DOI: 10.2507/35th.daaam.proceedings.xxx

ENHANCING THUNDERSTORM PREDICTION AND USER PROTECTION: THE EVOLUTION OF DIGITAL PROCESSING SYSTEM ZEUS 2.0

Josip Krzelj, Kresimir Rakic & Zeljko Seremet

This Publication has to be referred as: Krzelj, J[osip]; Rakic, K[resimir] & Seremet, Z[eljko] (2024). Enhancing Thunderstorm Prediction and User Protection: The Evolution of digital Processing System Zeus 2.0, Proceedings of the 35th DAAAM International Symposium, pp.xxxx-xxxx, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-xx-x, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/35th.daaam.proceedings.xxx

Abstract

Zeus 2.0, an advanced iteration of Hrvatski Telekom's storm forecasting system, represents a significant leap in proactive weather management. This system, integral to the Proactive Care process, integrates a sophisticated algorithm with real-time meteorological data to enhance thunderstorm predictions and mitigate potential damage to user equipment. Building upon the earlier Zeus version, Zeus 2.0 features a refined algorithm that leverages the DBSCAN clustering method for accurate storm forecasting and precise lightning strike detection. It incorporates real-time data from UBIMET, a leading meteorological service known for its high precision, to deliver timely alerts through SMS and the Moj Telekom application. The system's technological foundation includes a robust stack with React for frontend development, Java for backend processes, and PostgreSQL for data management. Key tools such as Apache Kafka and Red Hat OpenShift ensure efficient data processing and system scalability. Zeus 2.0 not only improves user experience with its intuitive interface but also automates critical processes like equipment replacement and troubleshooting through seamless integration with Camunda BPM for business process management. This evolution marks a substantial advance in optimizing user safety and system performance, paving the way for future enhancements in weather prediction and user protection.

Keywords: digital processing system; thunderstorm prediction; DBSCAN algorithm; user experience.

1. Introduction

Unstable weather with high temperatures and storm clouds is ideal for the formation of thunderstorms. This fascinating natural phenomenon can be very dangerous for the safety of people and their property. The development of the Zeus 2.0 system, as a component of the Proactive Care process offered to its users by Hrvatski Telekom in the form of the Moj Telekom application, stems from the need to improve the system for forecasting storms, preventing potential damage to equipment due to lightning strikes, and the process of replacing equipment.

Digital Processing System (DPS) Zeus 2.0 represents the evolution of DPS, based on the previous version of Zeus. This new version was developed to improve the existing algorithm for predicting thunderstorms and allow users timely information and preventive measures to protect their property.

The previous version, although useful, proved to be less effective due to limited analysis and lack of user experience. The goal of the project is the integration of meteorological data with the user experience, which enables proactive management and reduces the need for user support interventions. For now, the linear algorithm has been put into production, and by cooperating with meteorological services, Zeus 2.0 lays the foundations for further innovations, with the ultimate goal of optimizing the user experience and reducing the need for user support. This system evolution represents a technological step towards better connectivity and user security.

Zeus 2.0 is designed to provide a comprehensive solution for thunderstorm prediction. Its advanced algorithm enables precise forecasting of weather conditions, while integration with process flows ensures automatic and timely reactions to upcoming storms. In this way, users can turn off their equipment in time, which minimizes the risk of burning or other forms of damage.

In addition to technical advantages, Zeus 2.0 also provides an exceptional user experience thanks to a modern and intuitive interface that allows easy access to key information and functionalities. Integration with different services enables faster and more accurate data processing, which results in reliable and accurate information for end users.

In the second chapter of the paper, we will describe the technologies used during the development of the Zeus 2.0 system. The functionalities of the system and its use cases are described in the third chapter. The fourth chapter describes the used algorithm for predicting the thunderstorms, implemented in the Zeus 2.0 system. In the fifth, concluding chapter, an overview of the described system and guidelines for continued research and development are given.

