UDC 004.273

O. Bondar, O. Lisovychenko

DATA ANALYSIS USING MICROSERVICES TO SOLVE FORECASTING PROBLEM

Abstract: This article considers the solution to the problem of increasing the efficiency and reliability of data analysis systems with the involvement of microservices to solve forecasting problems. The article analyzes the problem of forecasting data, for what purposes it is used, what are the methods and approaches to solve this problem. No less special attention is paid to the description of microservice architecture, the advantages and disadvantages of microservices in comparison with a monolithic architecture, the general structure of the data analysis system using microservices, methods to increase the reliability and efficiency of such systems, the existing data analysis systems based on microservices.

Keywords: data analysis, neural network, microservices, microservice approach, system architecture, forecasting.

Introduction

Data forecasting is an integral part of various areas of human activity, such as science, economics, and manufacturing.

To date, there are a large number of methods used for forecasting. Among them are the group method of data handling (GMDH), regression analysis, neural network models, and others. It is known that solving forecasting problems requires a thorough study and analysis of the input data that will be used to create models.

Due to the rapid development of information technology, there has been a sharp increase in the amount of data, so for their analysis are created automated systems that significantly accelerate this process. However, such systems are quite cumbersome and inoperable, as they require increased resources (vertical scaling) for stable operation, which are exhaustible.

Therefore, in recent years to build data automation systems began to use microservice architecture. This architecture involves the division of the system into independent services – microservices. This approach allows you to have several services that perform their tasks, allow you to use different technologies and data warehouses written in different programming languages. Data transfer between services takes place using the HTTP protocol, notifications, or event-driven. All services can be deployed and managed using special systems.

Formulation of the problem

It is necessary to consider the problem of forecasting, to consider microservice architecture: to make a comparative description of monolithic and microservice architectures, to find advantages and disadvantages of microservices, to analyze existing systems built on the basis of microservices.

Also, it is necessary to offer methods and technologies to increase the efficiency and reliability of data analysis systems to solve forecasting problems using neural network models with the involvement of microservice technologies.

Solving the problem

Data forecasting is one of the types of Data Mining tasks. As a result of solving forecasting problems, the risk of making wrong decisions is reduced, the efficiency of management of various spheres of human activity is increased. Neural network models, regression analysis, or GMDH are commonly used to predict data.

Regression analysis is a method of determining the separate and joint influence of factors on the performance trait and quantifying this influence by using appropriate criteria. Regression analysis is performed on the basis of the constructed regression equation and determines the contribution of each independent variable to the variation of the studied (predicted) dependent variable. The main task of regression analysis is to determine the influence of factors on the performance indicator (in absolute terms) [1].

GMDH is a method used for in-depth learning, forecasting, optimization, and pattern recognition. GMDH inductive methods make it possible to automatically find relationships in the data, select the optimal model or network structure, and increase the accuracy of existing algorithms. This method contains a set of several algorithms for solving various problems. It consists of parametric algorithms, clustering, probabilistic algorithms, etc [2].

A neural network is a model that can use not only the algorithms chosen for it, but also has the opportunity to learn with new data. Data prediction using neural networks has recently become quite popular and widespread. This popularity is caused, first of all, by the high accuracy of this technology, and the neural network allows you to predict any type of data. Thus, the further description of the solution of forecasting problems will include the use of neural networks [3].

Solving forecasting problems involving neural networks is divided into the following stages: 1. creation or preparation (analysis) of data; 2. choice of network topology; 3. network training; 4. The work of the model – data forecasting.

It should be noted that data analysis is one of the main stages, the purpose of which is to find the relationship between the data and remove incorrect data from the set. This stage is used in all methods to solve forecasting problems. The correctness of the results of this stage depends on the correctness of the forecasting in general.

Data analysis using microservices

In order to automate the process of data analysis, appropriate systems are created. Such systems perform analysis, automate the choice of neural network topology, to create a final model for solving the problem of data prediction. However, the existing automated data analysis systems, unfortunately, are not viable and stable, with a significant increase in data volumes, as they are built on a monolithic architecture. Based on this, have recently begun to implement microservice architecture not only in enterprises and businesses but also in the field of data analysis.

