Real-Time Digital Twin Platform: A Case Study on Core Network Selection in Aeronautical Ad-hoc Networks

Lal Verda Cakir[1,2], Mihriban Kocak[3], Mehmet Özdem[4], and Berk Canberk[1,5]

[1] School of Computing, Engineering and The Built Environment, Edinburgh Napier University, EH10 5DT, United Kingdom
[2] BTS Group, Istanbul, 34469, Turkey
[3] Department of Computer Science, Aalto University, 02150 Espoo, Finland
[4] Turk Telekom, Istanbul, 344469, Turkey
[5] Department of Artificial Intelligence and Data Engineering, Istanbul Technical University, Istanbul, 34469, Turkey

Abstract: The development of Digital Twins (DTs) is hindered by a lack of specialized, open-source solutions that can meet the demands of dynamic applications. This has caused state-of-the-art DT applications to be validated using offline data. However, this approach falls short of integrating real-time data, which is one of the most important characteristics of DTs. This can limit the validating effectiveness of DT applications in cases such as aeronautical ad-hoc networks (AANETs). Considering this, we develop a Real-Time Digital Twin Platform and implement core network selection in AANETs as a case study. In this, we implement microservice-based architecture and design a robust data pipeline. Additionally, we develop an interactive user interface using open-source tools. Using this, the platform supports real-time decision-making in the presence of data retrieval failures.

Keywords: Digital twin, real-time platform, aeronautical networks, data pipeline.

Gerçek Zamanlı Dijital İkiz Platformu: Ad-hoc Hava Ağlarında Çekirdek Ağ Seçimi Üzerine Vaka Çalışması

Özet: Dijital İkizlerin (Dİ) geliştirilmesi için dinamik uygulamaların taleplerini karşılayabilme kabiliyetindeki ve açık kaynaklı çözümlerin eksikliklerden dolayı sekteye uğramaktadır. Bu, son teknoloji Dİ uygulamalarının çevrimdışı veriler kullanılarak doğrulanmasına neden olmuştur. Ancak bu yaklaşım, Dİ'lerin en önemli özelliklerinden biri olan gerçek zamanlı verilerin entegrasyonu konusunda yetersizdir. Bu, ad-hoc hava ağları (AHHA) gibi vakalarda Dİ uygulamalarının doğrulanmasını sınırlayabilmektedir. Bunu göz önünde bulundurarak, bu çalışmada Gerçek Zamanlı Dijital İkiz Platformu geliştirilmiş ve örnek olay olarak AHHA'larda çekirdek ağ seçimini uygulanmıştır. Bunda mikroservis tabanlı mimariye sahip dayanıklı veri işleme hattı tasarlanmıştır. Ayrıca açık kaynaklı araçlar kullanarak etkileşimli bir kullanıcı arayüzü geliştirilmiştir. Bunlar sayesinde geliştirilen platform, veri alma hataları durumunda dahi gerçek zamanlı karar almayı destekler hale getirilebilmiştir.

Anahtar Kelimeler: Dijital ikiz, gerçek zamanlı platform, hava ağları, veri işleme hattı.

RESEARCH PAPER

Corresponding Author: Lal Verda Cakir, lal.cakir@napier.ac.uk

Reference: L. V. Cakir, M. Kocak, M. Özdem, B. Canberk, (2024), "Real-Time Digital Twin Platform: A Case Study on Core Network Selection in Aeronautical Ad-hoc Networks," *ITU-Journ. Wireless Comm. Cyber.*, 1, (1) 41–46.

