DOI: https://doi.org/10.15276/aait.05.2022.23
UDC 004.94

Development of infrastructure for anomalies detection in big data

Iuliia L. Khlevna[1)]
ORCID: http://orcid.org/0000-0002-1874-196; yuliia.khlevna@knu.ua. Scopus Author ID: 57191869873
Bohdan S. Koval[1)]
ORCID: http://orcid.org/0000-0002-3757-0221; bohkoval@gmail.com. Scopus Author ID: 57200141737
[1)]Taras Shevchenko National University of Ukraine, 60, Volodymyrska Str. Kyiv, 01033, Ukraine

ABSTRACT

The work describes the conducted analysis of models, methods, and technologies for detecting anomalies in data. It concludes that, based on the analysis, the solution to the problem of anomaly detection in data should be considered as a complex technology, which consists of the formation and application of mathematical models in combination with the research of data processing approaches. The article analyses the current state of big data stream processing technologies and reflects the peculiarities of the most commonly used and advanced of them, e.g. Apache Hadoop, Apache Spark, Apache Cassandra, Apache Kafka, Apache Storm, and Apache Beam. On top of these, it pays attention to the infrastructure, in which the created software models can be deployed and used, taking into account the high-load real-time nature of the data. The article proposes to form an infrastructure for anomaly detection in data as an applied example of big data processing cloud infrastructure. The paper demonstrates the developed infrastructure model for anomaly detection in real-time stream data, which is based on an expert method of forming requirements for a software component, choosing an algorithm for detecting anomalies, selecting tools, and improving the algorithm. The highlighted anomaly detection tools allow us to create a secure real-time anomaly detection solution using Dataflow, BigQuery ML, and Cloud DLP. The paper presents the applied implementation of anomaly detection in real-time using GCP and Apache Beam - data stream analysis of software logs in the information system and detection of fraudulent ones among them, which will help improve the cyber security of the system. In the end, the work demonstrates possible improvements to the basic model that could help to speed it up.

Keywords: Big data; anomaly detection; cloud computing; data processing; data ingestion

For citation*: Khlevna Iu. L., Koval B. S.** "**Development of infrastructure for anomalies detection in big data"**. ***Applied Aspects of Information Technology. **2022; Vol.5 No.4: 348–358. DOI: https://doi.org/10.15276/aait.05.2022.23**

INTRODUCTION

The active production of data in digital systems makes it impossible to process them in traditional ways and urges the development of big data processing methods. Anomaly detection in big data sets is the finding and identification of elements, events, or observations that do not correspond to expected behavior (patterns) or other elements of the data set [1]. Sometimes abnormal elements can cause many problems, for example, bank fraud, medical problems, problems with finding errors in text, etc. Thus, the problem of anomaly detection is quite widespread and can be used in many spheres of activity. In the literature, the term anomaly is synonymous with outliers, novelty, noise, deviations, and exceptional situations [2]. In a more classical sense, anomaly detection is a variant of the more general task of classification using discrete methods (k-nearest neighbors, decision trees, Bayesian method), regression methods (support vectors machine, logistic regression), and also using neural networks. However, in a broader sense, solving the problem of detecting anomalies is not limited to solving the problem of classification as such, but also includes pre- and post-processing of data: their ingestion, processing, storage (extract-transform-load), and also monitoring of the process. In addition, in modern conditions, anomaly detection usually occurs on large volumes of data: it can be gigabytes of input data, which, moreover, are received in real-time and require rapid processing. The ability of companies to quickly detect and respond to anomalies in large volumes of data flows is critical to success in the digital transformation environment. It allows companies to identify or even predict anomalous patterns in big data streams. For example, for telecommunications company customers, protecting their wireless networks from security threats is critical. Mobile data traffic is expected to reach 77.5 exabytes per month globally in 2023, with an average global growth rate of 46%

[3]. This growth in data increases the risk of attacks from unknown sources and prompts telecom customers to look for new methods and tools to detect threats.

Based on this, it is worth considering scientific and applied approaches to solving the problem of anomaly detection in conjunction with the formation and application of mathematical models (implemented, for example, in Python) in combination with the research of data processing methods.

LITERATURE ANALYSIS

The problems of big data processing to detect anomalies are presented in the work [4]. The work aims to formulate the main problems for solving the task of detecting anomalies in big data, to establish the current state of solutions in the field. Research has shown that existing approaches to network anomaly detection are not effective enough, especially for real-time detection. The reason for the ineffectiveness of current approaches is mainly due to the accumulation of huge amounts of data through connected devices. Therefore, it is imperative to propose a framework that efficiently processes big data in real-time and detects anomalies in networks. Taking this into account, this article attempts to consider the problem of detecting anomalies in real-time. But processing methods are represented insufficiently. The continuation of the research was reflected in the work [5]. The paper focuses on available methods for anomaly detection in big data processing. The work notes that many existing anomaly detection methods do not maintain sufficient accuracy due to so-called "big data", which is characterized by a large volume and high speed of data transmission from various sources. This phenomenon of having both problems together can be called the “curse of high dimensionality”, which affects existing methods in terms of both performance and accuracy. To address this gap and understand the underlying problem, it is necessary to identify the unique subproblems associated with anomaly detection for both high-dimensional problems and big data. The paper aims to document the state of anomaly detection in high-dimensional big data by presenting unique challenges using the triangular vertex model: the problem (high-dimensionality), methods/algorithms (anomaly detection), and tools (big data technologies). In addition, the limitations of traditional approaches and current high-dimensional data strategies are discussed, along with new big data processing techniques needed to optimize anomaly detection. But the work does not pay enough attention to the application of scientific research in practice. Hands-on application of big data processing in real-time mode is reflected in the work [6]. The article focuses on practical applications in the field of marketing. It notes that the collection of big data from various sources, such as the Internet of Things, social media, and search engines, has created significant opportunities for industrial marketing organizations to have an analytical perspective on developing programmatic marketing approaches for online advertising. Cleaning, processing, and analyzing such large data sets pose challenges for marketing organizations, especially for real-time decision-making and benchmarking. Using a problematization approach [7] contributes to exploring the connections between big data and real-time processing. This study subsequently covers relevant big data sources and efficient batch and real-time processing related to structured and unstructured datasets. The article develops interdisciplinary dialogues that include working with big data tools such as Apache Storm and Hadoop. Let’s list the tools, which can be handy for anomaly detection in big data processing:

1. **Apache Hadoop** [8] – the most popular and widely used Big Data framework on the market. Hadoop allows distributed processing of large data sets across clusters of computers. It is one of the best big data tools out there, scaling from a single server to tens of thousands of computers. Instead of storing and processing all the data on a single computer, Hadoop connects multiple computers in a scalable network and analyzes the data in parallel. This procedure typically uses the MapReduce programming model, which distributes remote computers to coordinate the processing of big data. It is an open-source platform based on Java. It can handle both structured and unstructured data. Hadoop also offers cross-platform support for its users. Today, it is the best big data analysis tool widely used by many tech giants like Amazon, Microsoft, IBM, etc.
2. **Apache Spark** [9] – a free and open-source software solution. It connects multiple computers and allows them to process big data in parallel, which speeds up and simplifies big data operations. Spark is gaining popularity thanks to the use of machine learning and other technologies that increase speed and efficiency. Spark comes with a set of tools that can be used for a variety of functions, including structured data and graph processing, Spark Streaming and machine learning analysis, and extensive APIs for Scala, Python, Java, and R. Both services can be used together or as

separate The main difference between Spark and MapReduce (used in Hadoop) is that Spark processes and stores data in memory for further steps, while MapReduce processes data on disk. As a result, for smaller workloads, Spark is 100x faster than MapReduce.

3. **Apache Cassandra** [10] – an open-source NoSQL distributed database used to store large amounts of data. It is one of the most popular data analysis tools and has been highly praised by many technology companies for its high scalability and availability without compromising on speed and performance. The tool is capable of performing thousands of operations every second and processing petabytes of resources with almost no downtime.

4. **Apache Storm** [11] – a reliable, user-friendly tool used for data analysis, especially in small companies. The best part about Storm is that it has no language barrier and can support any of the programming languages. Designed to handle large data pools in a fault-tolerant and horizontally scalable manner. When we talk about real-time data processing, Storm leads the way due to its distributed real-time big data processing system, due to which many tech giants use Apache Storm in their systems today. Some of the most famous names are Twitter, Zendesk, NaviSite, etc.

5. **Apache Kafka** [12] – a distributed event processing or streaming platform that allows applications to quickly process large amounts of data. It is capable of processing billions of events every day. It is a scalable streaming platform with excellent fault tolerance. The streaming process involves publishing and subscribing to streams of records in the same way that messaging systems do, archiving these records and analyzing them later.

6. **Apache Beam** [13] – a unified, open-source programming model for defining and executing data processing pipelines, including ETL, batch, and streaming (continuous) processing. Beam pipelines are defined using one of the provided SDKs and run on one of the supported Beam clusters (distributed processing systems), including Apache Spark.

The work [14] provides an example of creating an anomaly detection system on Amazon DynamoDB streams using Amazon SageMaker, AWS Glue, and AWS Lambda. The paper does not analyze other tools for detecting anomalies in big data. From the conducted analysis, it was concluded that the solution to the problem of detecting anomalies in data should be considered in a complex way as the formation and application of mathematical models (implemented, for example, in Python) in combination with the study of data processing approaches. It is relevant to analyze attributes from network logs and create a stream analytics pipeline to detect anomalies.

FORMULATION OF THE PROBLEM

The problem that this work tries to solve is that although there are already researches that reflect the features of big data processing, but it is appropriate to supplement existing solutions with scientific and applied approaches to processing large volumes of data, as an infrastructure for detecting anomalies in them.

THE AIM OF THE STUDY

The aim is to develop a data processing infrastructure model for detecting anomalies and present its practical implementation.

In accordance with the set goal, the following research tasks were formed:

– to form requirements for big data processing tools;

– to highlight the best practices of data processing, as well as the possibilities of their implementation with the help of available cloud technologies;

– to determine the requirements for the software infrastructure of streaming data when detecting anomalies in real-time;

– to form the basic recommended implementation of the real-time anomaly detection system architecture and to present improvements to the data processing approach.

RESEARCH RESULTS

Definition 1. Data anomaly detection infrastructure is a collection of models, methods, algorithms, tools, software solutions, and business processes necessary for the data anomaly detection process.

Conceptually, the infrastructure model for detecting anomalies in data is presented in the Fig. 1.

Software solutions for working with big data can be conventionally classified into: proprietary, open-sourced, supported by large companies, and developed by a group of enthusiasts.