Microservice architecture is one of the types of system architecture, which consists in dividing the system into separate independent parts (microservices) that perform different tasks and communicate with each other usually using the HTTP protocol [4].

To prove the efficiency and reliability of microservices, a comparative characteristic of monolithic and microservice architectures is given (Table 1).

Table 1

Comparative characteristics of monolithic and microservice architectures

	Monolithic architecture	**Microservice architecture**
Language programming	Uses one programming language entirely	Every service can be written in different languages
Scaling	Scaling is difficult because it is required scale the whole system	Scaling is simple, it is possible to scale each service separately
Support	Hard to support for cumbersome account systems	Easy enough to maintain
Introduction new functional	Heavy and slow introduction of a new one functional	Easy and fast introduction new functionality
Deployment systems	Slow deployment for large code base account	It is possible to deploy only certain services from all over systems
Data model	One data model for the whole system	Each service can contain its own data model
Occurrence errors in codebase	Errors fail operation of the entire system	An error in one of the services will not affect the operation of other system functionality

Also, a study was conducted on the productivity of solving problems of microservice architecture and monolithic. The results are shown in fig. 1 – comparative graph of system performance.

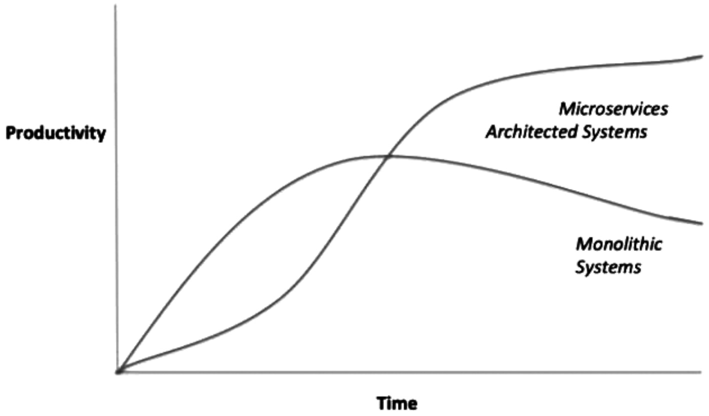


Figure 1. Comparative graph of productivity of monolithic and microservice architectures

Obviously, the microservice architecture has many advantages, however, despite this, there are also disadvantages in this approach to systems architecture:

– complexity during system deployment, as it requires additional settings for each service;

– there is a possibility of network delays;

– it is more difficult to ensure fault tolerance and balancing of the system.

The above shortcomings are significant, but with the correct design of the microser vice architecture, the occurrence of such shortcomings is quite small. Accordingly, to address these shortcomings, there are some approaches:

–services should be light enough and at the same time perform their main tasks;

–to avoid the complexity of the deployment, you need to use systems for orchestration of services (OpenShift, Kubernetes);

–all services must be able to scale;

–when building microservices it is necessary to take into account service delays or their failure.

In the general case, the microservice architecture for data analysis systems can be represented as follows (Fig. 2).

