UDC 520.6.05
DOI https://doi.org/10.32838/2663-5941/2019.6-1/10

Dychka I.A.
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Yerastova V.V.
National Technical University of Ukraine "Igor Sikorsky Kyiv polytechnic institute"

Oleshchenko L.M.
National Technical University of Ukraine "Igor Sikorsky Kyiv polytechnic institute"

Yurchyshyn V.Ya.
National Technical University of Ukraine "Igor Sikorsky Kyiv polytechnic institute"

SOFTWARE METHOD OF SCIENTISTS CLUSTERING BASED ON AUTHORS ASSOCIATION WITH PUBLICATIONS KEYWORDS

The article describes a software method that allows clustering authors based on their association with the keywords of scientific publications to search for potential like-minded colleagues and colleagues to develop collaborative research projects. The proposed work aims to demonstrate the potential and challenges facing working with Big Data. Big Data technologies make it possible to process a large amount of unstructured data, systematize them, analyze and determine patterns when the human brain will never notice them. The article describes the basic tools and methods that are used to process very large data sets. The MapReduse methodology implemented in Python and Apache Hadoop are used. For research a data set from a DBLP server is used. DBLP contained more than 1.2 million bibliographic records. The bibliographic records are contained in XML file2. The DBLP web server lists all known papers published by a person on her/his "person page". This simple mapping becomes tricky, as soon as a person has several names (synonyms) or if there are several persons with the same name (homonyms). The main obstacles are the bad habit to abbreviate (given) names beyond recognition and spelling errors. On the AWS m5.xlarge instance and Ubuntu Server 18.04 LTS (HVM), SSD Volume Type, 64-bit x86 is used. For processing data MapReduce are running on five nodes. Mapping part consists of parsing xml dataset. Tags <author>, <journal> and <title> tags from each <article> block is extracted. Then non-ascii characters from every parsed tag is removed. Title is divided into keywords. It is made lowercase and then we remove nonsense words from it, so, it will be like thematic keywords. Reducer part is collecting information for every author, made it as id for reducing. The author's name is compared in way that it written. As a result, the information for every author is collected about the journals where he was published and keywords in his works are collected too. The k-means algorithm is used for clustering. As a result of the analysis of these publications, the authors were divided into clusters depending on their keywords. The disadvantages and prospects of further use of the proposed method are given.

Key words: *Big Data, AWS, Hadoop, MapReduse, scientific publication, author, keywords, cluster analysis, k-means algorithm, Python.*

Introduction. Problem statement. Methods and tools for working with structured data IT industry has created a long time ago, this is a relational data model and database management system. But the current trend is the need to process a large amount of unstructured data, and this is an area where previous approaches work poorly or do not work at all. This need requires a new method of handling data, and now the model for working with Big Data is becoming increasingly popular. Dealing with the problems of Big Data is the task of finding a software and technical solution that can easily integrate into the existing infrastructure and provide all three stages of information processing: collection, its organization and analysis [1]. Big Data technology has interconnection with a network of publications. A regular person reads a page of text in about 2 minutes. So, reading 10 to 20 pages of a scientific paper will take approximately 20-40 minutes. How long will it take to read hundreds, thousands, or even millions of such papers? Not easy task, we can say that it is not even feasible task for one person or even group. But let's imagine that group of people can do it in reasonable time, but the next task is to combine their acquired knowledge from these scientific publications and establish correlations between articles and terms of

interest, find common patterns related to a specific subject.

This is one of the major challenges now facing health professionals and researchers. It's estimated that around 2 million new scientific publications appear every year (with an exponential growth in the past decade), which brings the total to well over 50 million publications documented. Obviously, the advancement of science and society relies on researchers sharing the knowledge and results of their arduous work via scientific publications. However, when it comes to consuming and mining that vast amount of information, there's clearly room for improvement [2].