Let's walk through the most effective solutions for streaming analysis of big data in real-time and give them a brief description of how this or that solution can facilitate the process of streaming big data in real-time to detect anomalies in them. But before that, let's form the requirements that must be satisfied:

- **Flexibility and scalability**: a basic requirement that must be met. Data processing tools

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

must work efficiently both on small amounts of data (on which training and hypotheses are made), and in the future adapt to ever-increasing amounts of data in real-world settings. In practice, it functions as the development of the company's innovative projects. Meeting this requirement contributes to the development of the company, in particular, at the early stage of the company's development, it allows you to gain a competitive advantage and meet the requirements during periods of rapid growth. In addition, this criterion is also necessary for the functioning of the system in real-time - the data stream is unstable and, for example, can fluctuate significantly depending on the time of day (the number of bank transactions).

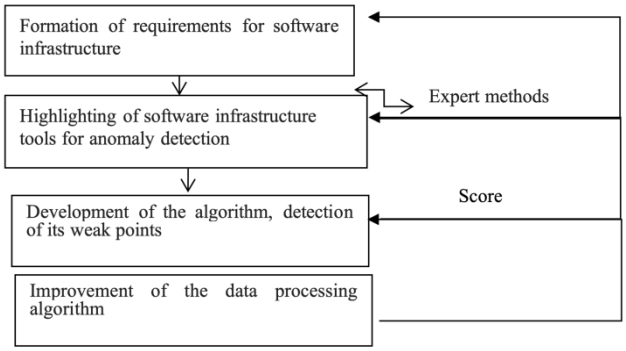

Fig. 1. **A conceptual model of the anomaly detection infrastructure in streaming data**
Source: **compiled by the authors**

- **High level of security**: the data may contain confidential information, so it is necessary to evaluate the level of security of the tools used to ensure that the most effective methods of protection (encryption, authorization, authentication, etc.) are applied to protect our data. Standard security protocols and procedures must also be established at all levels to limit which individuals or groups have access to the data.
- **Seamless integration**: when choosing a tool for big data, it is worth evaluating which solution is suitable – standalone (that is, one that works completely independently) or integrated (which is based to one extent or another on the existing infrastructure). Here it is worth taking into account such factors as: the source of data (the channel of their receipt), licensing, and laws of the country of use. Choosing even an autonomous solution that does not require integration now, is worth predicting possible further integrations that can be implemented in the future at the planning stage.
- **Compliance with business objectives**: the resulting solution should satisfy both current and future potential needs for streaming analysis of big data to detect anomalies in it. In our case, the tools must function effectively in terms of detecting anomalies.
- **Simple user interface**: tools should be user-friendly and adaptable to a wide range of users. Even non-technical people should be able to easily create and understand dashboards and reports. Although aesthetics may seem unimportant, having unattractive graphics on our panels will reduce user perception.
- **Advanced analytics**: a big data analytics tool should be able to identify patterns in data and predict future events and outcomes. It must go beyond simple mathematical calculations to provide context-relevant information that will enable the creation of sophisticated prediction algorithms.

Based on the identified requirements, an expert evaluation was conducted and each requirement was prioritized for software for streaming big data analysis and anomaly detection (Table 1). The motivation for such research was the need to formalize software requirements taking into account the business processes of highly loaded software products that use real-time input data analysis for their classification and detection of patterns, such as payment systems or road traffic control systems.

The results were placed on a 3-point scale. “-1” – not a critical requirement for a big data streaming system to detect anomalies in the system's program logs; 0 – does not matter; 1 – an important requirement for a streaming big data analysis system to detect anomalies in the system's program logs.

The results of the expert assessment were processed according to [15]. Let’s go through the requirements one by one and examine their importance for our practical example of system logs analysis and anomalies detection:

- Flexibility and scalability: 1 – usage rate is usually unstable, and the number of connected clients can vary from a few dozens to thousands, therefore the scalability and flexibility are extremely important to respond to the incoming data traffic.
- High level of security: 0 – the software and infrastructure are not to be exposed publicly and to be hosted inside VPC (Virtual Private Cloud), therefore it all will be encapsulated and the security level of a tool does not really matter.
- Seamless integration: -1 – the software for anomaly detection in system logs does not require complex integration: it reads plain simple text data

and stores the results in a data store; therefore any generic solution will be suitable.

- Compliance with business objectives: 1 – it is important to have the correct tool, which is specialized in the problem we solve. Not every big data processing tool will satisfy the need, therefore we must find the one, which is made for processing big data streams of incoming logs (text data) and perform classification of them.
- Simple user interface: -1 – not critical, secondary requirement since all the process is a back-end job and is run in the background.
- Advanced analytics: 0 – it is nice to have the built-in tool to analyze the processed data and see the results in a grouped way, although it does not affect the process directly, therefore is not a primary requirement.

The importance of the requirements for software to detect anomalies in streaming data in real-time using the example of a software log classification system for improving the cyber security of software is established. Accordingly, the principle of software selection is based on an established ranked number of software requirements.

Table 1. **Expert assessment of the requirements for data processing tools that must be met for effective detection of anomalies in streaming data in real-time using the example of a software log classification system to improve software cyber security**

Requirement	The importance of the requirement to detect anomalies in the streaming data of the program logs of the system
Flexibility and scalability	1
High level of security	0
Seamless integration	-1
Compliance with business objectives	1
Simple user interface	-1
Advanced analytics	0

Source: **compiled by the authors**

Let's look at the existing software solutions that fully or partially satisfy the mentioned requirements, and point out what can be helpful in our use case:

1. **Apache Hadoop**:

a. Provides a high level of security when using HTTP servers.

b. It includes a diverse set of Big Data tools and technologies that together create a robust ecosystem that meets a developer's analytical needs.

c. Fast and easy data processing thanks to distributed processing.

d. Offers fast access via HDFS (Hadoop Distributed File System).

e. Very flexible and can be easily implemented using MySQL and JSON.

