

DATA PROCESSING AND VISUALIZATION FOR CORROSION DETECTION

Roman Mysiuk¹, Volodymyr Yuzevych^{1,2}, Iryna Mysiuk¹

¹*Ivan Franko National University of Lviv,
50 Drahomanova St., UA-79005 Lviv, Ukraine
mysiukr@ukr.net*

²*Karpenko Physico-mechanical Institute of the NAS of Ukraine,
5 Naukova St., UA-79601 Lviv, Ukraine*

Ukraine is one of the countries through which many gases and oil underground metal elements of constructions pass. Under the influence of external and internal factors, the metal is subject to corrosion and other types of defects. One of the biggest issues is detecting defects in underground metal elements of constructions. The main reason why we should explore the possibility of rapid detection of such problems is the possibility of accidents that can lead to man-made disasters and dangerous emissions into the air and land. It is proposed to use the processing of current and potential measurements to detect corrosion. The approach of development of client-server architecture of the software with visualization of results in the web application is used in Amazon Web Service cloud environment. There is deployed Elasticsearch for data storage. Data processing is automated and will detect corrosion based on visualization of results. The system can replace manual work on data collecting, processing, and visualization in real-time.

Keywords: data processing, data visualization, underground metal elements of constructions, a system for collection data, cloud technologies.

Overview

Nowadays, the main goal of IT industry is automating all manual tasks for people. There are different ways to detect defects or corrosion [2]. In this case, we can simplify the model for collection and processing data for detecting defects. From devices polarizing potential meter with positioning module (VPP-M) and contactless current meter with positioning module (BVC-M) gathered a lot of typical unprocessed data. There are a lot of complex monitoring defects in underground software, but it is worth considering a simplifying system that can be suitable with existing devices for measurement currents and polarization potentials. Data for processing from two locations, namely underground main gas constructions [1].

There is a desktop program Reader VPP for collecting information from VPP device using RS-232 interface. It requires manual parsing of data and using a personal computer for data processing [1].

The system contains web service and web application. Web application used for visualization data structure using received data from web service it helps to show changes in real-time. Web service relates to database where data is stored. This client-server software architecture allows the possibility to handle requests and responses from existing data [7]. In

the future, it is proposed to use real data collected in microcontroller or minicomputer instead of mocked existing data. From the device point of view, Internet connection can be provided by using GSM modem [15]. Main HTTP methods are the main approach for communication between different layers of the system. Data obtained from a different location can be processed quickly using existing application programming interfaces (API) from the database. In Elasticsearch the JSON file this data can be pushed in a parallel way. The data ingestion process is used for uploading in the specified format to database directly [16]. It is crucial to visualize results from stored data. The web application allows displaying of gathered information in a different representation. Charts and hierarchical graphs help to understand and simplify the process of defect analysis. The model of the suggested system of data processing and visualization is shown in Figure 1.

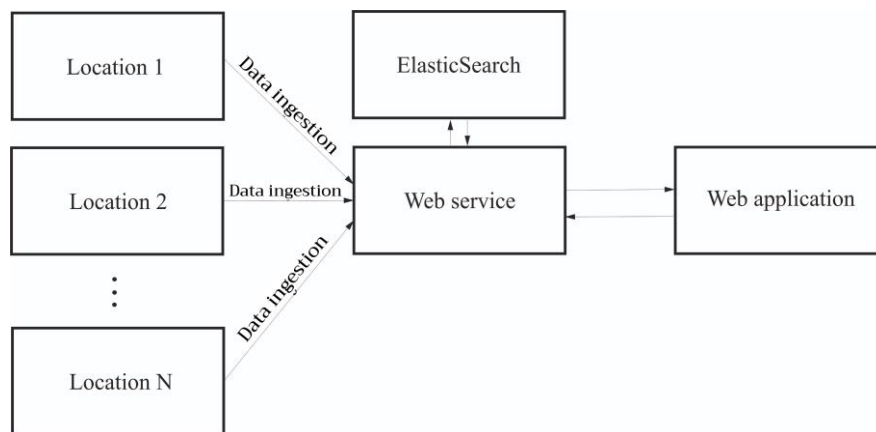


Fig. 1. A model that represents a structure of a system for data processing and visualization.

There are many popular models of parallel and distributed processing in the big data architecture, including Message Passing Interface, General-purpose computing, MapReduce, and others. There are also cloud computing and cloud architecture used to work with data in this system [18].