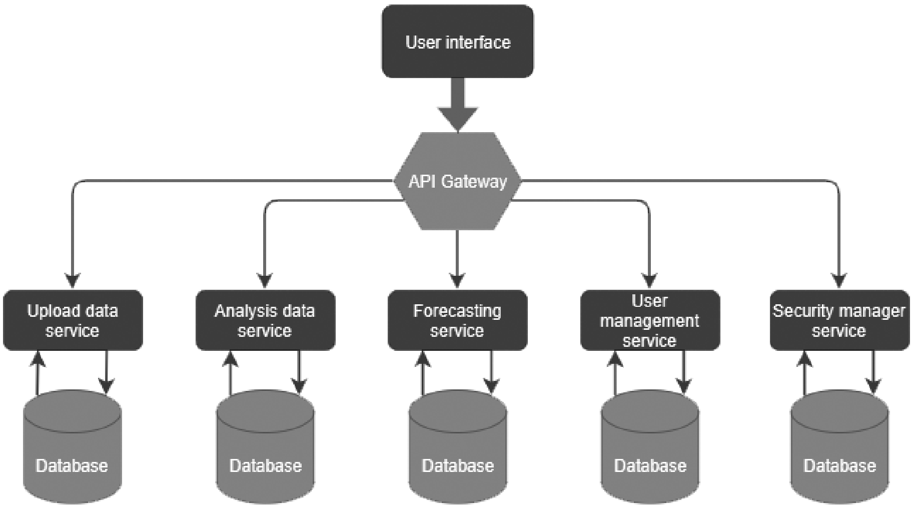


Figure 2. General view of microservice architecture for data analysis systems

Interaction with the system takes place using a special service – API Gateway, which, depending on the request to the system redirects this request to the appropriate service. To use the system, you need to enter it by authorization. The Security manager service is responsible for the authentication and authorization work in the system, and the User management service is responsible for creating a new user or changing an existing one. Upload data service – a service used to upload and convert data to the system. After downloading the data, this data is processed and analyzed using the appropriate service – Analysis data. The results of the Analysis data service can be used by the service to perform the forecasting – Forecasting service.

Based on the presented scheme, the question of the stability of this system may arise. The system is accessible to more than one user, so when a large amount of data is downloaded, especially when further analysis of this data and forecasting, the system may crash or slow down. In monolithic architecture, this problem is tried to solve by parallelizing the tasks of data analysis and forecasting. However, this approach works well when using large capacities.

For fault tolerance of the service/services in the microservice architecture of the analysis system, it is necessary to add an additional service called Load Balancer. Its main task is to distribute the load on the instances of the same service (Fig. 3). Using Load Balancer allows you to reduce the load on services that analyze and forecast data: Analysis data service and Forecasting service.

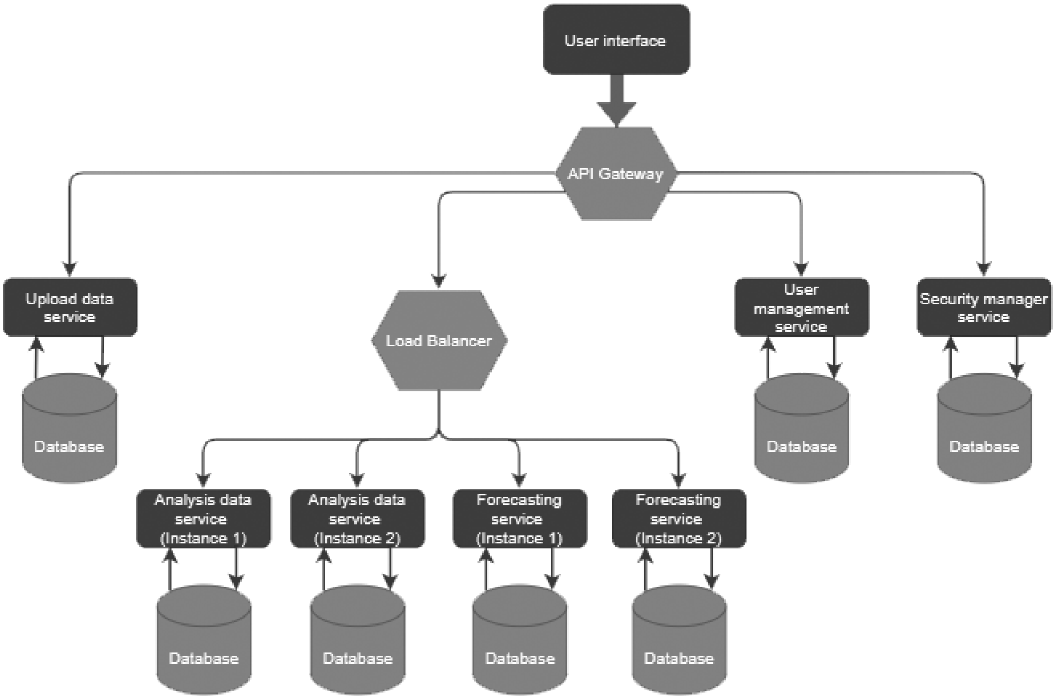


Figure 3. Example of microservice architecture using load balancing service – Load Balancer

Building a system using microservices will allow you to use different technologies to solve problems. Therefore, Python forecasting language and auxiliary libraries are best suited to create analysis and forecasting services today: Keras, Pandas, and others. Java with the Spring framework is best for creating data download, user, and security services. Because the data being downloaded may have a different format, it is best to store this data in a NoSQL database, in other tasks – regular SQL databases.

