-oriented systems.	Contain different types of data: structured, semi-structured, quasi-structured, unstructured.
Time interval	Typical data arrays contain information for a short period of time - the current month, quarter, reporting year. This is sufficient for accounting and primary analysis tasks.	Contain information for a long period of time

All of the above features apply to big data in a generalized form. When using big data consisting of information about real estate in a large city, the listed features are complicated by the specifics of the subject area:

1. Lack of guarantees for the correspondence of one record to one object in the real world - an object can be introduced by different people, from different sources, in different ways, and finally, at different times, sometimes very long. Thus, taking the average of the records, we will not get the real average in the real world. If some sources have a larger amount of data on a specific attribute of an object, then the calculation will be deliberately incorrect, and the new data will be far from the actual indicators.
2. Possible incompleteness in many records - information obtained from various sources, most often does not have a rigidly defined structure. This imposes additional requirements

for analysis: it is necessary to skip missing data, while taking into account those that are explicitly set.

To implement the processing task, let us describe the data model in the most suitable way for analysis and implement our own algorithm for processing big data in order to obtain new knowledge from the incoming information.

Although this paper does not imply a full-fledged development of a software solution, we will describe individual parts of the system in C# 4.0 in the form of classes. Big data is stored on the MongoDB server [12], which is suitable for both document-oriented records and fuzzy data [13]. For data visualization and user access interface, a web interface based on HTML and CSS is used, which allows using the system on any HTTP server, both in a local network and on a remote dedicated server.

3 Results

3.1 Development of a solution algorithm

The developed algorithm for analyzing big data consisting of information about real estate in a large city is based on two ideas:

1) Abstraction “metahouse”;

2) Aggregator [14] of data from different sources, filling the abstraction “metahouse” based on incoming records.

A “metahouse” is an abstract object that does not exist in the real world, but most resembles a typical house, apartment, or other real estate object in a given set of big data.

The aggregator divides record data into two parts: construction criteria and metahouse properties. The aggregator also calculates these properties based on the developed algorithm: let’s represent all records of the big data array as points in a multidimensional space N, where dimensions 0…K are construction parameters, and dimensions K+1…N-1 are properties of records. Then the “metahouse” will have the following property: it is located so that the sum of the distances from it to all records in the property space is the minimum possible. Due to this, we reduce the influence of errors in the original big data set, repeated and inaccurate data perform a shift in the coordinates of the “metahouse”, but to a lesser extent than with conventional computational methods, such as calculating the average or median value.

3.2 Solution architecture

The big data solution architecture consists of three blocks:

1) collecting records from sources in the repository by converting to the format of a specific class;

2) passing this data through the aggregator to obtain a set of filled “metahouse” in accordance with the analysis settings;

3) presentation of processed data in tabular and graphical form for the user of the big data analysis system.

A generalized diagram of a big data collection and analysis solution architecture is shown in Figure 1.

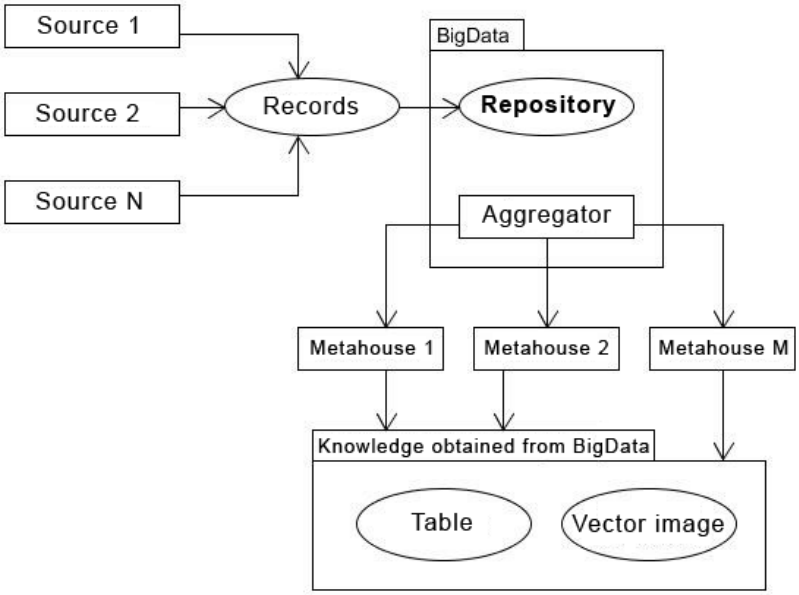

Fig. 1. Generalized solution architecture diagram.