Submission Date: Aug, 19, 2024
Acceptance Date: Sept, 26, 2024
Online Publishing: Sept, 30, 2024

1 INTRODUCTION

The DT field has limited availability of platforms that cater to the diverse and complex requirements of various applications. Existing platforms are often too generic, lacking the specific features, capabilities, and programmability needed to support highly dynamic and intricate environments [1]. Additionally, many of these are not open-sourced, restricting access for researchers and developers. This shortage can significantly impact the ability to perform a comparative evaluation of DTs [2]. Due to this, existing studies on DTs have frequently relied on pre-existing data in closed environments. This lack of real-time data integration limits the ability to accurately reflect the current state and ensure the testing is valid for real-world implementations. This challenge is especially prominent in applications that require immediate responses, such as core network selection in aeronautical ad-hoc networks (AANETs). To ensure the continuous connectivity of AANETs, the incoming data must be processed, and a new decision should be made at the right time without interrupting the service. While this can be realised to some degree using the existing data within the closed environment, a new set of challenges can arise when real-time data is integrated [3], [4]. For instance, the discrepancies and irregularities in the data collection may cause the DT model to be in temporal misalignment[5]. At this, applying estimation of these unknown factors has been a key approach [6]. Therefore, the solutions must be tested beforehand by means of integrating real-time data to identify, address, and overcome such obstacles.

1.1 Related Work

The DT technology has grown in popularity due to its ability to mimic, emulate, or replicate the characteristics or specific behaviour of corresponding physical entities. This technology corresponds to creating a software model that can create real-time virtual representations of an object, process or system [7], [8]. The capabilities of DTs are supported by the rapid development of Artificial Intelligence (AI) and Machine Learning (ML) technologies [9]. With this, DTs differentiate from classic simulation models by constantly learning and adapting [10]. In this way, predictions for the future can be made and directly integrated into management's decision-making process.

Moreover, DTs have become an important technology in the field of aviation connectivity. In this, aeronautical ad-hoc networks (AANETs) are widely used by commercial and military aviation organizations to facilitate communication. This connectivity is crucial to allow for data exchange of flight plans and weather updates between aircraft. Moreover, according to the survey results in [11] 66% of the passengers claim that in-flight connectivity (IFC) is an essential requirement while it was seen as a luxury service in the past. Overall, AANETs are required to fulfil the high-quality internet connection demands of aviation organizations. To address this, in [12], the DTs are used to increase the efficiency of core network selection in WIFI-enabled in-flight connectivity (W-IFC). This study has used the recorded data from [13], which is used in sampled periods offline. While the results acquired validated the potential of DTs in this field, the proposed architecture disregarded the streaming nature of the data. Here, developing the interaction mechanisms and related interfaces is highly important to ensure the DT model can operate accurately [14]. As also was pointed out in the reference DT platform in [15], the core functionalities of connection, DT interaction and DT visualizations are necessary. This use of microservice-based architecture offers a promising solution to ease the development and deployment of this platform [16]

1.2 Contributions

Considering these, we develop the Real-Time Digital Twin Platform with a case study on core network selection in AANETs and contribute to the literature as follows:

- We develop the real-time digital twin platform using open-source tools and leveraging microservice-based architecture. In this, we use a time-series database as the data store and develop a flask application for an interactive user interface.
- We implement the core network selection in AANETs as a case study and use the Openskynetwork API to retrieve real-time data on the aircraft. Thanks to testing with real-time data integration, we reveal that the API can fail to return data in some instances due to unknown factors. This is a highly possible case in real-world scenarios that can occur due to factors such as congestion, failures and backlogs.
- The robust data pipeline is developed using a stream processing engine, Kapacitor, to support real-time decision-making. In this, the feature scaling, clustering, and recommendation operations of core network selection are performed streamingly. We design this pipeline so that it can handle the case of an extract operation failing to return or empty data. For this, we implement the projection module, which is activated to calculate the information on the aircraft. This robust design can allow core network selections to continue without disruption to the service.

1.3 Organization of the Article

The remainder of this article is structured as follows: Section 2 presents the proposed Real-Time Digital Twin Platform, and Section 3 provides the performance evaluations. Then, we conclude the article and provide a discussion on future works in Section 4.

2 REAL-TIME DIGITAL TWIN PLATFORM

In this study, we design the Real-Time DT Platform using a three-layered approach, which includes Physical, DT, and Service Layers as illustrated in the Figure 1. In the design of this platform, we opt for microservice-based architecture over a monolithic one, considering its performance implications [17]. Moreover, monolithic architectures are often built with a single and unified codebase, which may force any change to one part of the system, often requiring redeploying the entire platform. For a real-time DT platform, this would reduce flexibility and increase the risk of failure. Also, improving in a monolithic style and managing and debugging the platform can become complex as the platform grows.