Related research. In paper [3] an unrelated approach, which integrates community detection method and author-topic (AT) model is proposed. Specifically, it consists of three steps: to detect overlapping academic communities with the clique percolation method, to discover underlying topics and research interests of each researcher with author-topic (AT) model, and to label research topics of each community with top n most frequent collaborative topics between members belonging to the community, where common topics between researchers are seen as collaborative topics. The framework is composed of four parts: preprocessing data, detecting communities in scientific collaboration network, discovering collaborative topics between authors and to uncovering topics of academic communities (fig. 1).

The method integrates community detection model using *k*-clique-community algorithm and the author-topic model. The approach of *k*-clique-community algorithm is to detect overlapping communities in scientific collaboration network, while the approach of AT model is to discover topics and authors' topics. Authors use common topics of coauthored researchers as their collaborative topics. Finally, they count all collaborative topics and select the most frequent collaborated topics among authors as research topics of communities [4].

In paper [5] a principled framework that goes beyond regular temporal community detection and can capture sudden changes in network topologies in dynamic cases is proposed. In this framework, temporal communities are detected with the smoothness constraint that temporal expected node popularity preservation is enforced. They argue that although the observed network may change rapidly, its latent structure (e.g., node popularity) often evolves much more

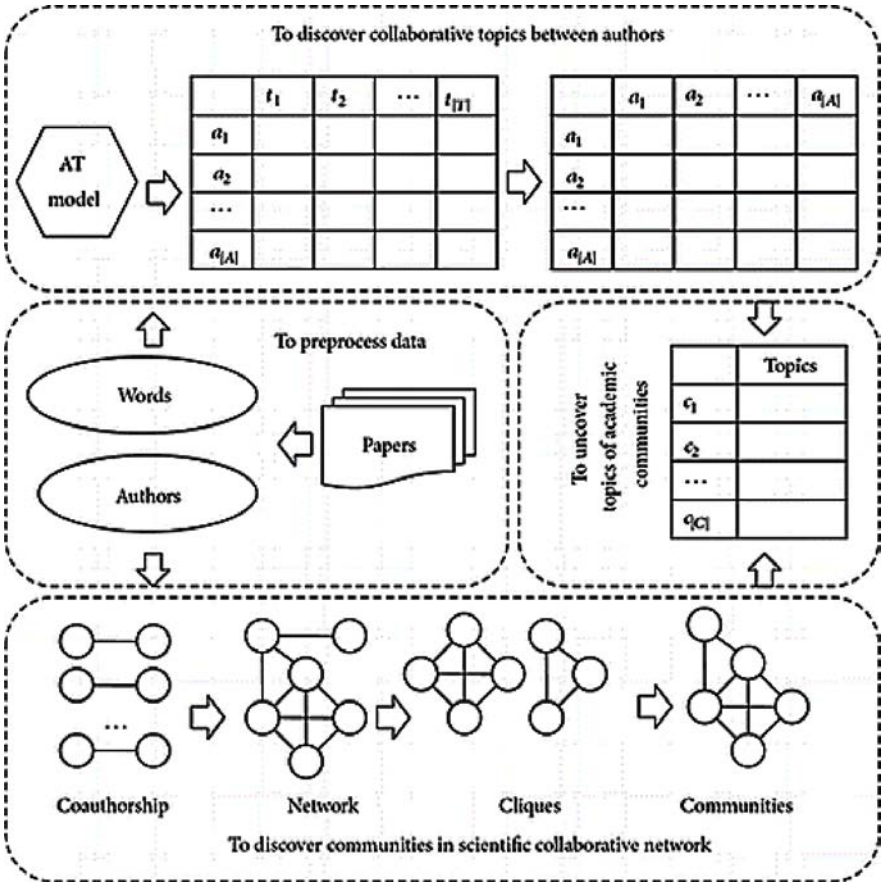


Fig. 1. Framework of the method [3]

slowly. For example, contact-based social networks might change from day to day because of people's varying daily activities, but the popularity of the individual is much more stable. By identifying such latent time-persistent structure, they are able to uncover the fundamental continuous evolution of the forces that form the structure of networks. In most model formulations of existing methods, temporal communities are detected with the smoothness constraint that the communities that the nodes belonging to are stable, but this is not consistent with the evolving real-world networks. It often occurs that the left of the community hub nodes from one community would result in the movements of large number of nodes in that community. In this case, the community membership has a big change, and hence methods with the enforced community membership smoothness constraint will suffer from poor performance of community detection. To solve this problem, they introduce the community membership transition matrices in our model. The transition matrices are able to detect significant changes among detected communities and further reveal the movements of the inner nodes [5].