For testing purpose, there is used Amazon Web Service. This platform is the most popular cloud environment for deployment applications [19]. There are a bunch of useful tools based on cloud technology to build the infrastructure for configuration, security, network tracking, etc. It is a good approach to use virtual physical resources instead of the real ones. The application and database were configured and hosted for public access. There is selected the instance with Linux operation system and t2.micro instance type [3].

There are the following ways to connect to the instance:

- EC2 Connect Console;
- session Manager;
- cryptographic network protocol Secure Shell (SSH);
- huffy application, a free SSH client for Windows;
- Windows Subsystem for Linux (WSL).

In Figure 2 the web interface for instance setting is displayed. The status of instance, instance type public and private IPv4, and unique machine value instance ID are the general

information that can be reviewed on the main page. There are several tabs at the bottom to configure security, networking, storage, monitoring tags, and status check [4].

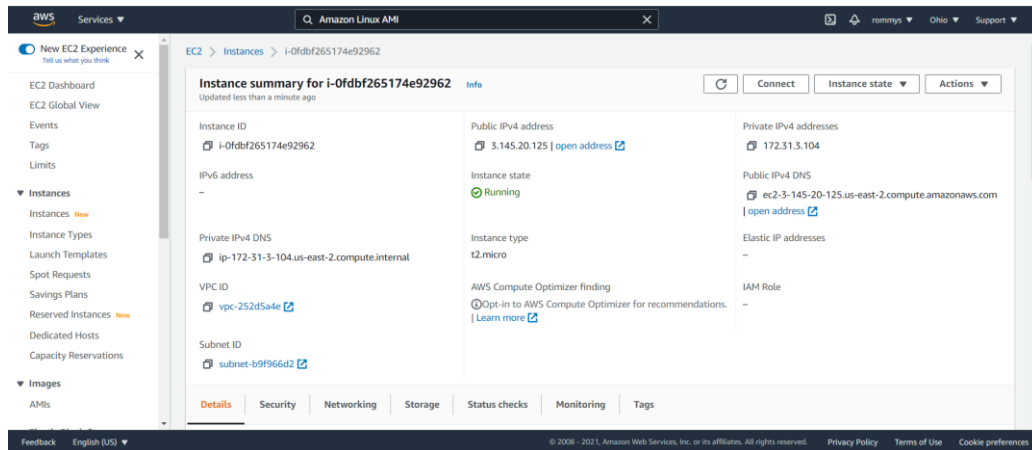


Fig. 2. Configuration of the testing instance

Java and Elasticsearch should be installed to start running the system. After changing network host parameter in elasticsearch.yml configuration file to 0.0.0.0/0 in an EC2 Connect Console allows using localhost instead of a long URL [4]. There is provided public access to Elasticsearch outside of the EC2 instance for testing purposes. There is an anticipation that data will be covered in a specified JSON file and sent to the data warehouse through web service. In Figure 3 status code 201 at the end of each item indicate that it is pushed successfully, and the record is created.

```
[ec2-user@ip-172-31-3-104 ~]$ curl -XPOST localhost:9200/measurements_location_1/doc/_bulk --data-binary @location_bulk_1.json
{"took":206,"errors":false,"items":[{"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEG","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHA","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHB","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHC","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHD","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHE","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHF","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHG","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHH","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHI","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHH","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHH","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHH","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHH","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201},"create":{"_index":"measurements_location_1","_type":"doc","_id":"AxydJdFibsgvDuo3qEHH","_version":1,"shards":{"total":2,"successful":1,"failed":0},"status":201}}]}[ec2-user@ip-172-31-3-104 ~]$
```

Fig. 3. Pushing test data to Elasticsearch

Application

Postman is a desktop tool for executing HTTP requests to web resources like an API client. There is a possibility to use it for monitoring API, running a test groups of test requests,

and mocking services, etc. The request and response can be converted to pretty, raw, and visualized formats. Response with available indices is displayed in the response body. In Figure 4 in desktop application Postman, it reaches out to the hosted Elasticsearch. There are 2 indices that can be divided according to a number of locations. The first column in the response indicates the health status. There are green, yellow, and red health statuses. The green health indicates that the configuration is completed. The yellow health status means that the number of shards and replicas should be changed, but it is not a major issue. The red status noticed that something went wrong with the setting or node state. The second column indicates open or close status. The index names are displayed on the third column. The next columns are primary shards, replicas number, documents count, deleted document count, store size, and primary store size.