Table 1 Data Retrieved from OpenSky Network API [18]

Field Name	Description
$icao24$	Transponder Address
$callsign$	Callsign of Aircraft
$origin\_country$	Country Name
$time\_position$	Timestamp of Position
$last\_contact$	Timestamp for Update
$longitude$	WGS-84 Longitude
$latitude$	WGS-84 Latitude
$baro\_altitude$	Barometric Altitude (m)
on_ground	If surface position report
$velocity$	Ground Velocity of aircraft (m/s)
$true\_track$	Direction of Aircraft
$vertical\_rate$	vertical velocity of Aircraft (m/s)
$geo\_altitude$	Geometric Altitude
spi	Flight Status
$category$	Aircraft Category

2.1 Physical Layer

The Physical Layer consists of the entities and processes in which their DT is created. In the scope of this article, we consider the core network selection in the AANETs use case and specifically chose the UK area because of its active and diverse air traffic, which carries the main characteristics of AANETS. Accordingly, we retrieve real-time air traffic data from this layer using the OpenSky Network [19]. The details of the data are given in Table 1.

2.2 Digital Twin (DT) Layer

At the DT Layer, the DTs are created and maintained using the real-time data coming from the Physical Layer and their data is stored in the time-series database, InfluxDB [20]. In our DT Platform, this process is handled by the Robust Data Pipeline design, which is responsible for extracting, transforming and loading (ETL) operations and handling the disruption in the data flow.

2.2.1 Robust data pipeline

The real-time core network selection in our DT Implementation is performed according to the following data pipeline as illustrated in Figure 1.

1. **Extract**: The data that the details given in Table 1 is retrieved repetitively from the API. Here, in case the incoming data is empty, the last records on the aircraft are extracted from the time-series database.

2. **Transform**: The incoming data undergoes the following operation steps.

   (a) **Projection:** The timestamp of the position ($time\_position$) in the data may not be the same for each aircraft. To account for that, the location is corrected using the following equations:

   $$\Delta x = velocity * cos(true\_track) * \Delta t \tag{1}$$

   $$\Delta y = velocity * sin(true\_track) * \Delta t \tag{2}$$

   $$\Delta z = vertical\_rate * \Delta t \tag{3}$$

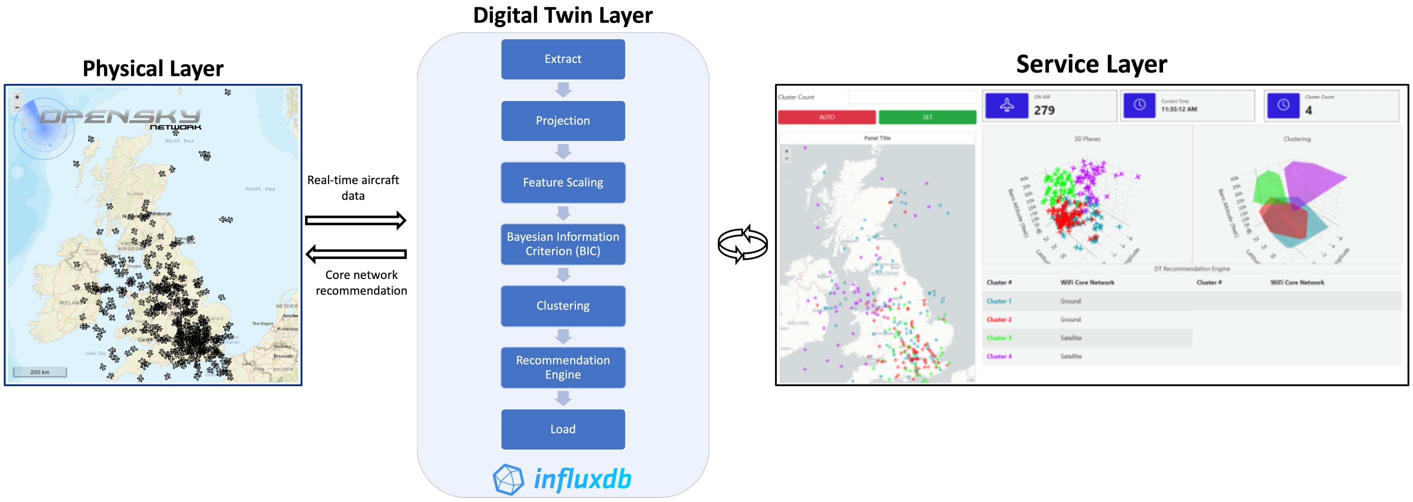
   (b) **Feature scaling:** We scale features to a range of 0 to 1 using the Min-Max Scaler from the Scikit-learn library [21].