2. **Apache Spark**:

a. Specially designed to work with real-time data streaming.

b. SQL queries, streaming data, and advanced analytics including machine learning are supported.

c. Flexible: It can run on Mesos, Kubernetes, or the cloud.

3. **Apache Cassandra**:

a. Flexibility of data storage: It supports all forms of data including structured, unstructured, and semi-structured, and allows users to modify it according to their needs.

b. Data distribution system: Easily distribute data by replicating data across multiple data centers.

c. Fault tolerance: If any node fails, it will be replaced without delay. The ability to duplicate data on several nodes contributes to high fault tolerance.

4. **Apache Kafka**:

a. It is easily scalable and there is no risk of downtime.

b. Can easily handle large volumes of data streams.

c. Kafka offers high throughput for both publishing and subscribing to messages.

5. **Apache Storm**:

a. Data processing: Storm processes data even if the node disconnects

b. Speed: Apache Storm's speed is impeccable and can handle up to 1 million 100-byte messages on a single node.

6. **Apache Beam**:

a. Unified – it is using a single programming model for both batch and streaming use cases, which simplifies the development process and maintenance.

b. Portable – it executes pipelines in multiple execution environments. Here, execution environments mean different runners. Ex. Spark Runner, Dataflow Runner, etc. That means that we can easily deploy it on cloud computing resources, e.g. GCP.

c. Extensible – it allows to write custom SDKs, IO connectors, and transformation libraries. We can extend or alter the existing implementation, which can be very helpful for testing different approaches to find the optimal one.

Based on the analysis of instrumental solutions,

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

it was established that for their functioning during data processing, power is required, as well as their constant availability and scalability of the computer resources themselves (RAM, processor power, etc.). For this, it is advisable to use cloud technologies - to use the capabilities of one of the providers for ease of management.

One of the most popular providers are Amazon (Amazon Web Services), Google (Google Cloud Platform), and Microsoft (Microsoft Azure).

In our example, we will consider Google Cloud Platform. With GCP, we can manage data throughout its lifecycle, from acquisition and storage to processing and visualization:

1. Sourcing stream input data is implemented using services such as Compute Engine or Kubernetes Engine.
2. Storage is possible thanks to Cloud Storage or Cloud SQL, which provide fast access to data.
3. Data processing is carried out by Cloud Dataflow or Dataproc tools.
4. Complex visualization is implemented using Cloud Datalab or Data Studio.

An important component of GCP is the ability to archive data, that is, store replicas on special secure servers, which will be useful if you need to have highly reliable backup copies.

Among other things, GCP is generally a cheaper alternative to other advanced cloud solutions – AWS and Azure, and also offers a free set of products for research purposes, so let's stop our attention on it.

Let us note the Google Cloud Platform products that satisfy our requirements for big data streaming tools to one degree or another [16]:

1. **Dataproc** – fully managed and highly scalable service to run Apache Spark, Apache Flink, Presto and has more than 30 open source tools and frameworks. Dataproc is used to modernize data storage, ETL, and secure data on a global scale, fully integrated with Google Cloud.
2. **Dataflow** – unified serverless streaming and batch data processing, fast and cost-effective.
3. **BigQuery** – serverless, highly scalable, and cost-effective multi-cloud data storage built for business agility.
4. **Vertex AI** – rapidly build, deploy, and scale ML models using pre-trained and customized tools within a unified AI platform.

The selection of **data processing tools for detecting anomalous behavior in logs to detect a cyber security threat** is implemented by an expert method depending on business processes. Based on the analysis of data processing tools and the formed requirements for detecting anomalies in streaming big data, an experimental part was conducted using Google Cloud Platform.

As a software infrastructure, the use of GCP cloud technologies from Google is proposed for detecting anomalies in large data sets, namely for streaming analysis of system software logs and detection among suspicious operations in real-time to improve system cyber security.

GCP tools enable advanced artificial intelligence (AI) and machine learning (AI/ML) capabilities combined with an enterprise-class streaming analytics platform. It's a combination of fast data and advanced real-time AI analytics. Therefore, the development of a model that would ensure such a process is of both scientific and practical interest.

DEVELOPMENT AND IMPROVEMENT OF THE ALGORITHM OF THE DATA PROCESSING INFRASTRUCTURE USING CLOUD TECHNOLOGIES (APPLICATION EXAMPLE)

From the model presented in Fig. 1 - the next step is to form the algorithm, identify its weak points and improve it.

The basis of the technique is the signature-based approach. A signature pattern examines network traffic by comparing it to signature stores derived from malicious objects. Although this technique works well for known threats, it is difficult to detect new attacks because there is no pattern or signature.

Synthetic data [15] were selected for the study. Consider the creation of a solution for detecting network anomalies based on machine learning with tool extraction (Fig. 2).