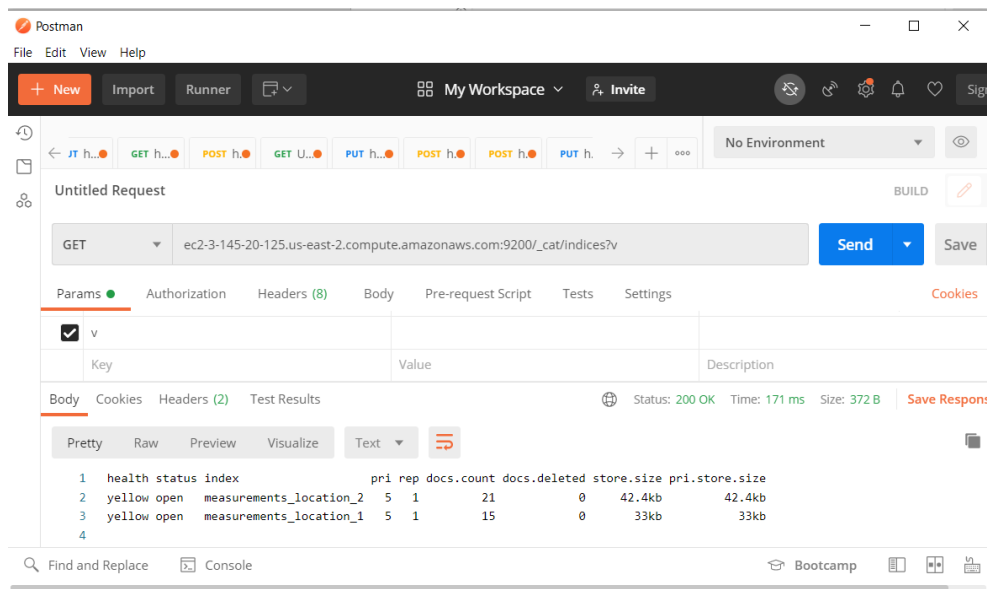


Fig. 4. Checking indices in Elasticsearch

Elasticsearch is a search engine that is used for data storage in this system. There are many helpful APIs integrated into this database. It can be considered like a NoSQL type of database due to using unstructured data JSON format for saved records and flexible schemas. Records are stored in separated indices it shown in Figure 4. The index has replicas and shards which helps to increase performance possibilities while the collected data will have a big size. In this case, there is a decided to divide data according to the location. HTTP web interface of Elasticsearch allows to execute different complex requests and receive a well-formatted response. The search engine of the data storage is based on the Lucene library. It supports full-text, partial and term-level queries which are useful for complex requests [6].

There are pros and cons of using this search engine. The main benefits are supporting full backups, parallel processing, node handling, well-documented, and open-source code. The drawbacks are a bit difficult to learn queries and terms, not supporting multi-language supporting, preparation a hierarchy of indexes and SSD's requirement [12].

The webservice is developed based on Restful API using Spring library [17] and Java programming language. Through HTTP internet protocol the request can be sent with headers and other settings. Besides the response can be received from 100 to 500 group numbers of status codes and appropriate JSON responses or messages. In Figure 5 the typical web service response is demonstrated. It contains the name of the field in keys and the result of measurements or appropriate names in the value [11].

Web service is a mediator between database and client. In this way, it allows to secure and divide responsibilities between all components of the system [7]. Values and units are displayed atomically for each parameter.

```

[
  {
    "id": "1",
    "location": "location1",
    "devices": [
      {
        "deviceName": "BVC-K",
        "l1": {
          "value": 0,
          "units": "m"
        },
        "J": {
          "value": 0.85,
          "units": "A"
        },
        "h": {
          "value": 1.7,
          "units": "m"
        }
      },
      {
        "deviceName": "VPP-M",
        "Lu": {
          "value": null,
          "units": "m"
        },
        "Umg": {
          "value": null,
          "units": "V"
        },
        "Vmg": {
          "value": null,
          "units": "V"
        },
        "Ugg": {
          "value": null,
          "units": "V"
        },
        "Vgg": {
          "value": null,
          "units": "V"
        },
        "Up": {
          "value": null,
          "units": "V"
        }
      }
    ]
  }
]