3.3 Input data and their aggregation

Let us describe the structure of the input data for a cluster solution for processing big data on MongoDB [15]. Each entry in Figure 2 is a JSON structure [16], independent of the others.

```
{
        "datetime":"",
        "Floor":"",
        "FloorAll":"",
        "Price":"",
        "S":"",
        "Sk":"",
        "Sh":"",
        "TypeCode":"",
        "OperCode":"",
        "Lat":"",
        "Lon":""
}
```

Fig. 2. The structure of the input data.

Missing or incomplete data is represented by a null code to avoid confusion with null values, which, unlike incomplete data, have a different mathematical meaning.

Class aggregation can be performed in a relational database, file storage, cloud format [17], or in a document-based database as chosen in our implementation. The only important thing is that due to the declared amount of data, it is not possible to contain them in ordinary collections like a vector or a list [18]. The accumulation of big data is ongoing, the flow of data is carried out by analyzing open sources, collecting data from websites, and adding archived data of organizations. The conditions for controlling the reliability,

accuracy and relevance of data are not specified here. A qualitative transition is achieved due to the amount of data, even if some of the information contains inaccurate or deliberately incorrect information.

The aggregator class in Figure 3 has only two methods - setting parameters and receiving a new object at the input. It also contains a link to an object, which is an element in the structure of the code, but in architecture, it is a metahouse, an abstract object that has the most typical characteristics for the given parameters after all the entries have been made.

```
public class Agregator
    {
        public void setParameter(String code, int value);
        public void addElement(Element el);
        public Element result;
```

Fig. 3. Aggregator class.

Such a simplified approach to architecture [19], nevertheless, allows solving the problem in full. The passage of all records through the aggregator allows getting the output data and can be repeated with different sets of parameters to analyze the spatiotemporal states of objects.

3.4 Output data and its presentation

We have split the output data in two parts:

1. Tabular component [20] - we get some numerical data, which represent new knowledge about the array of urban real estate, obtained from the full set of data for the entire period;
2. Visual component - allows you not only to get numbers, but also to see them in a visual form.

The tabular component is represented by ordinary tables according to the characteristics of the object, in the form of key-value pairs. These tables can be loaded into additional programs and, in turn, used as a data source for deeper analysis. Table 2 shows an example of the output data in tabular form - a “metahouse” with specific characteristics.

Table 2. Tabular view of a “metahouse”.

Floor	2.5
Number of storeys	5.7
Price	2530000
Area	60.89
Living area	30.12
Kitchen area	7.51
Type	Apartment
Operation	Demand

The class in Figure 4, which performs the given construction, has one method that generates a table based on the “metahouse” object

```
public class TableBuilder
    {
        public String getTableView(JSONObject el);
    }
```

Fig. 4. A class that forms a table.

Visualization allows displaying information about the "metahouse" in a graphical form [17], so that the user can quickly understand the meaning of the results obtained and use them to extract practical benefits, for example, analyzing the dynamics of urban development or tendencies of changes in the area of houses.

At the software level, visualization is implemented as the output of a vector schematic image of one or more houses, the parameters of which (height, width, line thickness, image type) are responsible for the presentation of individual characteristics. Images are signed by criteria and grouped according to one or several parameters. An example of visualization of one "metahouse" of the house type is shown in Figure 5.

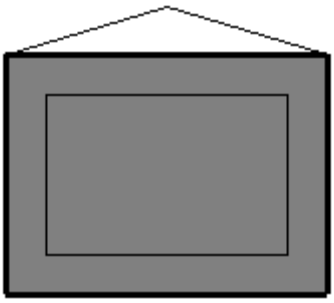

Fig. 5. An example of the "metahouse" visualization by a vector image.

Specific characteristics of visualization of "metahouses":

Area - displayed as the base of the vector diagram. The larger the area, the wider the base. Area dependence is calculated as square root.

Number of storeys - displayed as the height of the vector diagram. The higher the floor, the higher the diagram. Floor dependence is linear.

Price - displayed as line thickness of the vector diagram. The higher the price, the thicker the line. Price dependence is logarithmic.

Supply and demand - set by the background fill color of the vector diagram, where white code 0xFFFFFF is absolute demand, and gray code 0x808080 is absolute supply.