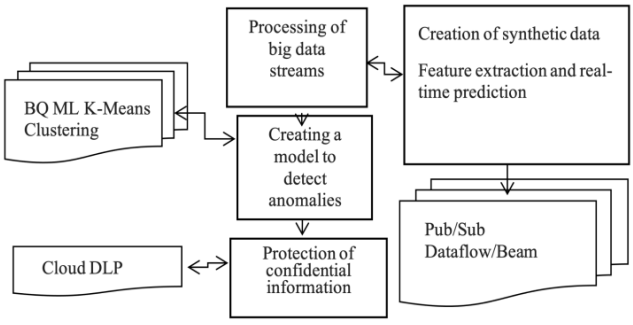

Fig. 2. **Highlighting of the tools for detecting network anomalies**
Source: **compiled by the authors**

For evaluation, we selected a big data processing algorithm, the results of which are presented in [15], and we will form an improvement of the data processing process using Apache Beam tools.

The sequence of the study is displayed below:

1. *Dataset formation.* Synthetic data were selected for the study. Conveyor publication of synthetic data at a speed of 250.000 messages/sec.

2. *Finding the subnet of the destination IP address, dstSubnet, and caller ID.* Implementation – using the Apache Beam transformation.

3. *Avoiding the storage of identification information* in the form of plain text in BigQuery with the help of Cloud DLP, for de-identification of numbers – IMSI. Deterministic encryption, where data can be de-identified (or tokenized) and re-identified (or de-tokenized), is appropriately implemented with CryptoKey.

4. *Minimizing the frequency of requests to the DLP cloud service.* Implementation – using a micro-batch approach, using the state API and the Apache Beam timer.

5. *Data training and normalization in BigQuery.* To process large volumes of daily data, it is proposed to use partitioned reception timetables and clustering. Storing data in a partitioned table allows you to quickly select training data using filters for a set time limit and a group of subscribers. It is proposed to use the k-means clustering algorithm [17, 18] in BigQuery ML to train the model and create clusters.

6. *Detection of emissions (anomalies).* To find anomalies, this step calculates how far the input vector is from the nearest centroid. If the distance is three standard deviations greater than the mean, the data points are anomalies in the BigQuery table.

7. *Query the anomaly table in BigQuery.* The result is a subscriber ID that is stored in a de-identified format. To get the original data in the Pub/Sub subscription, we use the data re-identification pipeline.

8. *Visualization of the functioning of the obtained anomaly detection pipeline in real-time.* To implement this stage, it is appropriate to use the Looker tool, which is part of the Google Cloud Platform, to visualize the functioning of the received anomaly detection pipeline in real-time.

From the conducted research, a flaw was found - the speed of processing results and obtaining data. Data processing speed is critical to the operation of a real-time anomaly detection system. Therefore, the question of improving the existing basic model arose.

The Dataflow solution automatically scales anomaly detection processes and allocates the optimal amount of computing resources, so we cannot influence this part (and do not need to). However, there remains the side of programmatic use of the Apache Beam tool itself, which can be used to rationally adapt the script to improve performance without changing the amount of resources. At its core, this task boils down to the efficient use of algorithms and data structures for data processing. Optimization using Apache Beam is offered with the execution of parallelization, manual management of allocated RAM, and sorting of input data.

Parallelization. To transform several attributes, a sequential linear transformation is usually used (Fig. 3).

As an alternative, it is proposed to use parallel transformations in Apache Beam, which, other things being equal, reduce complex calculations linearly proportional to the number of parallel calculations (provided that the parallel calculations are approximately the same in terms of required resources). In the study, the object (body) of the connection consists of 12 attributes, each of which is subject to processing (it can be such operations as grouping, aggregation, etc.). Theoretically, the creation of parallel transformations (Fig. 4) would speed up processing by 12 times. But it is worth noting that horizontal scaling is never accompanied by a factor of 1, as it requires additional processes that also require resources: creating new instances, copying objects for each individual transformation, etc.

Manual management of allocated RAM. This part deals with efficient interaction with RAM. For example, by default, the pipeline (and each of its individual transformations) expects attributes up to 512 bytes long and allocates the necessary space for them. Memory allocation logic (which is implemented internally in the tool) allocates the necessary memory based on the largest element among all elements of all groups (Fig. 5). In the first group of data, none of the elements occupy all the space that is used inefficiently. To do this, you can reduce the size of the first group and free up the memory allocated for it (Fig. 6).

Fig. 3. **Linear conveyor with sequential transformations**
Source: **Apache Beam Documentation [13]**

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

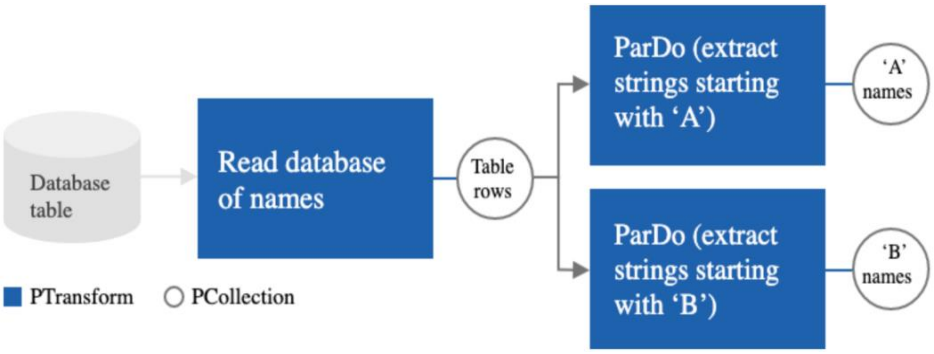

Fig. 4. **Parallel pipeline on the example of 2 transformations, which are simultaneously applied to the same collection**
Source: **Apache Beam Documentation [13]**