```

Fig. 5. Web service response

Visualization results

The most comfortable way to analyze is a visualization of results. Nowadays, there is a list of techniques for data visualization [9]:

- bar chart is used for categorical data or clustered groups. It can be used to compare values;
- histogram allows demonstrating data in ranges;

- scatter plot is helpful to visualize data using Cartesian coordinates to highlight the correlation between values in the axis;
- network shows the relationships between entities;
- heat map represents the magnitude of something in two dimensions;
- line chart is a typical way to visualize dependencies between two parameters;
- radar charts represent multidimensional data in multidimensional charts;
- in addition, there are other types of data representation: flowchart, treemap, Gantt chart, stripe graphic, etc.

Data from web service can be represented in charts or graphs. All visualizations are displayed one under the other on the web page. The charts will be changed dynamically when new data is from the web service.

For this purpose, D3.js library is used in web application for the building data models. This JavaScript library is used to bind data to a web page using a data-driven approach to apply it in the document. In addition to simple graph visualizations using D3, it is allowed to create interactive graphs and other graphic images. The latest versions of the library are multi-browser [10].

The line chart is the most popular way to visualize data in two-dimension axes. This 2D representation allows identifying changes between measured results [8].

As we can see in Figure 6 the graph of change of depth of measurement of underground metal elements of constructions with distance is shown. In this figure, the upper graph with location 1 label is based on received data and the lower with location 2 label, respectively, distribution underground metal elements of constructions.

The X-axis shows the depth of the underground construction in meters which is indicated by H letter, and the Y-axis shows the distance in meters which is indicated by l letter. This graph shows that the depth of the underground metal elements of construction increases with distance in the study areas [1].

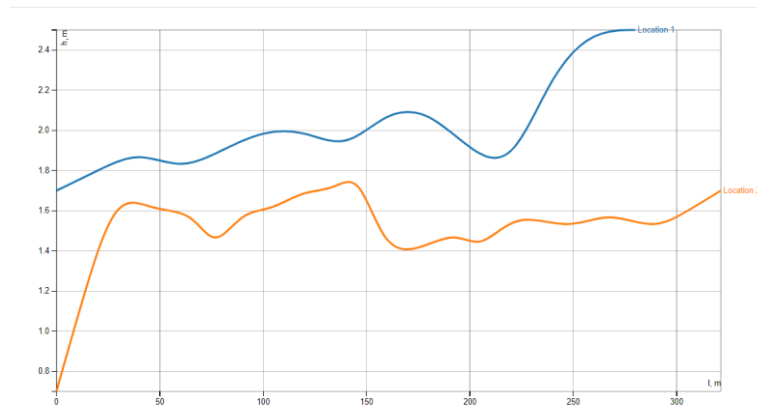


Fig. 6. The results of visualization of the depth h of the underground metal elements of construction along the main construction on BVS-K for two locations.

Figure 7 shows the results of visualization of the polarization potential in underground metal elements of constructions with distance using VPP-M for two mentioned locations. On

this graph, we can see that the polarization potential increases in a positive direction in the location on the contrary decreases in negative values.

In addition, based on this graph, we can determine the possible location of damage to the insulating cover of the underground metal elements of construction. In the area where the largest peak requirements of electrochemical protection are violated, namely, the value is approximately equal to -0.8 V. Thus, it is possible to accurately determine the location of such a defect that requires possible repair. This area can be indicated on the graph to report a problem [1].

As you know, graphics are better perceived by people than raw data [20]. This representation allows even non-technical people to see visually a change in the behaviour of a system or phenomenon. There is a possibility to construct similar graphs in such representation. To do this, you only need to change the link to the desired resource with data in the web service.

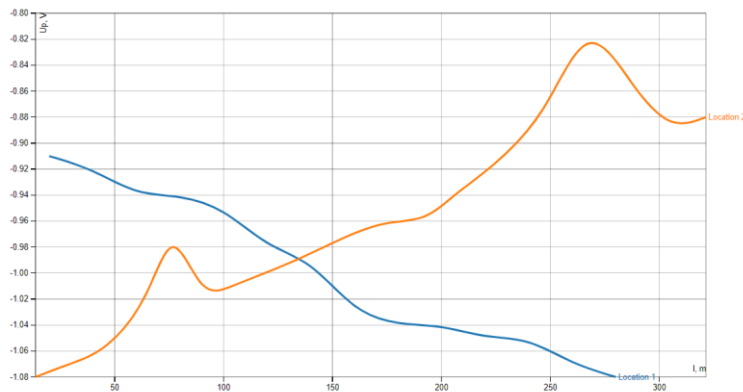


Fig. 7. The results of visualization of the polarization potential with distance for underground metal elements of construction using VPP-M for two locations.

