Utilizing Distributed Machine Learning Environments for Earthquake Detection

Alexandros S. Kalafatelis*[a], Charis Michailidis[a], Georgios Alexandridis[a], Andreas Oikonomakis[b], Georgios Xylouris[b], Eleni Smyrou[c], Ihsan Bal[c], Urtza Iturraspe Barturen[d], Michail-Alexandros Kourtis[b], Panagiotis Trakadas[a]
[a]National Kapodistrian University of Athens, Psachna 34400, Greece
[b]National Center of Scientific Research "Demokritos", Patr. Gregoriou E 27 Neapoleos Str, 15341 Agia Paraskevi, Greece
[c]Senso Engineering BV, Netherlands
[d]Tecnalia Research Innovation, Astondo Bidea, Edificio 700, 48160 Derio, Biscay, Spain

Abstract. In this work, a novel approach to earthquake detection by integrating deep learning architectures with decentralized data management is introduced. To this end, variational autoencoders are trained within the OASEES framework, employing the InterPlanetary File System for data and model storage, moving beyond traditional centralized cloud/edge processing. The obtained experimental results demonstrate the model's ability in classifying seismic events, achieving an accuracy level of 97.24%. The proposed distributed architecture not only achieves top performance, but also aligns with the heterogeneous cloud-fog-edge computing continuum, offering improved data governance and control.

Keywords: Earthquake detection, Variational Autoencoders, Data Federation, Cloud-edge computing.

* alexkalafat@pms.uoa.gr

This preprint has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this contributions is published in the 2nd International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications (FAIEMA) held in Athens (Greece), 01st - 02nd October 2024.

1 Introduction

Earthquakes are considered to be among the most dangerous natural disasters, with a mortality rate exceeding 50% compared to other natural disasters.[1] Therefore, seismic activity monitoring and prediction are of the utmost importance in protecting people's lives and infrastructure. Currently, earthquakes are measured by relevant agencies around the world, such as the United States Geological Survey and the National Observatory of Athens, which report earthquakes within a time frame of 1 to 30 minutes, depending on their magnitude. The aforementioned agencies are able to identify and characterize earthquakes by analyzing measurements from multiple seismic sensors, scattered around the globe, using tools and methodologies from the scientific field of Statistical Seismology.[2]

However, in recent years, machine learning (ML) approaches have gained momentum, due to their ability to outperform traditional methodologies in terms of efficiency and robust performance.[3,4] Specifically, in the field of earthquake seismology, various models have been developed to detect seismic events, classifying seismic waveforms into two (usually earthquakes and sounds/noise) or more distinct classes. By carefully selecting appropriate training data, classifiers can be trained to detect broad ranges of seismic events, including surface waves,[5] volcanic earthquakes[6] low-frequency earthquakes[7,8] and mining-induced earthquakes.[9,10]

Despite their success, the aforementioned models, trained on unified datasets one a single machine (an approach also known as centralized machine learning - CML), also exhibit certain limitations,[11,12] the most important of which is their inability to process data streams from a continuously evolving set of seismic sensors, geographically dispersed around an earthquake epicenter, as depicted in Figure 1. Therefore, distributed computing approaches at the edge[13] could mitigate this effect, albeit currently available solutions revolve around centralized clouds, limiting data governance and identity management, even though the future of cloud computing is expected to be distributed and heterogeneous.[14]

For the task at hand, a holistic solution would be ideal, encompassing the entire computing continuum, from central infrastructures to smart edge devices. Nevertheless, there is a lack of open management frameworks, and commercial hybrid core/edge management solutions, like Azure Stack Edge,[16] are closed and not suitable for private use. Another gap to be addressed deals with edge orchestration solutions. While public clouds offer user-friendly interfaces, enabling data scientists to prioritize on data management and AI model optimization rather than dealing