Parameter scatter - represented by an inner box that shows how wide the scatter is within a given big data array. The farther the inner frame is from the outer one, the greater the scatter of parameters was found during the construction of this "metahouse".

Thus, Figure 5 shows a "metahouse" of the house type with a large area, average number of storeys, a predominance of supply over demand, a high price, and a moderate spread of parameters.

Here are some more output visualizations for comparison regarding parameter changes. Figure 6 shows a "metahouse" with a smaller parameter spread, a lower price, and a predominance of demand over supply.

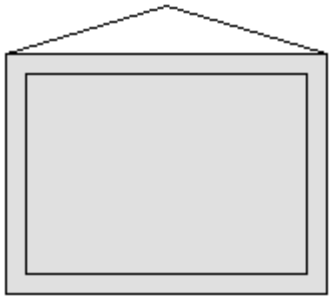

Fig. 6. An additional example of the "metahouse" visualization by a vector image.

Figure 7 shows a "metahouse" of the apartment type with a high number of storeys, a small area, balanced supply and demand, and a large scatter of parameters.

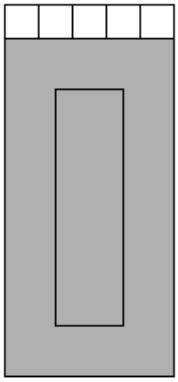

Fig. 7. An additional example of the "metahouse" visualization by a vector image.

A similar "metahouse", but with an absolute predominance of demand over supply, a high price and a minimum scatter of parameters is shown in Figure 8.

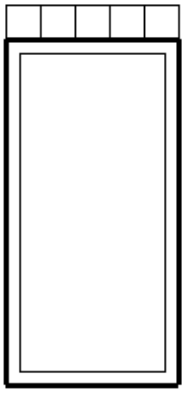

Fig. 8. An additional example of the " metahouse" visualization by a vector image.

The class in Figure 9, which implements a graphical representation similar to a textual one, has a method for creating an image from an incoming object.

```
public class GraphicBuilder
    {
        public Bitmap getGraphicView(Element el);
    }
```

Fig. 9. A class for creating an image from an incoming object.

4 Discussion

Working with big data is still quite difficult from a practical point of view and requires additional theoretical study.

In the course of the research, it was hypothesized that the presentation of big data in graphical form has several advantages over the traditional presentation of big data in tabular form. To prove the hypothesis, an abstraction “metahouse” was designed by processing big data.

Tabular data for a large city was obtained by analyzing three million records containing more than 10 groups of data with a basic set of parameters: floor, number of storeys, price, area, living area, kitchen area, type, operation. To solve the problem of processing such an array of data, a cluster was created on MongoDB on several computers, each of which was engaged in its own set of data without converting intermediate results.

The result of this data mining is an 8 by 10 cell table, which has several disadvantages:

1. the table is difficult to reduce without losing the convenience of the perception of numerical indicators. This visual experience is inconvenient for the user;
2. to perceive the differences between different groups of data for different indicators, it is necessary to build graphs. This, in turn, leads to additional economic and time costs.

The use of the concept of “metahouse” and the presentation of groups of the obtained data in a graphical (vector) image has a number of advantages over the tabular presentation of data:

3. vector image is easy to scale;
4. presentation of different indicators in different parts of the image allows them to be compared without building graphs - the height, width, color and thickness of the lines immediately provide a clear and visible comparison for the user.

The proposed scheme for visualizing big data by constructing abstract vector images with variable characteristics is an alternative to traditional tables or diagrams, allows looking at data arrays and the results of their processing from a different angle.

The results of the computational experiment showed that when using the graphical form (vector) for presenting big data, the costs and time for interpreting data mining were reduced.

The combination of methods for processing big data and their presentation through graphical abstraction allows getting new results for existing datasets.

5 Conclusions

The obtained results can be used both for the primary study of big data processing technologies and as a basis for the development of real applications in the following areas:

1. Analysis of changes in the number of storeys in urban areas by city districts - by writing the parameters of the big data aggregator for coordinates in the code, it is possible to get a table with “metahouses”, the number of storeys of which will reflect the real state of the

floors of urban areas and how it changes in space. Such indicators of building height are important both for rescue services and for optimizing urban traffic.
2. Analysis of changes in the area of houses over time - by writing the parameters of the big data aggregator for a time, it is possible to get a table with "metahouses", the characteristics of the areas of which will reflect the real state of urban areas and how it changes over time. Such indicators make it possible to determine more well-built areas, and when analyzing time frames - to assess the speed of solving the problems of the availability of high-quality housing.
3. Dynamics and distribution of supply and demand - by writing the parameters of the big data aggregator for time or coordinates in the code, it is possible to get a visualization from which the ratio of supply and demand in different regions in different years will follow. Such dynamics are very important for identifying growing or depressed areas, and predicting a crisis or market growth based on stably changing aggregate indicators.
4. Analysis of price dynamics by time, regions and individual types of objects - although prices are not objective indicators, unlike area or number of storeys, nevertheless, visualization of distributed market prices allows getting an overall picture of urban development, highlighting more active areas among others and observing for a change in the state of the market for a long period of time.