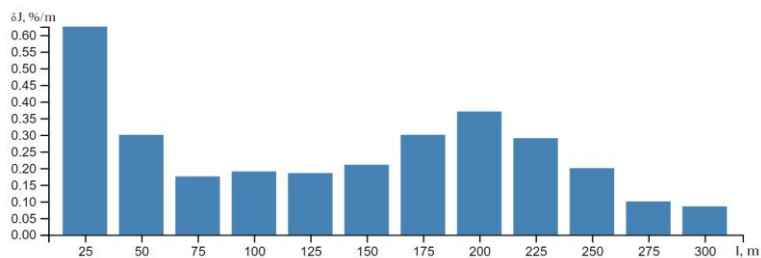


Fig. 8. The results of visualization of the relative current consumption with distance for underground metal elements of construction for location.

The change in relative current consumption [1] is shown in the histogram in Figure 8.

The most convenient is the JSON format, the type of data supply for graph visualizations and further work with them. In this way, it allows you to track the structure and hierarchy of dependencies based on keys and values [5].

The graph is one of the popular data representation forms. Nodes and edges are the main elements in this structure [13]. In Figure 8 two locations are displayed which contains two devices with measured parameters. The visualization is based on one test data that returns from a web service.

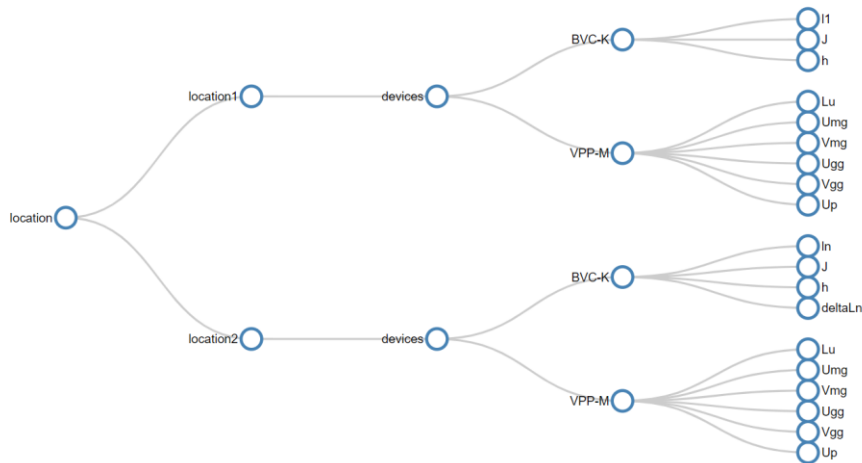


Fig. 8. Hierarchy visualization of the test data

The main benefit of such visualization is to understand hierarchical sequences. In case when relationships between entities the network graph should be shown [14]. The structure can be changed dynamically according to received parents and child nodes from web service. The process of transformation data from regular JSON format to appropriate format for network graph visualization includes reading and mapping data [21]. In Figure 9 the network graph is displayed. This visualization is interactive that is why it can be moved to the web page.

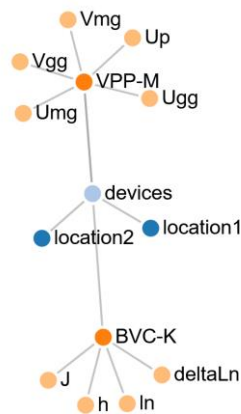


Fig. 9. Visualization of network graph based on test data

Conclusion

Remote control of systems with client-server architecture simplifies communication between all components of the developed monitoring system and is accompanied by a large amount of computation. Because this type of calculation is often performed manually, it is important to monitor the situation in the system in real-time. Appropriate modeling includes data collection, processing, visualization and use of results to predict optimal remote modes in the monitoring data processing system. An appropriate approach to data processing using cloud computing technology helps to automate the selection of data on underground metal elements of constructions (UMEC) and analyze them in the context of diagnosing corrosion processes near the border of the metal-aggressive environment.