2. Technologies Used in the Application Development

Zeus 2.0 is a network application that uses advanced technologies for thunderstorm prediction, with support for synchronization and actions through Camunda workflows. Integration with different services further improves the user experience, making the application more intuitive and efficient for end users.

Globally, Zeus 2.0 implements a wide range of enterprise technologies that ensure high levels of performance and reliability. Technologies used for frontend development include React, a JavaScript library for building user interfaces. Java technology was used for the backend development. It is an object-oriented programming language known for its robustness and portability, which is crucial for large companies. Key elements of the backend support system include Camunda, a business process automation platform that enables process modelling and execution. PostgreSQL is used as a database. Other significant technologies that support the operation of the system are Apache Kafka, a distributed platform for Internet streaming, which enables high throughput and low latency data processing in real-time, and Red Hat OpenShift, a cloud platform for management containers, which enables the orchestration of applications in the Kubernetes environment with additional tools for development and deployment.

2.1. Frontend Development Technologies

Front-end technologies enable the building of user interfaces and interaction with applications, whether on web or mobile platforms. They enable content display, event management and communication with backend services.

React is the most powerful JavaScript library used to create user interfaces. It was developed by Facebook and is often used to build single-page applications (SPA) and mobile applications. React was first released in May 2013 and has since become one of the most popular JavaScript libraries for building user interfaces. It is widely adopted by the developer community, and there are many third-party libraries and tools available that integrate with React. React allows developers to create reusable UI components, which can make it easier to build and maintain user interfaces. In addition, React uses a virtual document object model (VDOM), which represents a key innovation in React, improving the performance and efficiency of updating the user interface. VDOM becomes crucial in situations where there are frequent updates to the application state. Speed and efficiency are achieved by optimizing and updating only the parts of the page that have changed, instead of the entire DOM. This reduces the cost of updates, providing a faster user experience. Another important feature of React is JavaScript XML (JSX), which connects Javascript with XML. HTML-like syntax makes development easier for developers familiar with web technologies. JSX allows JavaScript expressions to be embedded, providing dynamism to the code. It is also important to mention React Hooks, which represent a significant step towards a more functional approach to component development [1].

2.2. Backend Development Technologies

Backend technologies are responsible for managing business logic, databases, and communication between servers and client applications. Java is an object-oriented programming language that is known for its portability, robustness and wide application in the development of business applications.

Developed at Sun Microsystems, Java was originally conceived as a language for managing household appliances. However, it quickly became apparent that Java has wider possibilities and is adapted for developing secure and platform-independent applications that stand out for their ability that its code written once can run anywhere. This platform-independent nature stems from the use of the Java Virtual Machine (JVM), which allows the execution of Java code on different operating systems without recompilation [2].

In the context of the Zeus 2.0 system implementation, Java proved to be the optimal choice due to its object-oriented nature, high level of abstraction and platform-independent performance. This flexibility enables the development and -implementation on different operating systems without compromising efficiency and ease of maintenance. Java was also chosen for its security, automated memory management, and parallel execution capability, which is extremely important for complex systems such as Zeus 2.0 [3].

2.3. Database Technologies

In today's information age, accessing and handling information is one of the key activities in both professional and private life. Databases are systems for storing, managing and retrieving data in an organized manner.

PostgreSQL has its beginnings at Berkeley University in California, in the form of the POSTGRES project (POST inGRES). In its initial phase, POSTGRES was conceived as a database management system that would extend the functionality of the earlier INGRES system, particularly by supporting object-relational concepts and extending SQL [4]. In 2006, PostgreSQL went fully open source, paving the way for a global community of developers, users, and database enthusiasts. The community has become the key force behind the development and improvement of PostgreSQL, thus maintaining a high standard of quality and innovation [5].