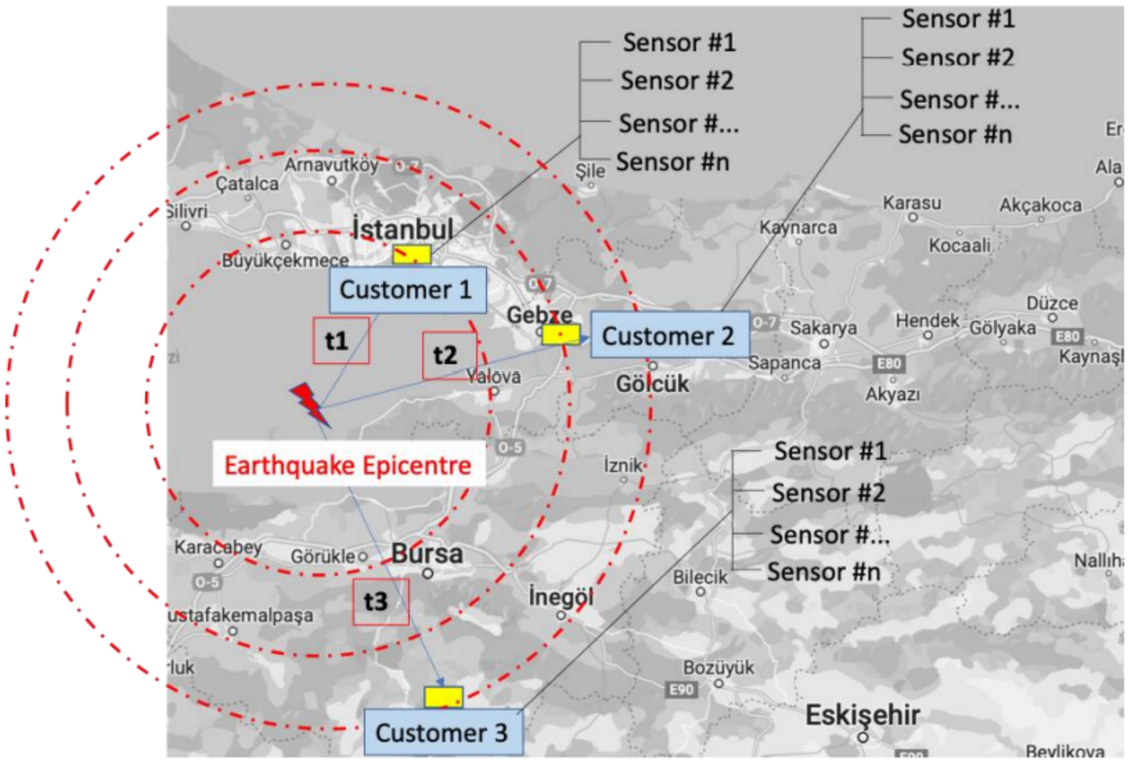

Fig 1: Geographically distributed seismic sensors around an earthquake epicenter[15]

with the management of physical and virtual resources required for these tasks, edge orchestration solutions available today do not include this functionality.[17] Moreover, the aspect of security is also vital, as edge infrastructures can be dynamic and require continuous verification of services and infrastructure throughout the continuum.

In this respect, the current work aims to facilitate the deployment of ML and deep learning (DL)-based earthquake detection models, by utilizing distributed learning approaches over the Open Autonomous programmable cloud appS & smart EdgE Sensors (OASEES) framework.[18] The OASEES framework, which is going to be described in detail in Section 2, is a fully open-source, decentralized and secure Swarm programmability framework for edge devices, leveraging various artificial intelligence and ML accelerators (such as spiking neural networks and quantum computing devices), while at the same time supporting a privacy-preserving Object ID federation process. In the context of the current work, extensive experimental testing is conducted in order to assess the performance of a DL-based seismic detection model based on variational autoencoders (VAEs) and determine the practicality of integrating models and data into distributed file systems, leveraging OASEES' capability for creating data and model products, as well as for deployment purposes.

The remainder of this paper is organized as follows; Section 2 provides a comprehensive overview of the OASEES framework, covering the processes of data and model integrations into the InterPlanetary File System (IPFS)[19] and Gaia-X, as well as the product creation and deployment processes Section 3 describes the experimental methodology, including details about the dataset and the model architecture. Section 4 discusses the findings of the study, while Section 5 concludes the paper and summarizes the results, also identifying potential areas for future research.

2 The OASEES Framework

Recent advances in edge computing have led to the development of various infrastructure and service management platforms, including open-source options. To fully harness the benefits of edge computing, a comprehensive strategy that integrates intelligent devices with central systems is essential. Despite the distributed and diverse nature of cloud computing, there is a notable absence of open management frameworks.