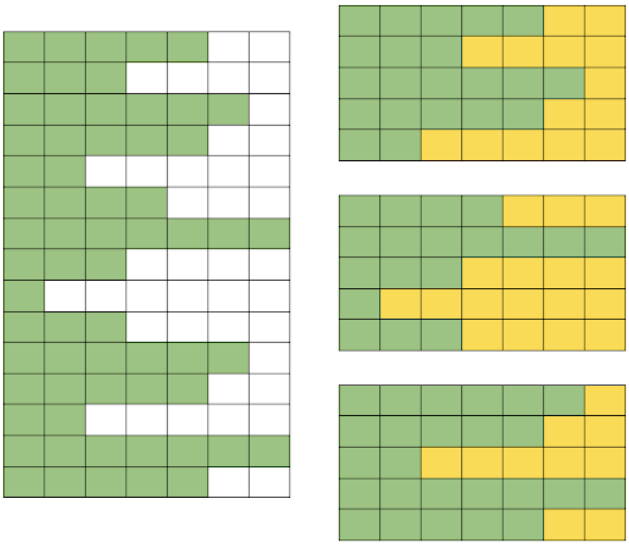

Fig. 5. **Visualization of allocated memory in automatic mode**
Source: **compiled by the authors**

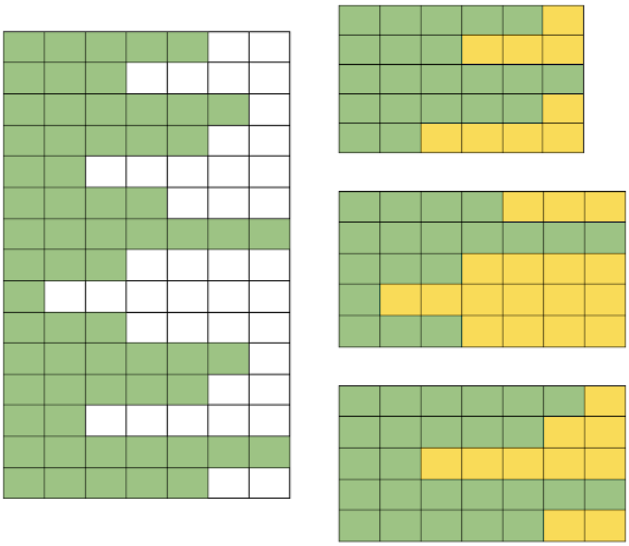

Fig. 6. **Visualization of allocated memory with manual dimension control**
Source: **compiled by the authors**

Sort input data. Figure 6 shows that there is a lot of unused space that could be freed up (yellow cells). This can be achieved by sorting the input data, which would allow memory to be allocated with minimal overhead. The sorting results are presented in Fig. 7.

An important feature of this method is that it takes less time to process a smaller group, so you can combine several groups in one process to achieve optimization of allocated resources. So, if, for example, we create 2 parallel processes, we can give the smallest and largest groups of data to one process, and to the other – data of medium volume. Thus, these processes will finish processing data at approximately the same time.

To visually demonstrate the result of optimizing the data processing using the above methods, let's create a new set of synthetic data and test its processing in 4 forms: non-optimized (basic) pipeline, pipeline with parallel calculations, pipeline with optimized work with memory, pipeline with parallel calculations and optimized work with memory (all together).

Obtained results:

– an unoptimized (basic) pipeline - 83 minutes 4 seconds;

– a pipeline with parallel calculations - 9 minutes 12 seconds;

– a pipeline with optimized work with memory - 82 minutes 15 seconds;

– a pipeline with parallel calculations and optimized work with memory - 9 minutes and 3 seconds.

The research found that the best result was obtained using parallel computing methods. This is explained by the fact that several processes occur simultaneously, and not sequentially. And the more processes we can run sequentially, the greater the effect we will observe. However, you should not forget that the resources provided for this are not free, and therefore you should still find a balance between the speed and the cost of the solution.

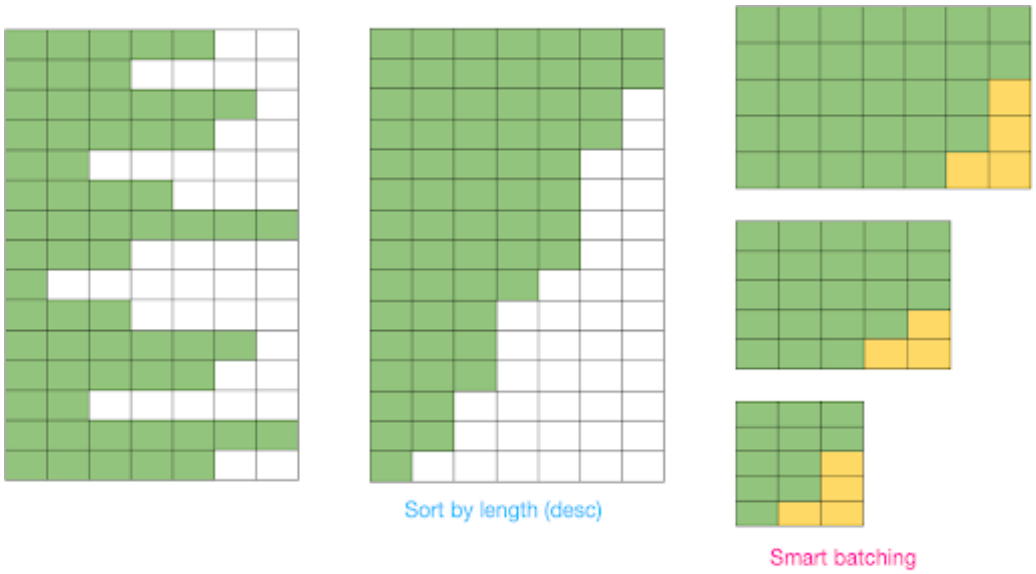

Fig. 7. **Sorting of input data and their division into groups**
Source: **compiled by the authors**

On the other hand, memory management does not provide such a significant increase in speed, but also does not require additional costs (except for the costs of directly developing the solution).