   (c) **Bayesian Information Criterion (BIC):** Before clustering, we decide the optimal number of clusters (k) using the Bayesian Information Criterion (BIC) method, which is claimed as a superior alternative to the elbow method [22].

   (d) **Clustering:** We perform clustering using the K-means algorithm thanks to its simplicity and effectiveness in similarity-based clustering. We use the KMeans class from the Scikit-learn library [21] over the scaled data in the previous step.

   (e) **Recommendation:** We perform recommendations for the core network selection based on the methodology introduced in the [12].

3. **Load**: The results are loaded into the time-series database with the buckets Physical and DT. Here, the Physical bucket stores only the data retrieved from the



Cluster #	WiFi Core Network	Cluster #	WiFi Core Network
Cluster 1	Ground		
Cluster 2	Ground		
Cluster 3	Satellite		
Cluster 4	Satellite		

Fig. 1 The three-layered architecture of real-time digital twin platform with robust data pipeline.

API, while the DT bucket stores the data after projection, cluster assignment and the corresponding recommendation. Here, the data is represented as line protocols that are in the following form:

$$measurement\ tag, tag1, \ldots\ field, field3, \ldots\ timestamp \quad (4)$$

2.3 Service Layer

In the Service Layer of our Real-Time Digital Twin Platform, we develop a web application interface using Flask framework [23] and Grafana [24] and Plotly [25]. By this, the monitoring, clustering results, and recommendations are visualised in an interactive dashboard as shown in Figure 1. Within this, aircraft clusters are displayed through three different dashboards in which the aircraft are colour-coded based on their respective clusters. The first dashboard portrays aircraft on a map with their 2D coordinates (longitude and latitude) determining their positions. The second dashboard presents the aircraft clusters with a 3D scatter plot. The third dashboard plots the distinctive 3D coverage areas of aircraft clusters, encompassing all aircraft belonging to each specific cluster. Moreover, the recommendations for the clusters for their core network selection are given.

3 PERFORMANCE EVALUATION

We evaluate the Real-Time Digital Twin Platform by using the latency, clustering accuracy and cluster change rate metrics. Here, the latency is categorized as the preprocessing latency and the decision latency, corresponding to the time taken for the data integration and the decision-making in the recommendation engine. To withstand the dynamic of the environment, the platform should operate with low latency. To evaluate that, we measure the decision latency of our implementation, as shown in Figure 2. Here, as the number of aircraft increases, the delay also increases due to the processed data volume. Moreover, while the latency incurred from just the decision-making process shows similar results to the initial study [12], we observe that the preprocessing of the data mainly dominates the total latency. This is because, as the number of aircraft increases, the data retrieved from the API gets larger linearly. Correspondingly, it is processed for use in clustering. After the cluster decisions are made, the recommendation engine only processes one data point for each aircraft, making the decision latency comparably small to the preprocessing latency. This also showcases the importance of designing an efficient data pipeline.

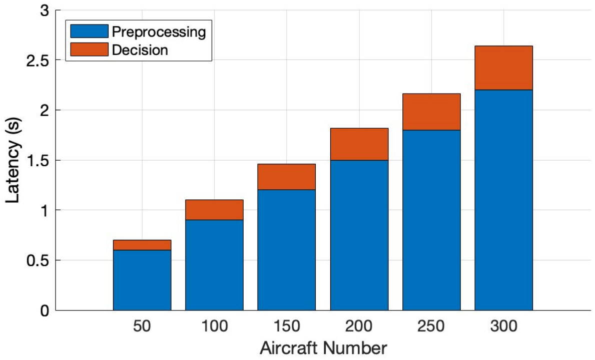


Fig. 2 Evaluation of the latency for preprocessing and decision making.