Also, microservices make it possible to easily and quickly introduce additional functionality, which demonstrates the flexibility of such an architecture. For example, it is possible to integrate work with the product Jupyter Lab, which will allow you to analyze the forecast data, the ability to create scripts for additional functionality, etc.

Existing systems for data analysis using microservices

To date, there are few systems for analyzing data based on microservices. An example of one such system is the Qunxian platform. This platform is designed for large data processing and further forecasting. With it, it is possible to use the total computing power of the server. The microservice approach in building the Qunxian platform allows you to easily and quickly add third-party systems to automate data analysis tasks, there is the possibility of parallel computing using Apache Spark. JupyterHub and a visual modeling system are used as the two main microservices. Qunxian is deployed using Docker technology.

The general microservice architecture of the Qunxian platform is shown in fig. 4.

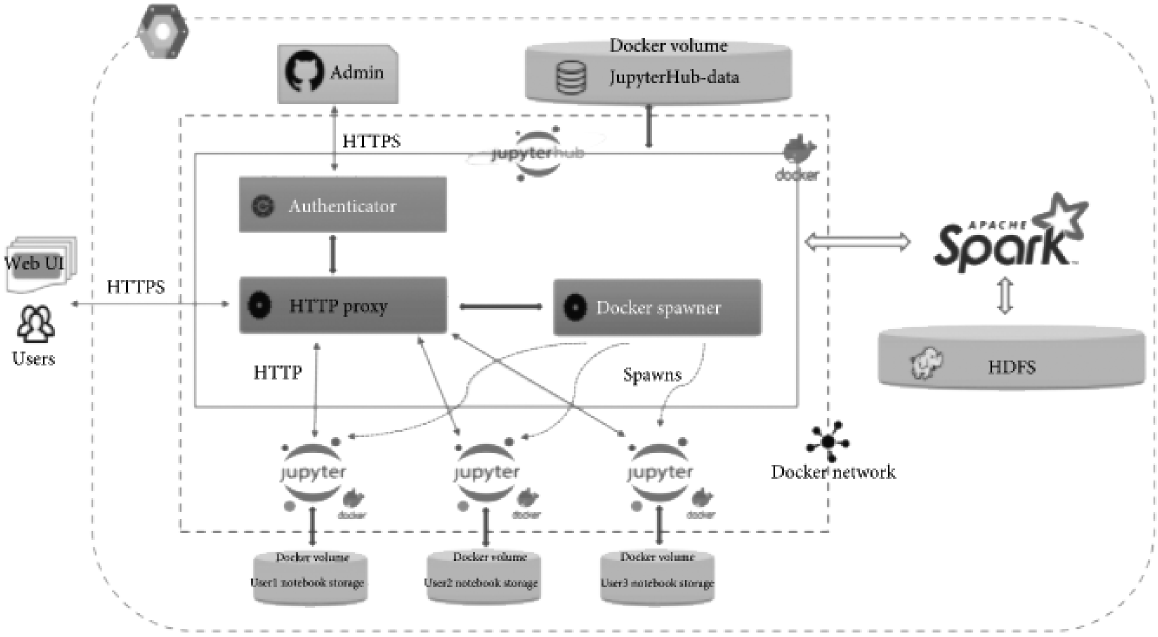
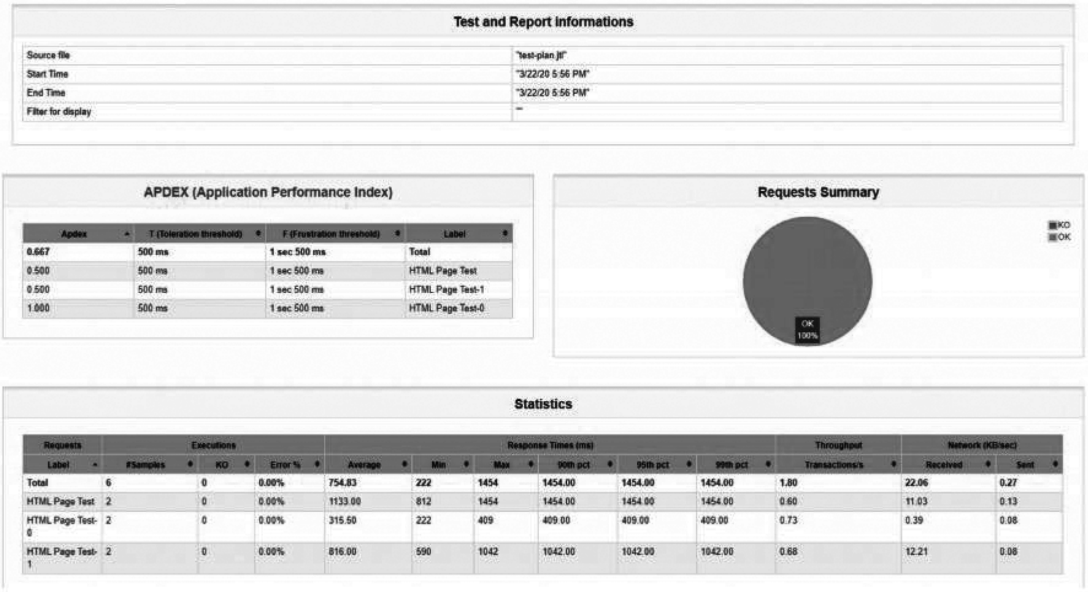