PostgreSQL was selected for the Zeus 2.0 project because of its advanced support for SQL standards, robustness, flexibility in handling different types of data, and ability to handle large workloads and complex queries. PostgreSQL is widely used in industry, from small projects and startups to large corporations and organizations that require powerful and scalable databases. Features such as support for JavaScript Object Notation (JSON) binary data types, advanced query management and optimization capabilities, and various enhanced functionalities testify to the constant work on innovation within PostgreSQL.

PostgreSQL is efficient in handling large amounts of data and flexible in data processing. Zeus 2.0 includes a geographic information system, and PostgreSQL provides robust support for spatial data. As an open-source system, PostgreSQL provides transparency and customizability, which is crucial for projects like Zeus 2.0. which require customization according to specific requirements [6].

2.4. UBIMET Data Source

UBIMET is an Austrian company specializing in meteorological services and weather monitoring technologies. UBIMET has built an unparalleled network in Europe, setting a new standard in localization precision and efficient detection. The data is updated every thirty seconds, providing real-time information, unlike traditional radars that update every 10 minutes. The positioning error is less than 100 meters, which ensures exceptional accuracy and reliability.

Patented 3D technology is used to analyse strength, polarity, height, and the relationship between clouds and the ground or each other. For the described case, it is important to note that the analysis of the relationship between clouds and the ground is used. This sophisticated approach enables a deeper understanding of atmospheric phenomena [7].

UBIMET stands out for its patented algorithms for predicting lightning behaviour. These internal algorithms enable hazard analysis and timely issuance of severe weather warnings. All these advanced features make UBIMET a reliable partner for monitoring weather conditions with extraordinary precision and speed.

The Zeus 2.0 system requires cloud-to-ground lightning strike data, while cloud-to-cloud strike data is not collected. This information is updated once a day and allows for monitoring surveillance and improving the level of preparedness [8].

2.5. Other Technologies

Red Hat OpenShift and similar platforms are used to deploy, manage and scale applications in containers. Entire DevOps processes and all projects can be built on them because they are cloud platforms that enable the development, testing, implementation and management of applications in a highly available and scalable environment. Red Hat Open Shift is the leading hybrid cloud application platform that combines the power of Kubernetes to provide a robust environment for developing, modernizing, deploying, running and managing applications. This innovative platform integrates proven services aimed at overcoming challenges throughout the application lifecycle. The OpenShift Container platform is a platform for developing and running containerized applications. It is intended to enable applications and data centres to scale from a few machines and applications to thousands of machines serving millions of clients. The ELK platform (Elasticsearch, Logstash, Kibana) and Grafana are used for observability, analysis and alarming. We use these powerful tools to monitor and interpret logs, providing an efficient solution for system analysis and immediate response to potential challenges [9].

Apache Kafka is a distributed web streaming platform used to build real-time applications that can process and analyse large amounts of data with low latency. The task of these technologies is to enable reliable and scalable data exchange between systems and applications. Kafka is often used in a big data infrastructure environment. Apache Kafka is an open data stream processing system that was originally developed at LinkedIn to solve the problem of connecting different applications and services. Since its introduction as an open-source project in 2011, it has become a popular tool for processing large amounts of data in real-time. It quickly became a key component of many microservice architectures and data processing platforms. One of the reasons for using Apache Kafka for Zeus 2.0 is to save log records and history of actions, which enables monitoring and analysis of all system activities. Furthermore, big data integration provides a great advantage because Kafka integrates with this concept, enabling the transfer of large amounts of data to different topics within the Kafka cluster. Connectivity with tools such as ELK, Grafana and Prometheus enables the Zeus 2.0 system to provide detailed insight and monitoring of system performance through visualization of logs and metrics [10].