The main goal of the article is to solve problem of association and correlation analysis in scientific publications and given a set of keywords describing research area/topics:

1. define and discover associations between authors and keywords (taken from title and/or the abstract);

2. perform clustering of authors based on their association to keywords so as to recommend potential collaborators.

Dataset and methodology of research. Developed software allows to analyze and discover interesting collaborations among scientists and researchers. For this task dataset DBLP was used. DBLP contained more than 1.2 million bibliographic records. The bibliographic records are contained in XML file2. The DBLP web server lists all known papers published by a person on her/his "person page". This simple mapping becomes tricky, as soon as a person has several names (synonyms) or if there are several persons with the same name (homonyms). The main obstacles are the bad habit to abbreviate (given) names beyond recognition and spelling errors. The DBLP data set is available from the location http://dblp.uni-trier.de/xml/

The file dblp.xml contains all bibliographic records which make DBLP. It is accompanied by the data type definition file dblp.dtd. dblp.xml has a simple layout:

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dblp SYSTEM "dblp.dtd">
<dblp>
record 1
...
record n
</dblp>

The header line specifies ISO-8859-1 ("Latin-1") as the encoding, but in fact the file only contains characters [5]. In this work MapReduce methodology implemented on Python programming language and Apache Hadoop is used. We use Hadoop distributed file system (HDFS), which is responsible for storing data on a Hadoop cluster and MapReduce system

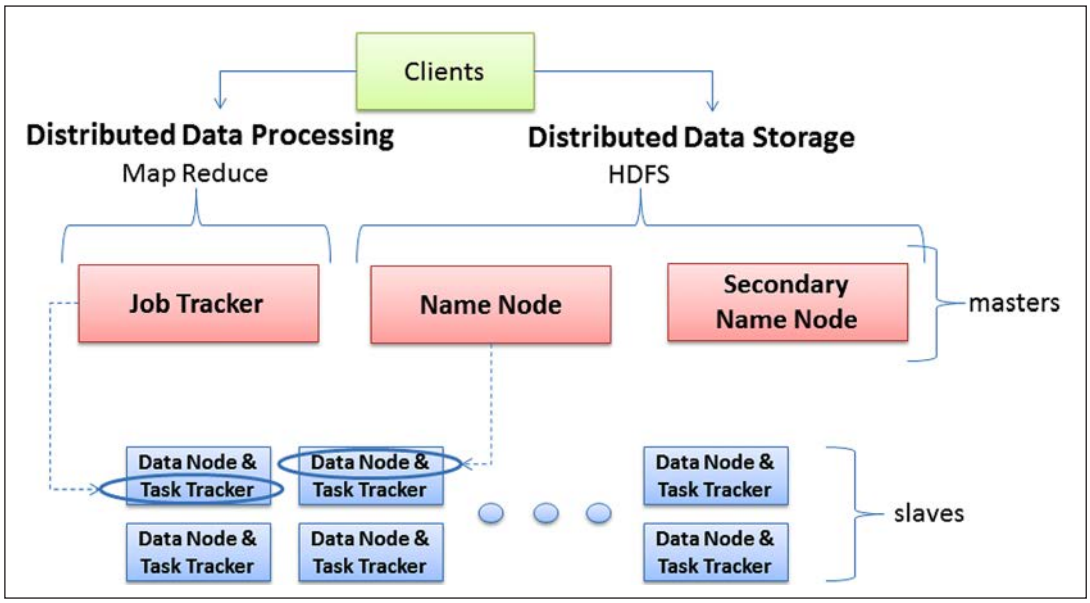


Fig. 2. Hadoop server roles [6]

designed for computing and processing large amounts of data on a cluster.