Moreover, in Figure 3, the accuracy of the cluster assignment when the projection is applied is shown with the corresponding cluster change rate. Here, the clustering accuracy is calculated by using the projection step for an instance of extracted data and comparing the results for the next extraction step. As for the cluster change rate, it is the

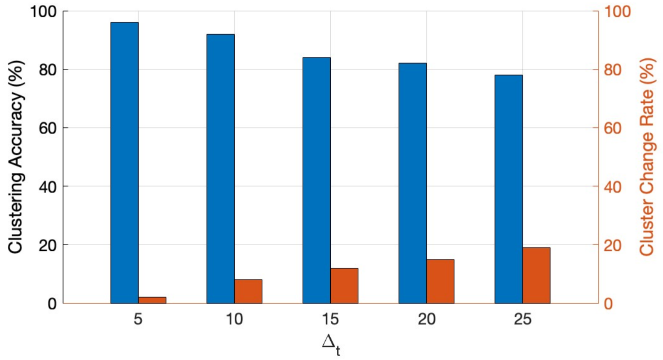


Fig. 3 Evaluation of the data preprocessing pipeline for decision making.

percentage of cluster assignment that has to be changed compared to the baseline. In the core network selection, the cluster change occurs to provide better connectivity for the aircraft, and the projection is applied to ensure that the necessary cluster change decision can be made if the data cannot be retrieved. Here, the results show that the platform can make these decisions thanks to this robust mechanism. However, it is also observed that, as the Δt increases, the accuracy of the results lowers, which is due to Equations 1-3 not considering the aircraft's acceleration and the emergence of the new aircraft. Meanwhile, using projection in further time differences causes a higher cluster change rate. This shows that the wrong cluster assignments affect the decision-making process's effectiveness.

4 CONCLUSION AND FUTURE WORKS

In this article, we implement a real-time digital twin platform for core network selection in aeronautical ad-hoc networks. This platform is built on a microservice-based architecture with three layers: Physical, DT, and Virtual Layers. Moreover, we design a robust data pipeline to integrate real-time aircraft data, handle the missing data, and decide on the core network selection. Then, we develop an interactive user interface using open-source tools. Correspondingly, we analyze the results for latency, accuracy of cluster assignment and cluster change rate. The results showcased that the proposed platform can integrate real-time data and apply decision-making with delays lower than 3 seconds. In this analysis, we revealed that preprocessing operations mainly dominate the system's latency. Considering this latency, we employed a projection step which ensures that the necessary cluster change decisions can be made in time. In future work, we plan to investigate using artificial intelligence (AI) / machine learning (ML) based methodologies at the projection step to improve the clustering accuracy and reduce the cluster change rate.

ACKNOWLEDGEMENTS

This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) 1515 Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab. Project 5239903.

REFERENCES

[1] D. Lehner, J. Pfeiffer, E.-F. Tinsel, *et al.*, "Digital twin platforms: Requirements, capabilities, and future prospects," *IEEE Software*, vol. 39, no. 2, pp. 53–61, 2022. DOI: 10.1109/MS.2021.3133795.

[2] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020. DOI: 10.1109/ACCESS.2020.2998358.

[3] K. Duran, M. Özdem, T. Hoang, T. Q. Duong, and B. Canberk, "Age of twin (aot): A new digital twin qualifier for 6g ecosystem," *IEEE Internet of Things Magazine*, vol. 6, no. 4, pp. 138–143, 2023. DOI: 10.1109/IOTM.001.2300113.

[4] E. Ak and B. Canberk, "Fsc: Two-scale ai-driven fair sensitivity control for 802.11ax networks," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322153.