Camunda is a platform for automating business processes that enables the modelling, execution and monitoring of processes using the Business Process Model and Notation (BPMN), Case Management Model and Notation (CMMN) and Decision Model and Notation (DMN) standards. Camunda is often chosen because of its openness, flexibility, support for standards and ease of integration with existing systems, which enables efficient and scalable automation of business processes, this is one of the key reasons why it was chosen. Camunda engine is a powerful tool for automating business processes. Using the BPMN standard enables visual modelling of business processes, which facilitates collaboration between teams of business analysts and developers. Zeus 2.0 uses these models to define, optimize and implement its business processes [11]. There are several reasons for this:

- Camunda enables the efficient execution of business processes. Each step in the BPMN model represents a specific activity that can be automated or assigned to the user.
- Camunda stands out for its flexibility and enables dynamic adaptation of business processes to changes in real-time.
- Camunda is an open-source program support, which means that it is available for everyone to use, adapt and improve.
- BPMN models provide a clear and visual representation of business processes, facilitating communication between teams, enabling better process understanding among stakeholders, and helping to detect potential problems before implementation.
- Camunda can be integrated with a wide range of technologies enabling smooth coordination between different parts of the system.

The combination of these characteristics makes the Camunda BPMN engine extremely suitable for Zeus 2.0, providing a reliable and efficient environment for managing business processes [12][13].

3. System for Preventive Notification about Thunderstorms Zeus 2.0

One of the DPS systems integrated into the Proactive Care process is Zeus 2.0. The Proactive Care process consists of the following units and systems:

1. Ingestion. To retrieve data from the network, different platforms and systems, advanced systems like Grabber and multiple instances of Flume are used. These systems enable efficient data entry into a big data architecture, ensuring a high level of reliability and scalability.
2. Analysis. In the data analysis phase, sophisticated algorithms and machine learning (ML) models are applied. Java Spark and Spark ML programs on the cloud platform are used, and the big data team uses PySPARK programs with built-in ML models running through AirFlow. Additionally, Proactive Care Analysis hosted on OpenShift provides advanced data analysis to provide deeper insights.
3. Actions. After data analysis, actions are initiated through Proactive Care Action based on the Camunda BPM system. This system enables the implementation of proactive measures based on the results of the analysis, contributing to the continuous optimization of the process. Integration with Camunda BPM provides an agile approach to managing business processes in accordance with the results obtained from big data analysis.

This complete architecture enables an efficient and structured approach from data retrieval, through analysis to the implementation of actions, which ensures process optimization and informed decision-making.

Integrations play a key role in the exchange of data between different systems. In the current version of the system, Big Data Kafka is used for efficient and reliable communication between different parts of the system that manipulate large data sets. As for integrations with legacy systems, API integrations are currently being implemented according to PCAC standards, in order to enable smooth and standardized communication with existing systems. This integration helps maintain compatibility with previous systems and enables efficient data exchange between new and older technology solutions. All these integrations together make the architecture adaptable, resilient and efficient in handling data, both within the system and in interaction with external sources and targets.

Part of Proactive Care is a system that performs preventive actions based on data obtained from Big Data Kafka Topic. The system uses a POSTGRES database to store information. System monitoring is done using the ELK stack with a focus on Application Performance Monitoring (APM) with *apmcamprob* sample monitoring.

This system was developed in the Java programming language, using the Camunda engine for managing business processes. Camunda provides an environment for modelling, executing and monitoring business processes, while the POSTGRES database serves as a reliable data storage system.

3.1. Zeus 2.0 Use Case 1

A dynamic system was created to monitor meteorological data using the UBIMET API. 1,200 key locations across Croatia were defined, clusters were made, and information about lightning strikes in real time is updated daily through API requests, and comprehensive weather forecasts, daily, weekly, and precise lightning strikes in real time are periodically retrieved. The big data platform regularly captures this data, providing a complete overview of forecasts for the current day and the next 7 days at all 1,200 locations. This amount of information lays the foundation for in-depth analysis and research.

The system provides insight into complex patterns of weather changes across the country, providing a valuable resource for making informed decisions and conducting thorough research, with always up-to-date and accurate data.

3.2. Zeus 2.0 Use Case 2

This use case represents a proactive approach to weather monitoring and consists of two parts.