The received models and approaches can be applied in real computer systems to analyze system logs in real-time, especially via customizing the default GCP data processing pipelines and data lakes using external solutions, such as manually created data processing pipelines, or custom 3rd party solutions (Apache Beam), which can set up specifically per use case, what this work demonstrates.

CONCLUSIONS

The paper proposed the definition of infrastructure for detecting anomalies in data. The requirements for software-based detection of streaming data anomalies have been formulated. It is recommended to implement the selection using expert methods. Also, the work presents the results of the research on the expert selection of the software infrastructure for the streaming analysis of big data and the detection of anomalies among them in real-time. The existing software solutions for processing large volumes of data and their possible application for solving the problem of detecting anomalies in real-time are formulated based on the basis of the formed requirements. Such products under the Apache license as Hadoop, Spark, Cassandra, Kafka, Storm, and Beam are described. The article specified cloud technologies based on these solutions. They can be used as final tools for implementing the process of analyzing incoming data streams. The template of the solution architecture for detecting anomalies in real-time is provided, which is based on these examples. It is a service for streaming analysis of system program logs and detection of suspicious operations among them to improve the cyber security of software.

The work determines that real-time anomaly detection artificial intelligence models have the greatest impact. Also, it highlights anomaly detection tools that allow you to create a secure real-time anomaly detection solution using Dataflow, BigQuery ML, and Cloud DLP. It has been established that anomaly detection using a strictly-defined probability distribution may not be entirely accurate, further analysis is important to confidently identify any security risks.

The article provides a basic recommended implementation of the real-time anomaly detection system architecture using GCP and Apache Beam. Also, it demonstrates possible improvements to the base model to improve its speed. The peculiarity of the obtained result is that it provides and analyzes possible improvements in data processing using the capabilities of Apache Beam, which is deployed using cloud technologies – Google Cloud Platform. The work takes the example of applying big data processing infrastructure to detect anomalies in incoming data streams of system logs, showing the particularities of applying Apache Beam and Google Cloud Platform tools to modify, extend and improve the existing data processing pipelines.

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

REFERENCES

1. Chandola, V., Banerjee, A. & Kumar, V. "Anomaly detection: A survey". *ACM Computing Surveys.* https://www.scopus.com/authid/detail.uri?authorId=18435840500&origin=recordPage. 2009; 41 (3): 1. DOI: https://doi.org/10.1145/1541880.1541882.

2. Niu, Z., Shi, S., Sun, J., & He X.. "A survey of outlier detection methodologies and their applications". *Artificial Intelligence and Computational Intelligence. Lecture Notes in Computer Science*. https://www.scopus.com/authid/detail.uri?authorId=7101688915&origin=recordPage. 2011; 7002: 380–387. DOI: https://doi.org/10.1007/978-3-642-23881-9_50.

3. Anupriya, Singhrova, A. "Mobile traffic flow prediction using intelligent whale optimization algorithm". *Autom Softw Eng.* https://www.scopus.com/authid/detail.uri?authorId=57211458726& origin=recordPage. 2022; 29 (48): DOI: https://doi.org/10.1007/s10515-022-00349-7.

4. Ariyaluran Habeeb, R. A., Nasaruddin, F., Gani, A., Targio Hashem, I. A., Ahmed, E. & Imran, M. "Real-time big data processing for anomaly detection". *International Journal of Information Management.* https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57203807223&zone. 2019; 45: 289–307. DOI: https://doi.org/10.1016/j.ijinfomgt.2018.08.006.

5. Thudumu, S., Branch, P., Jin, J. et al. "A comprehensive survey of anomaly detection techniques for high dimensional big data". *J Big Data.* https://www.scopus.com/authid/detail.uri?origin=resultslist& authorId=57193405438&zone=. 2020; 7: 42. DOI: https://doi.org/10.1186/s40537-020-00320-x.

6. Jabbar, A., Akhtar, P. & Dani, S. "Real-time big data processing for instantaneous marketing decisions. *A problematization approach*". *Industrial Marketing Management.* https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=55397473800&zone. 2020; 90: 558–569. DOI: https://doi.org/10.1016/j.indmarman.2019.09.001.

7. Komleva, N. O., Liubchenko, V. V. & Zinovatna S. L. "Methodology of information monitoring and diagnostics of objects represented by quantitative estimates based on cluster analysis". *Applied Aspects of Information Technology. Publ. Nauka i Tekhnika.* Odessa: Ukraine. 2020; 3 (1): 376–392. DOI: https://doi.org/10.15276/aait.01.2020.1.

8. "HADOOP". – Available from: https://cwiki.apache.org/confluence/collector/pages.action? key=HADOOP. – [Accessed: Nov. 2021].