Figure 4. General microservice architecture of the Qunxian platform

This platform consists of four levels: 1. Hardware; 2. Resource management; 3. Data; 4. Platforms. The hardware layer uses Google Cloud Engine cloud computing technology, which allows you to easily scale the system, make it more flexible, have better and more secure data storage. Docker technology is used to deploy platform components on the cloud service. The level of resource management is built using Docker Swarm technology, which allows the orchestration of Docker containers. At the data level, HDFS (Hadoop Distributed File System), a technology for storing files on various data servers, and Greenplum, a parallel PostgreSQL-based database for working with large data sets, are being deployed. HDFS and Greenplum work with Spark, a framework for working with unstructured analytics. The platform layer represents two parts of the system – server and client. The server part contains JupyterHub technology, which allows you to do calculations and consists of two applications: for online programming and visual modeling. All work with the system takes place through the client part, using a graphical interface [5-8].

Let's analyze the test operation of the platform at different loads. The results are shown in fig. 5.



Test and Report informations

Source file	"test-plan.jtl"
Start Time	"3/22/20 5:56 PM"
End Time	"3/22/20 5:56 PM"
Filter for display	""

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.667	500 ms	1 sec 500 ms	Total
0.500	500 ms	1 sec 500 ms	HTML Page Test
0.500	500 ms	1 sec 500 ms	HTML Page Test-1
1.000	500 ms	1 sec 500 ms	HTML Page Test-0

Requests Summary

Statistics

Requests	Executions			Response Times (ms)						Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	6	0	0.00%	754.83	222	1454	1454.00	1454.00	1454.00	1.80	22.06	0.27
HTML Page Test	2	0	0.00%	1133.00	812	1454	1454.00	1454.00	1454.00	0.60	11.03	0.13
HTML Page Test-0	2	0	0.00%	315.50	222	409	409.00	409.00	409.00	0.73	0.39	0.08
HTML Page Test-1	2	0	0.00%	816.00	590	1042	1042.00	1042.00	1042.00	0.68	12.21	0.08

Figure 5. Test the operation of the Qunxian platform under load

Analysis of the result confirms that the requests sent to the platform were completed successfully and fairly quickly, quantitative indicators provide an opportunity to show the reliability and speed of the system built on the basis of microservices.

Another example of the use of microservices is the system described in the article [9]. This system solves the problem of self-adaptation of cloud and virtual infrastructure through forecasting. The authors consider the microservice architecture of the system, which, depending on the number of requests and the corresponding load, requires specialists to manually scale, which is desirable to do in advance so as not to stop the services. However, manual scaling is not advisable, because, with a large load on different services, it is impossible for professionals to track all the moments when it is worth scaling services. Based on this, as well as taking into account the use of microservices in the system, the authors decided to create an additional service for forecasting. The architecture of the described service is shown in fig. 6.