Part 1 performs the sending of weather warnings (thunderstorms, etc.) based on prognostic data of the big data platform via SMS or the Moj Telekom application. Rules are also defined for recognizing router burnout. Sending an automatic notification via the Moj Telekom application or SMS (with a link to install the Moj Telekom application) is done as follows:

- If the user has been logged into the Moj Telekom application for the last 30 days, the notification goes through the application.
- If the user is not logged in, then via SMS with a link to install the Moj Telekom application.

Part 2 starts after the initial thunderstorm warning and covers:

- Detection of burnt equipment via Zeus message.
- If working in group interference, it is set as a priority.
- Implementation of equipment replacement through the Moj Telekom application.
- Alternative channels for equipment replacement: T-Centre or distributors.
- The interference is opened via the chat application.

3.3. Zeus 2.0 Use Case 3

UBIMET continuously monitors data on thunderstorms throughout Croatia. Information is collected about Digital Subscriber Line Access Multiplexer (DSLAM) devices including:

- DSLAM name/slot/port
- Asset data, assetID
- x and y coordinates
- Address (town, city, postal code)
- User segmentation (business or residential)
- The services that users have are also monitored.

Spark is used for continuous analysis of input data. The Spark system checks to see if the user's modem is in the dying gasp status, which signals the modem to turn off power. If the power outage occurred because the user intentionally turned off the modem, those cases are discarded where we get information that the user turned off the power. In situations where the damage was caused by lightning, we do not record any information that the System Problem Analyst (SPA) records. Synchronization is also checked.

A modem is considered synchronized if it is connected to the exchange. Spark sends a message to the Kafka platform for Camunda. Camunda, after receiving a message, starts a certain process according to the defined business logic. This includes notification, automatic replacement or other steps depending on the situation. Camunda can make decisions like waiting a certain amount of time before taking further steps. All these operations are automated to respond quickly and efficiently to problems caused by thunderstorms.

3.4. Camunda Process Flows in Zeus 2.0

Three process flows defined by the BPMN standard are implemented in Camunda:

- SMS alerts, informs the end user about a thunderstorm.
- Burned-out equipment detection, describes the process of confirming the detected burnt-out equipment and notifying the end user.
- Replacement of equipment, opening a complaint in system for reported complaints Donat, and the procedure of replacing the equipment itself.

4. **Algorithm for Thunderstorm Prediction**

Algorithm Density-Based Spatial Clustering of Applications with Noise (DBSCAN) makes it possible to group data into sets based on the density of the spatial distribution of data. This algorithm identifies subgroups of data based on how densely they are distributed, ignoring areas of low density that are considered noise. The algorithm was first proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996 [14].

DBSCAN is a popular clustering algorithm used in data analysis to identify groups or clusters within a data set. It is especially useful when we have complex data structures with noise and when the number of clusters is not known in advance.

This algorithm is used to identify clusters of thunderbolts. Thunderstorms are considered part of the same cluster if their mutual distance is less than 3 km.

DBSCAN is based on two key concepts:

- Eps (ε): Radius of the environment within which adjacent data is searched. If there is a sufficient number of points within this radius, that environment is considered densely populated.
- MinPts: The minimum number of points required within the Eps radius for a point to be considered a centre point of a cluster.

The DBSCAN algorithm groups points based on density, i.e. how close the points are to each other. The work process can be described as follows [15]:

- The algorithm randomly selects a point from the data set. For a selected point, the algorithm finds all neighbouring points within the Eps radius.
- Checking the minimum number of points (MinPts): If a point has enough neighbours (≥ MinPts), it is considered the central point of the cluster, and the cluster is expanded to include all neighbours that also meet the density criteria.
- The process continues recursively for all points within the cluster, thus expanding the cluster.
- Points that do not belong to any cluster are considered noise.

Some of the advantages of the DBSCAN algorithm include:

- Does not require a predefined number of clusters.
- Can identify clusters of different shapes and sizes.
- Handles noise and isolated points well.