The use of cloud technologies for the deployment and use of applications in real situations of remote control UMEC is considered. The selected Amazon Web Service cloud allows to run the software as a service. The main advantages of using the selected tools and analyzed configuration environments related to UMEC are described.

Data processing is the conversion of data into the appropriate format and storage in Elasticsearch. This search engine is used due to the flexible text format of queries and the ability to create a complex query. Describes the main features of the search engine and the process of basic steps to configure it in the selected cloud environment. In addition, the corresponding technique allows to increase the performance of the proposed system due to the horizontal scalability of NoSQL databases.

For the purposes of UMEC testing, in the first approximation, we use two data sets from different places and expand this procedure to obtain more useful information. This approach helps to reduce the impact of the human factor and time on the selection and processing of information. An appropriate UMEC monitoring data processing system can be scaled with more data and this will reduce the processing time of large amounts of data.

Because the Restful web service is used, the main elements of communication are HTTP requests and the JSON data format. This data format is best suited for hierarchical data storage and visualization field dependencies for UMEC. In addition, the visualization of data in the graph display is easy to understand and informative. The main methods of data visualization are considered and the possibilities of a linear diagram, histogram and in the form of graphs with respect to polarization potentials are demonstrated.

It should be noted that the proposed approach of remote control of complex objects, taking into account cloud technologies, can be used to automate the process of data processing and visualization in other areas.

REFERENCES

- [1] *Lozovan V., Dzhalala R., Skrynkovskyy R., Yuzevych V.* Detection of specific features in the functioning of a system for the anti-corrosion protection of underground pipelines at oil and gas enterprises using neural networks // *Eastern-European Journal of Enterprise Technologies*. 2019. Vol. 1. No 5 (97). – P. 20–27. DOI: <https://doi.org/10.15587/1729-4061.2019.154999>.
- [2] *Юзевич В. М., Лозован В. П.* Вплив механічних напружень на ріст корозійної тріщини у стінці трубопроводу // *Фіз.-хім. механіка матеріалів*. 2021. Т. 57, № 4. С. 96-103.

-
- [3] Overview of Amazon Web Services. [Online]. URL: <https://d1.awsstatic.com/whitepapers/aws-overview.pdf>
- [4] Running Elasticsearch on AWS. [Online]. URL: <https://www.elastic.co/blog/running-elasticsearch-on-aws>
- [5] *Aleksei Voit, Aleksei Stankus, Shamil Magomedov, and Irina Ivanova*. Big data processing for full-text search and visualization with elasticsearch // *International Journal of Advanced Computer Science and Applications*, Vol. 8, No. 12, 2017. DOI: <https://doi.org/10.14569/IJACSA.2017.081211>
- [6] *Rogozinski M., Kuc R.* Elasticsearch Server. Packt Publishing, Birmingham B3 2PB, UK, 2014.
- [7] *Thomas, Manoj A. and Redmond, Richard T.* From the Client-Server Architecture to the Information Service Architecture, AMCIS 2009 Proceedings. 115.
- [8] *Stephen R.* Midway Principles of Effective Data Visualization. *Perspective*, Vol. 1 (9), 2020. doi: <https://doi.org/10.1016/j.patter.2020.100141>
- [9] Data visualization [Online]. URL: https://en.wikipedia.org/wiki/Data_visualization
- [10] Data-Driven Documents [Online]. URL: <https://d3js.org/>
- [11] *Mahdi Bennara, Michael Mrissa, Youssef Amghar*. An Approach for Composing RESTful Linked Services on the Web. *World Wide Web*, Apr 2014, Seoul, South Korea. doi: <https://doi.org/10.1145/2567948.2579222>
- [12] *Luburić N, Ivanović D.* Comparing Apache Solr and Elasticsearch search servers 6th Int. Conf. Inf. Soc. Technol. ICIST 2016, P. 287–291
- [13] *R. Pienta, J. Abello, M. Kahng, and D. H. Chau*. Scalable graph exploration and visualization: Sensemaking challenges and opportunities *Big Data Smart Comput. (BigComp)*, IEEE, 2015.
- [14] *R. M. Tarawneh, P. Keller, and A. Ebert*. A general introduction to graph visualization techniques // *OASISs-OpenAccess Ser. Informatics*, vol. 27, 2012.
- [15] *V. Гошовський, В. Дзіковський, Р. Мисюк, В. Рабик, І. Сасовець*. Система збирання інформації на основі мікрокомп'ютера Raspberry PI // *Електроніка та інформаційні технології*. 2017. Випуск 8. – С. 102–110
- [16] *García, S., Ramírez-Gallego, S., Luengo, J. et al.* Big data preprocessing: methods and prospects. *Big Data Anal* 1, 9 (2016). doi: <https://doi.org/10.1186/s41044-016-0014-0>
- [17] *Dashrath Mane, Ketaki Chitnis, Namrata Ojha*. The Spring Framework: An Open Source Java Platform for Developing Robust Java Applications // *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*. 2013. Vol. 3 (2). – P. 137-143
- [18] *Ji Changqing & Li Yu & Qiu Daowen & Jin Yingwei & Xu Yujie & Awada Uchechukwu & Li Keqiu & Qu Wenyu*. Big data processing: Big challenges. *Journal of Interconnection Networks*. 2013. DOI: <https://doi.org/10.1142/S0219265912500090>.
- [19] *Dutta, Pranay & Dutta, Prashant*. Comparative Study of Cloud Services Offered by Amazon, Microsoft and Google. *International Journal of Trend in Scientific Research and Development*. 2019. Volume-3. 981-985. doi: <https://doi.org/10.31142/ijtsrd23170>.