HDFS repeatedly copies data blocks and distributes these copies among the compute nodes of the cluster, thereby ensuring high reliability and speed of calculations. The Hadoop cluster configuration consists of one name server, one MapReduce wizard (JobTracker) and a set of working machines, each of which simultaneously runs a data server (DataNode) and a worker (TaskTracker) (fig. 2).

Each MapReduce runs in parallel and (if possible) locally on each data block. Instead of delivering terabytes of data to a program, a small, user-defined program is copied to the servers with data and does everything with them that does not require shuffling and data movement. On the AWS m5.xlarge instance and Ubuntu Server 18.04 LTS (HVM), SSD Volume Type, 64-bit x86 is used. For processing data MapReduce are running on five nodes. Mapping part consists of parsing xml dataset. Tags <author>, <journal> and <title> tags from each <article> block is extracted. Then non-ascii characters from every parsed tag is removed. Title is divided into keywords. It is made lowercase and then we remove nonsense words from it, so, it will be like thematic keywords. Reducer part is collecting information for every author, made it as id for reducing. The author's name is compared in way that it written. As a result, the information for every author is collected about the journals where he was published and keywords in his works are collected too. The next step was to do a clustering among keywords that we have after the MapReduce part. Keyword clustering is the process of combining related and similar keywords into separate clusters (groups). This process can show us a distribution of keywords used by authors and gives a very useful information for a future work. In the research *k*-means algorithm is used.

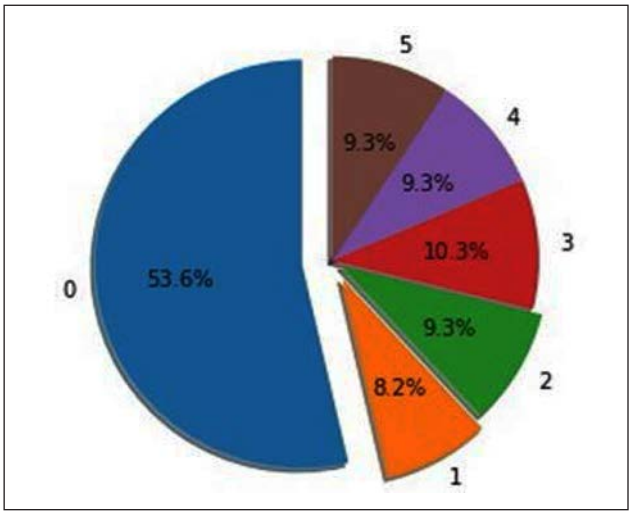
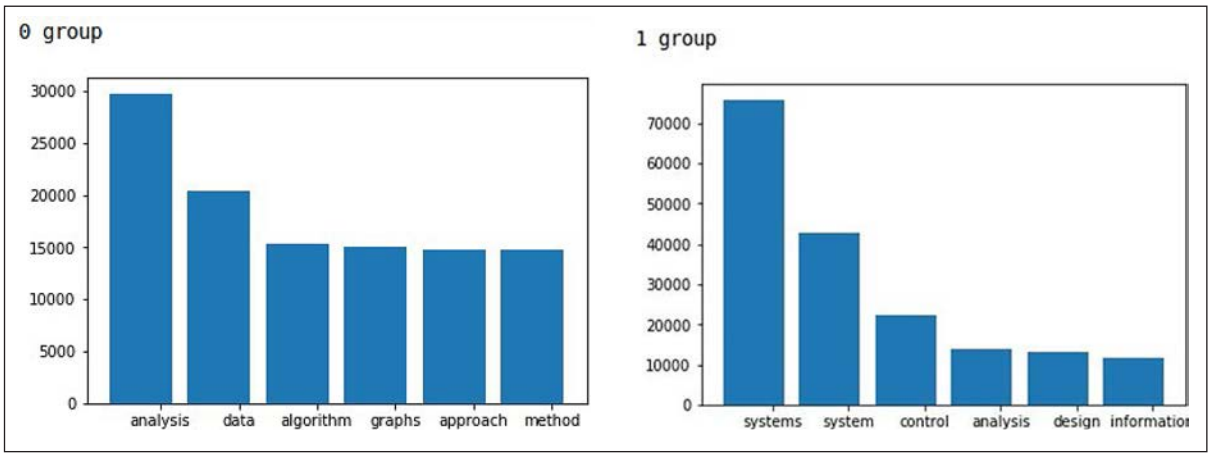