Disadvantages of the DBSCAN algorithm are:

- Selecting Eps and MinPts can be challenging and may require multiple testing for optimal results.
- Does not work well with data of different densities.

When DBSCAN is applied to lightning clustering, the points (lightning) are analysed based on their spatial proximity. The Eps and MinPts parameters are selected to match the expected lightning density in a given area. The clusters that DBSCAN identifies can help understand how lightning clusters during certain weather conditions or in specific geographic regions, providing useful information for meteorology and related scientific disciplines.

The implementation of the DBSCAN algorithm includes the following steps:

1. Checking the conditions for prediction. For each cluster, it is checked whether the conditions for making a prediction are met. If less than 1 minute has passed since the appearance of lightning or if the cluster contains less than 3 lightning, no prediction is performed.
2. Data processing by minute. Thunderstorm data is sorted by minute. For each minute, a centroid is created that represents the mean value of the lightning positions in that minute.
3. Performing linear regression. Based on the centroid, a linear regression is performed to predict the direction of the storm.
4. Projection and distance calculation. On the line obtained by linear regression, the first and last centroids are mapped by orthogonal projection. The distance between those two points on the line is calculated.
5. Calculation of the predicted distance and angle. Based on the calculated distance and the time difference between the first and last centroid, the predicted distance that the storm will travel in an hour is calculated. The angle and the slope of the direction of the storm movement are calculated.
6. Prediction of the movement point. Based on the last centroid and the calculated distance with the slope, the prediction point of the storm movement is determined.
7. Adjustment of the coefficient. If the storm moves north, the coefficient is 1. In other cases, the coefficient is adjusted according to the geographical background. For lowland areas like Slavonia, if the coefficient is greater than 1.5 and the cluster contains more than 500 lightning strikes, only the data for the last 15 minutes are taken into account and the process is repeated with possible adjustments.
8. Calculation of cluster radius. Finally, the cluster radius is calculated, which is used to determine the area within which alerts will be sent. This approach enables more precise analysis and prediction of storm movements based on lightning data, enabling better planning and response to weather disasters.

The functionality of the DBSCAN algorithm is shown in Figure 1.