9. "Unified engine for large-scale data analytics". – Available from: https://spark.apache.org/. – [Accessed: Nov. 2021].

10. "Welcome to Apache Cassandra's documentation". – Available from: https://cassandra.apache.org/doc/latest. – [Accessed: Nov. 2021].

11. "Documentation Storm". – Available from: https://storm.apache.org/releases/2.4.0/index.html.

12. "DOCUMENTATION Kafka 3.2 Documentation". – Available from: https://kafka.apache.org/documentation. – [Accessed: Nov. 2021].

13. "Apache Beam Documentation". – Available from: https://beam.apache.org/documentation. – [Accessed: Nov. 2021].

14. Khlevna, Iu. L. & Koval, B. S. "Development of the Automated Fraud Detection System Concept in Payment Systems". *Applied Aspects of Information Technology. Publ. Nauka i Tekhnika.* Odessa: Ukraine. 2021; 4 (1): 37–46. DOI: https://doi.org/10.15276/aait.01.2021.3.

15. Khlevna, Iu. "Expert method of forming the information space of project management meta-methodolog". *Management of the development of complex systems.* 2018; 35: 61–67.

16. "Why google cloud". – Available from: https://cloud.google.com/why-google-cloud. – [Accessed: Nov. 2021].

17. "Anomaly detection using streaming analytics & AI". – Available from: https://cloud.google.com/blog/products/data-analytics/anomaly-detection-using-streaming-analytics-and-ai. – [Accessed: Nov. 2021].

18. Manal A. “Big data mining. A classification perspective”. 2016. DOI: https://doi.org/10.1201/9781315375083-97.

Conflicts of Interest: the authors declare no conflict of interest

Received 10.11.2022
Received after revision 15.12.2022
Accepted 23.12 .2022

DOI: https://doi.org/10.15276/aait.05.2022.23
УДК 004.94

Розробка інфраструктури виявлення аномалій у наборах великих даних

Хлевна Юлія Леонідівна[1)]
ORCID:http://orcid.org/0000-0002-1874-196; yuliia.khlevna@knu.ua. Scopus Author ID: 57191869873
Коваль Богдан Сергійович[1)]
ORCID: http://orcid.org/0000-0002-3757-0221; bohkoval@gmail.com. Scopus Author ID: 57200141737
[1)] Київський національний університет імені Тараса Шевченка, вул. Володимирська, 60. Київ, 01033, Україна

АНОТАЦІЯ

У роботі представлено аналіз моделей, методів і технологій виявлення аномалій у даних. Зроблено висновок, що на основі проведеного аналізу рішення проблеми виявлення аномалій у даних слід розглядати як комплексну технологію, яка складається з формування та застосування математичних моделей у поєднанні з дослідженням підходів до обробки даних. У статті проаналізовано сучасний стан технологій обробки потоків великих даних та відображено особливості найбільш поширених і прогресивних з них, напр. Apache Hadoop, Apache Spark, Apache Cassandra, Apache Kafka, Apache Storm і Apache Beam. Окрім цього, увага приділяється інфраструктурі, у якій створені моделі програмного забезпечення можуть бути розгорнуті та використані, беручи до уваги високий характер даних у режимі реального часу. У статті запропоновано сформувати інфраструктуру для виявлення аномалій у даних як прикладний приклад хмарної інфраструктури обробки великих даних. У роботі продемонстровано розроблену модель інфраструктури для виявлення аномалій у потокових даних реального часу, яка базується на експертному методі формування вимог до програмної складової, вибору алгоритму виявлення аномалій, вибору інструментів та удосконалення алгоритму. Виділені інструменти виявлення аномалій дозволяють створити безпечне рішення для виявлення аномалій у реальному часі за допомогою Dataflow, BigQuery ML і Cloud DLP. У статті представлено прикладну реалізацію виявлення аномалій у режимі реального часу за допомогою GCP та Apache Beam – аналіз потоку даних програмних журналів в інформаційній системі та виявлення серед них шахрайських, що допоможе підвищити кібербезпеку системи. Робота демонструє можливі вдосконалення базової моделі, які можуть допомогти її прискорити.

Ключові слова: великі дані; виявлення аномалій; хмарні обчислення; обробка даних; споживання даних

ABOUT THE AUTHORS

Iuliia L. Khlevna - Doctor of Engineering Sciences, Associate Professor, Professor of the Department of Technologies Management, Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str. Kyiv, 01033, Ukraine
ORCID:http://orcid.org/0000-0002-1874-196; yuliia.khlevna@knu.ua. Scopus Author ID: 57191869873
Research field: Project management methodology; information technology in management; intelligent information technologies; data science

Хлевна Юлія Леонідівна - доктор технічних наук, доцент кафедри Технологій управління, Київський національний університет імені Тараса Шевченка, вул. Володимирська, 60. Київ, 01033, Україна

Bohdan S. Koval - postgraduate Department of Technology Management, Taras Shevchenko National University of Kyiv. 60, Volodymyrska Str. Kyiv, 01033, Ukraine
ORCID: http://orcid.org/0000-0002-3757-0221; bohkoval@gmail.com. Scopus Author ID: 57200141737
Research field: Information technology in management; intelligent information technologies; data science, machine learning

Коваль Богдан Сергійович - аспірант кафедри Технологій управління, Київський національний університет імені Тараса Шевченка, вул. Володимирська, 60. Київ, 01033, Україна