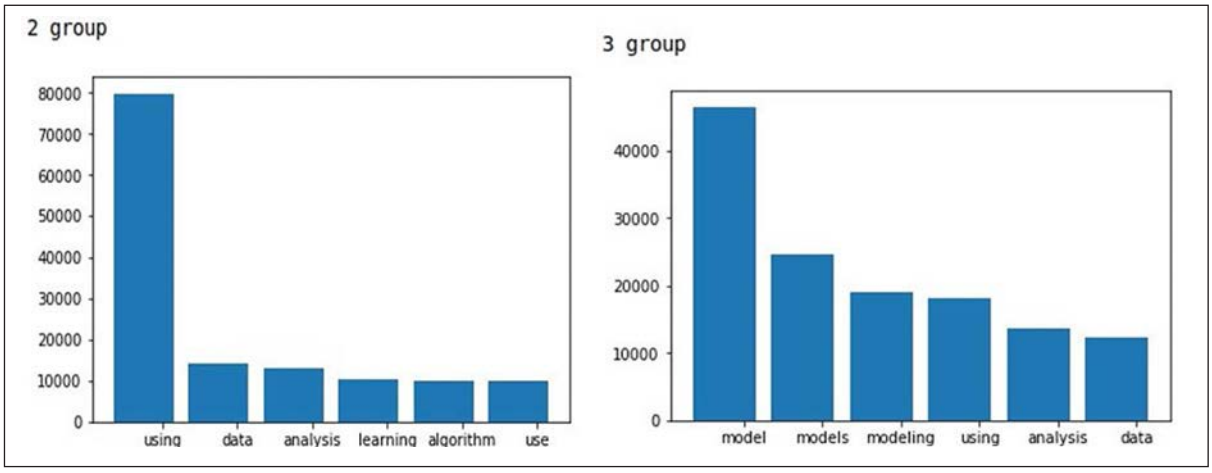
Fig. 3. Authors clustering (groups 0-5)

Given a set of observations $(x_1, x_2, ..., x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k ($\leq n$) sets $S = \{S_1, S_2, ..., S_k\}$ so as to minimize the within-cluster sum of squares. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster [7]. It splits the set of elements of a vector space into a previously known number of clusters k. The algorithm seeks to minimize the standard deviation at the points of each cluster. The basic idea is that at each iteration the center of mass is recalculated for each cluster obtained in the previous step, then the vectors are divided into clusters again, according to which the new centers was closer in the selected metric. The algorithm terminates when no cluster changes at any iteration.