The OASEES project[18] aims to develop a pioneering programmability framework using state-of-the-art technologies such as AI/ML accelerators (like spiking neural networks and quantum computing devices), along with a privacy-aware Object ID federation technique. This framework will facilitate decentralized and secure collaboration among edge devices by leveraging Blockchain technologies to ensure data integrity, enhance security, utilizing IPFS[19] to enable efficient, distributed and resilient data storage and sharing, further enhancing the robustness and scalability of edge computing environments.

The OASEES framework offers a complete solution for creating, training, and implementing ML/DL models, while also supporting secure data exchange across different areas within an organization. In addition, the framework fully complies/is compatible with the Gaia-X federation[20] and the International Data Space Association directives and specifications.[18] Key features of the framework are showcased in Figure 2 and include:

- **Decentralized Management:** The OASEES framework allows organizations to manage their infrastructure clusters without relying on a centralized authority. This decentralized approach enhances security, reduces single points of failure and ensures greater control over data and resources.

- **ML and DL Model Development:** Organizations can use the OASEES framework to develop, train, and deploy their ML/DL models efficiently.

- **Private IPFS Network:** The framework incorporates IPFS, creating a dedicated private network for secure and efficient data sharing within the organization. This enables teams to store and share large datasets necessary for training ML and DL models, ensuring data to be readily accessible across different regions, while maintaining privacy and security.

- **OASEES SDK:** A set of utilities that allow the seamless installation of all the necessary components. Also, it provides all the tools to develop, build and deploy ML/DL models.

- **OASEES Blockchain Marketplace:** The framework includes the OASEES marketplace, where organizations can publish and exchange data products such as ML/DL models, as