By adding tools for providing public APIs and developer documentation to the developed system, it is possible to obtain an effective tool for organizations and individuals who need reliable data on urban development.

Acknowledgement

The work was carried out at the expense of partial funding under the grant RFBR GR No. 18-29-22019.

References

1. S.S. Valeev, N.V. Kondratyeva, Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki **2(160)**, 392–398 (2018)
2. N. Marz, A.J. Warren, *Big Data: Principles and Best Practices of Scalable Real-Time Data Systems* (Manning Publications, Shelter Island, 2017)
3. A.R. Honarvar, A. Sami, Big Data Research **17(22)**, 222-226 (2019) DOI: 10.1016/j.bdr.2018.05.006
4. X. Xiao, X. Chao, Environmental Technology & Innovation **21**, 65-76 (2021) DOI: 10.1016 / j.eti.2021.101381
5. R. Lea, *Smart Cities: An Overview of the Technology Trends Driving Smart Cities. Researchgate* (2017) https://www.researchgate.net/publication/326099991_Smart_Cities_AnOverview_of_the_Technology_Trends_Driving_Smart_Cities
6. N. Ivanov, M. Gnevanov, Business Technologies for Sustainable Urban Development (2018) DOI: 10.1051/matecconf/201817001107
7. S. Mitrovic, Business informatics **4(42)**, 40–46 (2017) DOI: 10.17323/1998-0663.2017.4.40.46
8. S. Sivarajah, M.M. Kamal, Z. Irani, V. Weerakkody, Journal of Business Research **70**, 263-286 (2017) DOI:10.1016/j.jbusres.2016.08.001

9. P.A. Hurtado, C. Dorneles, E. Frazzon, IFAC-PapersOnLine **52(13)**, 838-843 (2019) DOI: 10.1016/j.ifacol.2019.11.234.
10. L. Hong, M. Luo, R. Wang, P. Lu, W. Lu, L. Lu, Data and Information Management **2(3)**, 175-197 (2018) DOI: 10.2478/dim-2018-0014
11. D. Zhang, 8th International Conference on Management and Computer Science **15(2)**, 275–278 (2018) DOI: 10.2991/icmcs-18.2018.56
12. S. Sharma, The international Journal of Big Data Intelligence **2(3)**, 201 - 221 (2015) DOI: 10.1504 / IJBDI.2015.070602
13. H. Abbes, F. Gargouri, *20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems* (2016) DOI: 10.1016/j.procs.2016.08.099
14. M. Babar, F. Arif, Future Generation Computer Systems **77**, 65-76 (2017) DOI: 10.1016/j.future.2017.07.029
15. S. Heripracoyo, R. Kurniawan, Telkomnika **14(3)**, 1083-1089 (2016) DOI: 10.12928/TELKOMNIKA.v14i3.3115
16. A. Celesti, M. Fazio, M. Villari, Future Internet **11(83)**, 1-17 (2019) DOI: 10.3390/fi11040083
17. C. Yang, Q. Huang, Z. Li, K. Liu, F. Hu, International Journal of Digital Earth **10(1)**, 13-53 (2017) DOI: 10.1080/17538947.2016.1239771
18. G.M. Novikova, E.J. Azofeifa, Discrete and Continuous Models and Applied Computational Science **4(26)**, 383 – 392 (2018) DOI: 10.22363/2312-9735-2018-26-4-383-392
19. E. Ignatova, S. Zotkin, I. Zotkina, IOP Conference Series: Materials Science and Engineering **365(6)**, 1-9 (2018) DOI: 10.1088/1757-899X / 365/6/062033
20. K. Venkatraman, S. Venkatraman, *Proceedings of the 3rd International Conference on Big Data and Internet of Things* (New York, USA, 2019) DOI:10.1145/3361758.3361768