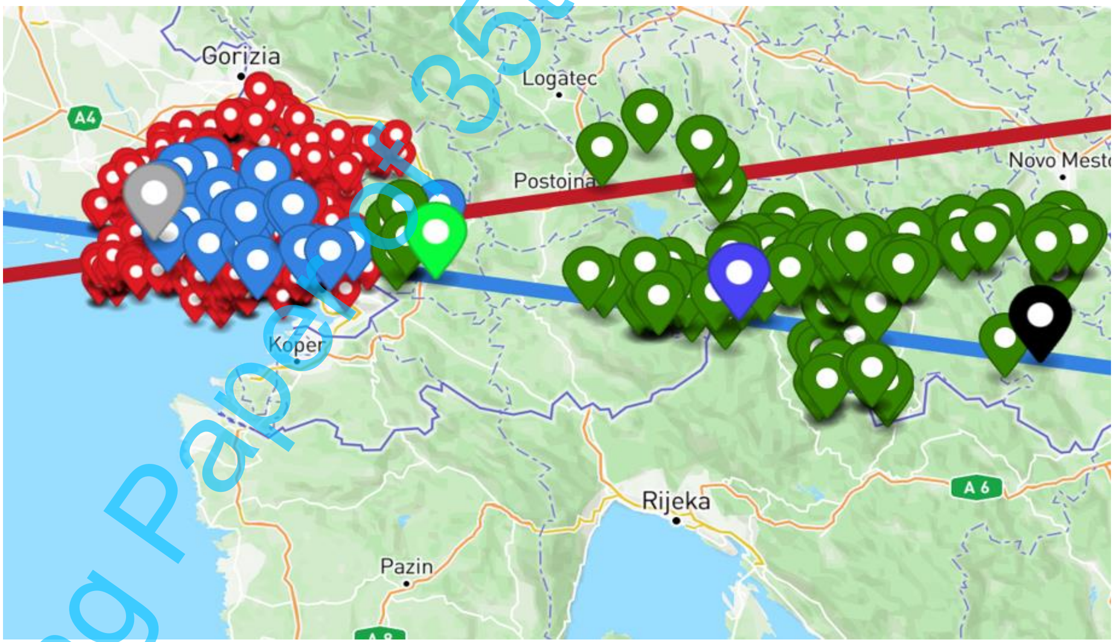

Fig. 1. Showing the direction of lightning on the Google map

The red tags are the thunderstorms with which the prediction was calculated, the green are the thunderstorms that occurred after an hour (±15 minutes), the dark blue tag is the prediction for half an hour, and the black tag is the prediction for the hour. The storm is actually moving along the blue line (it's a linear regression of the centroid of the lightning each minute), and the red would be a linear regression in general.

The ELK stack is used to monitor and control the very success of the model in such a way that the logs overtake the production environment, and the display map is automatically generated. ELK also allows easy manipulation of time. Logs are generated every 15 minutes [7]. The functionality of the ELK stack is shown in Figure 2.

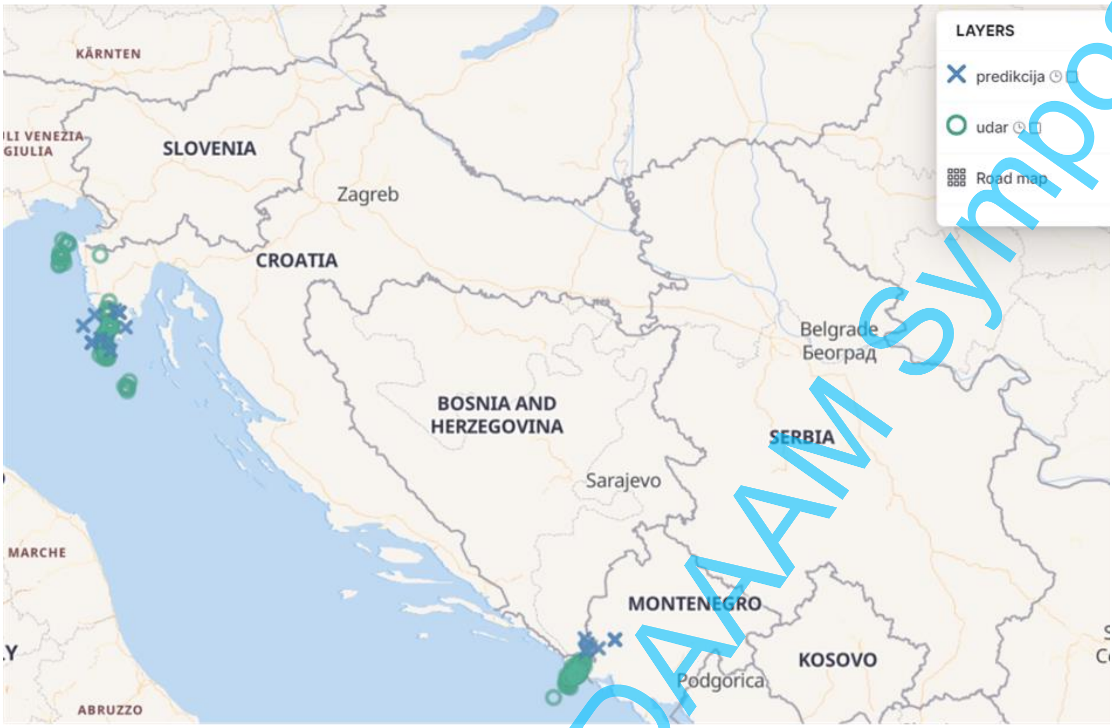

Fig. 2. Showing the direction of lightning on the Google map

5. Conclusion

The implementation of the Zeus 2.0 project represents a significant step forward in providing quality service to Hrvatski Telekom users, setting new standards in the telecommunications sector. This initiative enables a quick and efficient reaction to potential difficulties that users may have during thunderstorms. By monitoring and predicting the movement of storms, users are provided with important information and instructions on how to protect themselves from possible service interruptions. It also identifies users whose equipment has burned out, providing them with precise information about the type of disturbance and the steps they need to take.