The user interface of the service allows you to easily adjust various parameters, including timestamps, and also allows you to download and perform predictions with your own data or check the service using the provided sample data. Data loading is achieved using the Data Volume component. Scripts in the R programming language using user parameters are used to run forecasting scripts. The predicted results are stored using the InfluxDB database, and the time interval estimate and other data are saved by the MongoDB database.

It is obvious that the use of microservice architecture for the system described in the article [9], allowed not only to create the above additional service for forecasting, but also to leave and not change the existing functionality, which in turn proves the flexibility of microservices.

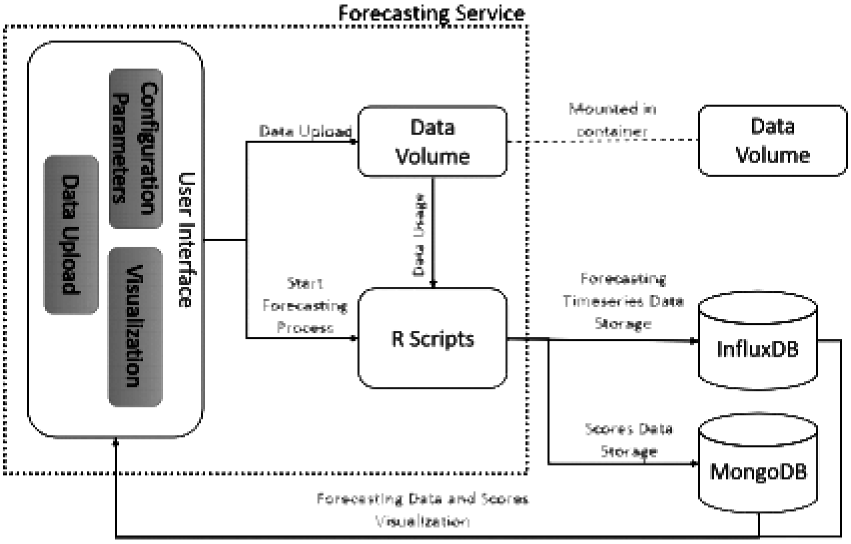


Figure 6. Forecasting service architecture for cloud
and virtual infrastructures self-adaptation

Conclusions

Existing data analysis systems for forecasting problems are rather slow and unreliable, especially with a large amount of information.

The article considers the stages of creating models for solving forecasting problems using neural network models and involving microservices in the construction of data analysis systems. The comparative characteristic of monolithic and microservice architectures is given. The advantages and disadvantages of microservices are analyzed and presented. The use of certain methods and technologies in the construction of fault-tolerant microservice architecture is demonstrated. The existing systems of automation of data analysis tasks, which are built on the basis of microservices, are analyzed.

It is proved that the proposed methods and technologies for building data analysis systems using models of neural networks with the involvement of microservices will increase the reliability and efficiency of the developed systems.

REFERENCES

1. Regression analysis URL: https://pidru4niki.com/17280924/ekonomika/regresiyniy_analiz (date of use: 03.10.2021)

2. GNDH URL: http://www.gmdh.net/ (date of use: 03.10.2021)

3. Bondar O. / Review and analysis one of the approaches to training the neural network model / O. Bondar, O. Lisovychenko // Interdepartmental scientific technical journal «Adaptive systems of automatic control». 2021. № 2 (37). C.

4. Fowler M., Lewis J. Microservices URL: http://martinfowler.com/articles/microservices.html (date of use: 24.09.2021)

5. HDFS Architecture Guide URL: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html#Introduction (date of use: 25.09.2021)

6. Greenplum Database URL: https://greenplum.org/ (date of use: 25.09.2021)

7. Apache Spark URL: https://spark.apache.org/ (date of use: 25.09.2021)

8. A Microservice-Based Big Data Analysis Platform for Online Educational Applications URL: https://www.hindawi.com/journals/sp/2020/6929750/ (date of use: 25.09.2021)

9. Forecasting Models for Self-Adaptive Cloud Applications: A Comparative Study URL: https://www.researchgate.net/publication/327545263_Forecasting_Models_for_Self-Adaptive_Cloud_Applications_A_Comparative_Study (date of use: 25.09.2021)