[5] L. V. Cakir, K. Duran, C. Thomson, M. Broadbent, and B. Canberk, "Ai in energy digital twining: A reinforcement learning-based adaptive digital twin model for green cities," in *ICC 2024 - IEEE International Conference on Communications*, 2024, pp. 4767–4772. DOI: 10.1109/ICC51166.2024.10622773.

[6] D. M. Gutierrez-Estevez, B. Canberk, and I. F. Akyildiz, "Spatio-temporal estimation for interference management in femtocell networks," in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, 2012, pp. 1137–1142. DOI: 10.1109/PIMRC.2012.6362517.

[7] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019. DOI: 10.1109/ACCESS.2019.2953499.

[8] E. Ak, K. Duran, O. A. Dobre, T. Q. Duong, and B. Canberk, "T6conf: Digital twin networking framework for ipv6-enabled net-zero smart cities," *IEEE Communications Magazine*, vol. 61, no. 3, pp. 36–42, 2023. DOI: 10.1109/MCOM.003.2200315.

[9] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021. DOI: 10.1109/JIOT.2021.3079510.

[10] Y. Yigit, B. Bal, A. Karameseoglu, T. Q. Duong, and B. Canberk, "Digital twin-enabled intelligent ddos detection mechanism for autonomous core networks," *IEEE Communications Standards Magazine*, vol. 6, no. 3, pp. 38–44, 2022. DOI: 10.1109/MCOMSTD.0001.2100022.

[11] Inmarsat Aviation, *Inflight connectivity survey – global whitepaper i august 2018*. [Online]. Available: https://www.inmarsat.com/content/dam/inmarsat/corporate/documents/aviation/insights/2018/Inmarsat%20Aviation%202018%20Inflight%20Connectivity%20Survey%20ENG.pdf.

[12] T. Bilen, E. Ak, B. Bal, and B. Canberk, "A proof of concept on digital twin-controlled wifi core network selection for in-flight connectivity," *IEEE Communications Standards Magazine*, vol. 6, no. 3, pp. 60–68, 2022. DOI: 10.1109/MCOMSTD.0001.2100103.

[13] Flightradar, *Live flight tracker - real-time flight tracker map*, en. [Online]. Available: http://FlightRadar24.com.

[14] C. Zhou, H. Yang, X. Duan, *et al.*, *Network digital twin: Concepts and reference architecture*, en. [Online]. Available: https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/.

[15] F. TAO, X. SUN, J. CHENG, *et al.*, "Maketwin: A reference architecture for digital twin software platform," *Chinese Journal of Aeronautics*, vol. 37, no. 1, pp. 1–18, 2024, ISSN: 1000-9361. DOI: https://doi.org/10.1016/j.cja.2023.05.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936123001541.

[16] M. Redeker, J. N. Weskamp, B. Rössl, and F. Pethig, "Towards a digital twin platform for industrie 4.0," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 39–46. DOI: 10.1109/ICPS49255.2021.9468204.

[17] G. Blinowski, A. Ojdowska, and A. Przybyłek, "Monolithic vs. microservice architecture: A performance and scalability evaluation," *IEEE Access*, vol. 10, pp. 20 357–20 374, 2022. DOI: 10.1109/ACCESS.2022.3152803.

[18] *Opensky rest api*, https://openskynetwork.github.io/opensky-api/rest.html, Accessed: 2023-05-19, 2023.

[19] M. Schafer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up opensky: A large-scale ads-b sensor network for research," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, 2014. DOI: 10.1109/IPSN.2014.6846743.

[20] *Get started with influxdb oss 2.7*, https://docs.influxdata.com/influxdb/v2.7/, Accessed: May 21, 2023.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] E. Schubert, "Stop using the elbow criterion for k-means and how to choose the number of clusters instead," *ACM SIGKDD Explorations Newsletter*, vol. 25, no. 1, pp. 36–42, Jun. 2023. DOI: 10.1145/3606274.3606278.

[23] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc., 2018.

[24] Grafana Labs, *Grafana documentation*, https://grafana.com/docs/, Online; accessed July 25, 2019, Jul. 2019.

[25] Plotly Technologies Inc., *Collaborative data science*, 2015. [Online]. Available: https://plot.ly.