In addition to increasing user satisfaction, the implementation of this project reduces the time required for troubleshooting and enables faster replacement of burned-out equipment. In addition, the project contributes to increasing the safety of users, by informing them in advance about the potential dangers of a lightning strike.

There are several directions for further improvement of the Zeus project. For example, using statistics on areas more frequently affected by lightning strikes would allow for the organization of repair technicians as needed, which would further speed up the response to difficulties. Also, consideration of protective equipment for users in such areas or protection of overhead installations could reduce the possibility of equipment damage during storms.

In the future, the development of machine learning models that would provide even more accurate data on lightning strikes would be a step further in ensuring optimal safety and user satisfaction. This initiative, open to upgrades and integration with other systems, emphasizes the advantage of continuous improvement and adaptation to new technologies, ensuring that users always get the best possible experience.

6. References

[1] https://react.dev/learn, (2024). React. The Library for Web and Native User Interfaces, Accessed on: 2024-08-21
[2] Kristaly, D. M.; Moraru, S.-A.] & Grigorescu, C.-M. (2011). Multi-Purpose Modular Software Platform for Rapid Development of Web Applications, Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1, ISSN 1726-9679 ISBN 978-3-901509-83-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2011
[3] https://www.w3schools.com/java, (2024) W3Schools. Java Tutorial, Accessed on: 2024-08-21

[4] Rosu, S. M.; Dragoi, G.; Rosu, L. & Guran, M. (2010). Virtual Enterprise Network Solutions to Support E-learning Sites Development, Chapter 63 in DAAAM International Scientific Book 2010, pp. 725-742, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-901509-74-2, ISSN 1726-9687, Vienna, Austria DOI: 10.2507/daaam.scibook.2010.63

[5] Raguž, A. (2019). Sustav za upravljanje bazom podataka PostgreSQL, Master's thesis, Faculty of Science, University of Zagreb, Zagreb, Croatia

[6] https://www.postgresql.org, PostgreSQL: The World's Most Advanced Open Source Relational Database, Accessed on: 2024-08-21

[7] Eisenbach, S., & Palmer, C. (2017). Digital advances in monitoring and planning for natural hazards along networks and remote track sections. In AusRAIL PLUS 2017, Rail's Digital Revolution, 21-23 November 2017, Brisbane, Qld, Australia.

[8] https://confluence.t.ht.hr/display/OSSAA/Zeus+2.0, (2024). Hrvatski Telekom Intranet. Confluence Zeus, Accessed on: 2024-05-15

[9] https://www.redhat.com/en/technologies/cloud-computing/openshift/container-platform, (2024). Red Hat OpenShift Container Platform, Accessed on: 2024-08-21

[10] https://kafka.apache.org/20/documentation.html, (2024). Kafka 2.0 Documentation, Accessed on: 2024-08-21

[11] https://docs.camunda.io/docs/guides, (2024). Camunda 8 Documentation, Introduction to Camunda, Accessed on: 2024-08-21

[12] https://docs.camunda.io/docs/guides/automating-a-process-using-bpmn, (2024). Camunda 8 Documentation, Design a process using BPMN, Accessed on: 2024-08-21

[13] Rucker, B. (2021). Practical Proces Automation: Orchestration and Integration in Microservices and Cloud Native Architectires, O'Reilly Media, Inc., ISBN: 9781492061458, Newton Massachusetts, USA

[14] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

[15] Khan, K., Rehman, S. U., Aziz, K., Fong, S., & Sarasvady, S. (2014). DBSCAN: Past, present and future. In The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014) (pp. 232-238). IEEE.