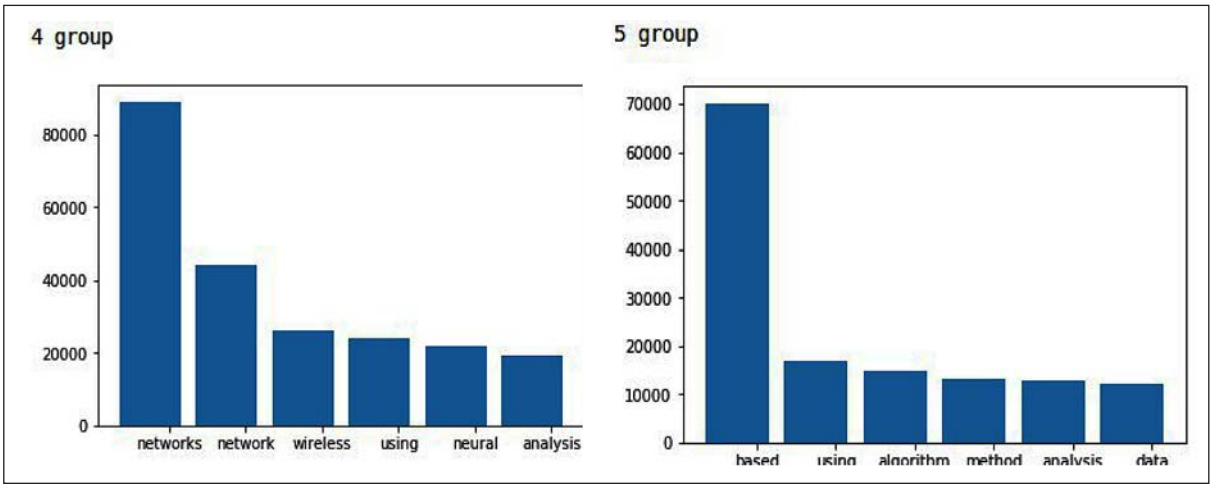
a. **b.**

Fig. 4. Most common keywords of a. a. – 0 cluster; b. – 1 cluster



a. **b.**

Fig. 5. Most common keywords of a. – 2 clusters; b. – 3 cluster



a. **b.**

Fig. 6. Most common keywords of a. – 4 clusters; b. – 5 clusters

Given an initial set of k centers, the k-means algorithm alternates the two steps:

1. for each center we identify the subset of training points (its cluster) that is closer to it than any other center;
2. the means of each feature for the data points in each cluster are computed, and this mean vector becomes the new center for that cluster.

These two steps are iterated until the centers no longer move or the assignments no longer change. Then, a new point x can be assigned to the cluster of the closest prototype. Using Python for implementation of the k-means algorithm, we get analysis of publications dataset. Authors were divided into clusters which depends on their keywords (fig. 3).

Each cluster has its own most common keywords.

Journals for publishing were analyzed. The most frequent journals are shown on fig. 7.

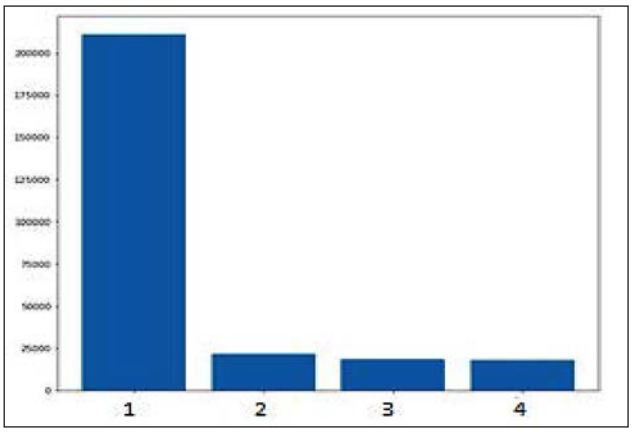


Fig. 7. Amount of publications in journals (1 – "CoRR"; 2 – "IEICE Transactions"; 3 – "Sensors"; 4 – "Applied Mathematics and Computation")

Conclusions and future work. In this research the clustering of authors of scientific publications based on their association with keywords using the k-means algorithm and Python is done. The selected algorithm has the following disadvantages for this

task. We must know in advance the number of clusters. The algorithm is very sensitive to the choice of initial cluster centers. The classic version implies a random selection of clusters, which very often was a source of error. As a solution, it is necessary to conduct the object studies to determine the centers of the initial clusters more accurately. Does not cope with the task when the object belongs to different clusters equally or does not belong to one. So one of the tasks for a future work is to use a more powerful algorithm for the clustering. The work can be improved in several ways: tracking a popularity of journals for publication in different times, a recommendation system for choosing a journal depending on the main theme, using several big datasets with exclusion of the same records.