- [20] Eric Hehman, Sally Y. Xie. Doing Better Data Visualization. *Advances in Methods and Practices in Psychological Science* October-December 2021, Vol. 4, No. 4, pp. 1–18. DOI: <https://doi.org/10.1177/25152459211045334>.
- [21] Peng, Dunlu & Cao, Lidong & Xu, Wenjie. Using JSON for Data Exchanging in Web Service Applications. 7. 2011

ОПРАЦЮВАННЯ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ ДЛЯ ВИЯВЛЕННЯ КОРОЗІЇ

Роман Мисюк¹, Володимир Юзевич^{1,2}, Ірина Мисюк¹

¹Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50, м. Львів, 79005, Україна
mysiukr@urk.net

²Фізико-механічний інститут ім. Г. Карпенка НАН України,
вул. Наукова, 5, м. Львів, 79601, Україна

Україна - одна з країн, через які проходять підземні металеві трубопроводи з газом та нафтою. Під впливом зовнішніх і внутрішніх факторів метал піддається корозії і при цьому виникають дефекти. Однією з важливих проблем є виявлення дефектів у підземних трубопроводах дистанційними методами. Основною причиною, чому ми повинні досліджувати можливість швидкого виявлення таких дефектів, є можливість аварій, які можуть призвести до техногенних катастроф та небезпечних викидів вуглеводнів у повітря та в землю. Для виявлення корозійних дефектів пропонується використовувати результати обробки вимірювань струмів та потенціалів. Одним із сучасних підходів до опрацювання даних є використання хмарних технологій, які дозволяють не тільки надавати публічний доступ до ресурсів, а й використовувати інструменти для роботи з інформацією. У хмарному середовищі Amazon Web Service використовується підхід розробки клієнт-серверної архітектури програмного забезпечення з візуалізацією результатів у веб-додатку. Центральним елементом є REST веб сервіс, який з'єднує всі елементи системи. Оскільки ця архітектура базується на використанні HTTP протоколу передачі даних та роботи з JSON форматом даних, то нереляційну базу даних Elasticsearch використано для зберігання даних. Це сховище даних є гнучким для роботи з даними, бо дозволяє виконувати складні запити для повнотекстового, часткового та фільтрованого пошуку інформації. Обробка даних автоматизована і на їх основі можна виявляти локалізацію корозійних дефектів з урахуванням візуалізації результатів. Графіки можуть відображати результати обчислень в режимі реального часу. Процес візуалізації відбувається на основі використання динамічної бібліотеки для побудови графіків D3.js, яка дає змогу працювати з більшістю типів представленнями даних. У роботі розглянуто застосування звичайної лінійна та стовпчаста діаграми. Крім цього, приведено модель комплексного застосування системи від етапу збирання даних до їх візуалізації. Така система обробки даних може замінити ручну роботу, пов'язану з відбором, опрацюванням та візуалізацією даних в реальному часі.

Ключові слова: опрацювання даних, візуалізація даних, корозійні дефекти, підземні трубопроводи, система відбору даних, хмарні технології.

Стаття: надійшла до редакції 05.11.2021,
доопрацьована 16.11.2021,
прийнята до друку 18.11.2021