References:

1. Олещенко Л.М. Технології Big Data аналітики в розподілених системах обчислень. *Проблеми інформатизації та управління.* № 4(60). 2017. С. 57–63.

2. Arif E. Jinha, Article 50 million: an estimate of the number of scholarly articles in existence. URL: https://onlinelibrary.wiley.com/doi/abs/10.1087/20100308.

3. Han H. Uncovering Research Topics of Academic Communities of Scientific Collaboration Network. URL: https://journals.sagepub.com/doi/pdf/10.1155/2014/.

4. Yuan L. Tracking and Detecting Dynamic Communities with Node Popularity Preservation. URL: https://ieeexplore.ieee.org/document/8371994.

5. Ley M. DBLP — Some Lessons Learned. URL: https://dblp.uni-trier.de/xml/docu/dblpxml.pdf.

6. Hedlund B. Understanding Hadoop Clusters and the Network. http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters.

7. Vettigli G. K-Means Clustering with Scipy. URL: https://dzone.com/articles/k-means-clustering-scipy.

Дичка І.А., Єрастова В.В., Олещенко Л.М., Юрчишин В.Я. ПРОГРАМНИЙ МЕТОД КЛАСТЕРИЗАЦІЇ НАУКОВЦІВ НА ОСНОВІ АСОЦІАЦІЇ АВТОРІВ З КЛЮЧОВИМИ СЛОВАМИ ПУБЛІКАЦІЙ

У статті описано програмний метод, який дозволяє згрупувати авторів на основі їх асоціації з ключовими словами наукових публікацій для пошуку потенційних однодумців та колег для розробки спільних науково-дослідних проектів. Запропонована робота має на меті продемонструвати потенціал та проблеми Big Data. Технології великих даних дають можливість обробляти велику кількість неструктурованих даних, систематизувати їх, аналізувати та визначати закономірності, коли людський мозок їх ніколи не помітить. У статті описані основні інструменти та методи, які використовуються для обробки дуже великих наборів даних. Використовується методологія MapReduse, реалізована в Python та Apache Hadoop. Для дослідження використовується набір даних з сервера DBLP. DBLP містить понад 1,2 мільйона бібліографічних записів. Бібліографічні записи містяться у файлі XML file2. Веб-сервер DBLP перераховує всі відомі документи, опубліковані людиною на її «особі». Це просте відображення стає складним, як тільки у людини є кілька імен (синонімів) або якщо є кілька осіб з однаковим іменем (омоніми). Головними перешкодами є погана звичка скорочувати (задані) імена, що зумовлює помилки під час розпізнавання та написання. Для екземпляра AWS m5.xlarge та Ubuntu Server 18.04 LTS (HVM) використовується SSD, 64-розрядний x86. Для обробки даних MapReduce працює на п'яти вузлах. Частина картографування складається з аналізу даних XML. Теги <author>, <journal> і <title> теги витягуються з кожного блоку <article>. Потім символи, що не належать до об'єктів, видаляються з кожного проаналізованого тегу. Заголовок поділяється на ключові слова. Він утворюється з малих літер, а потім видаляються недоречні слова, тож залишаються тільки тематичні ключові слова. Частина редуктора – це збирання інформації для кожного автора, що робить її ідентифікатором для скорочення. Ім'я автора порівнюється з тим, як воно написане. Як результат збирається інформація для кожного автора про журнали, де він був опублікований, а також збираються ключові слова у його роботах. Алгоритм k- means використовується для кластеризації. В результаті аналізу цих публікацій автори були розділені на кластери залежно від їх ключових слів. Наведено недоліки та перспективи подальшого використання запропонованого способу.

Ключові слова: *Big Data, AWS, Hadoop, MapReduse, наукова публікація, автор, ключові слова, кластерний аналіз, k- means алгоритм, Python.*