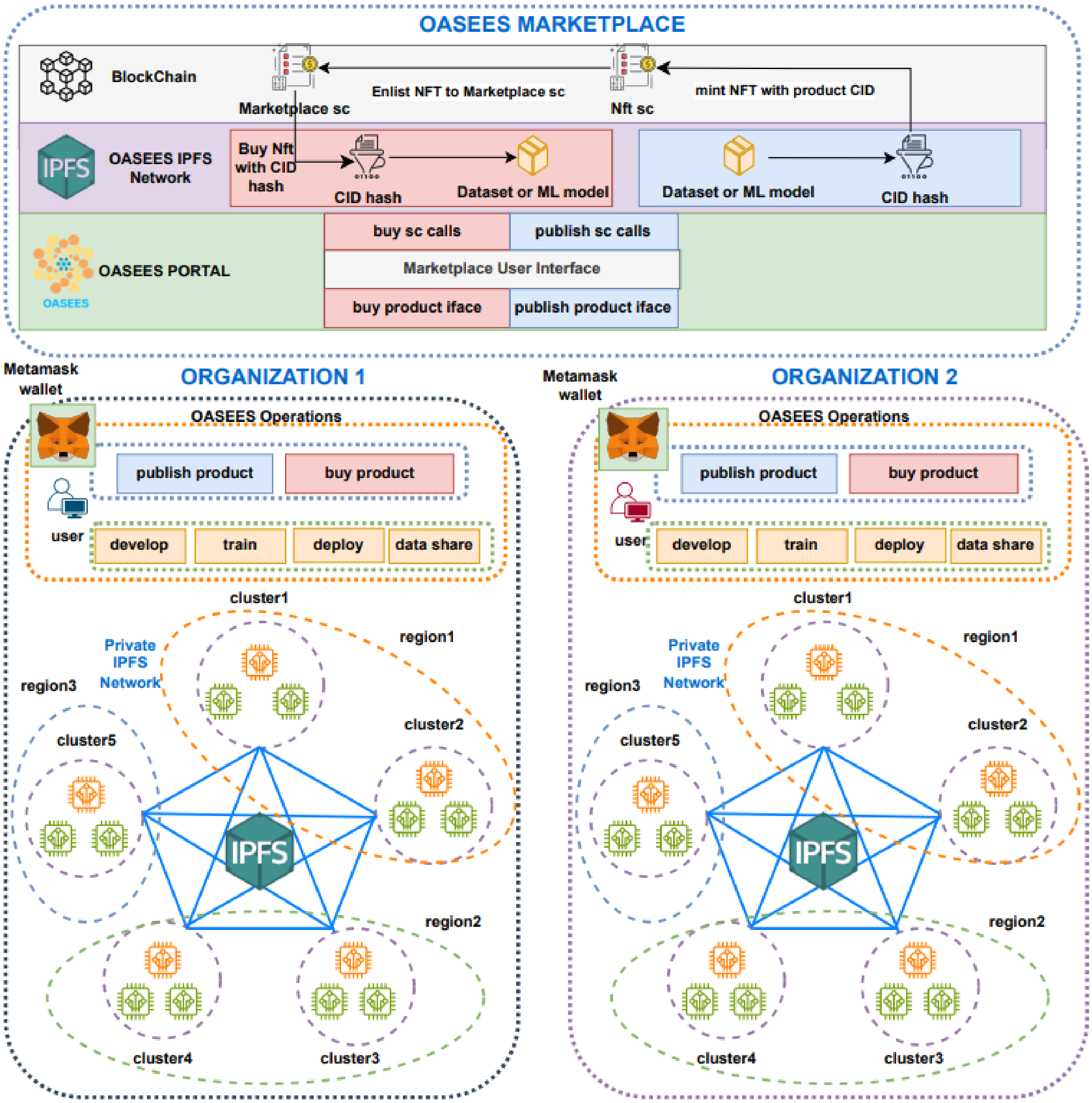

Fig 2: Key features of the OASEES framework

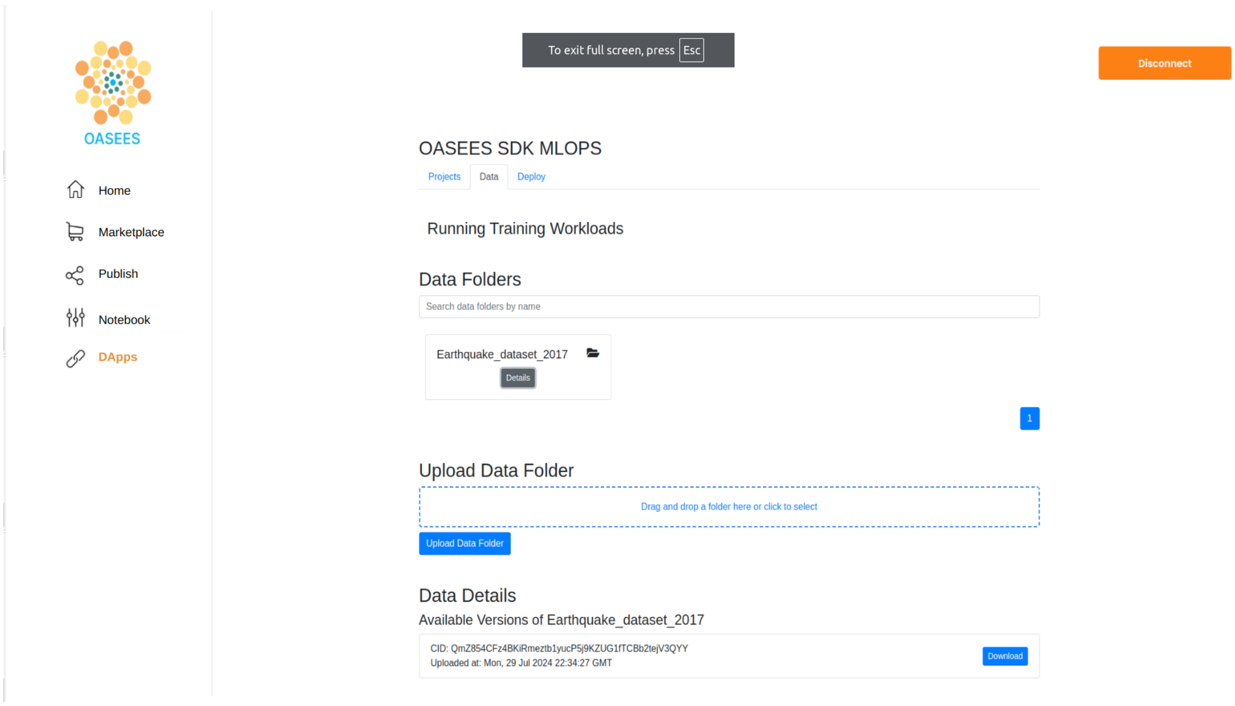

Fig 3: Uploaded dataset within a private IPFS network.

well as datasets. Leveraging blockchain technology, these data products are published as non-fungible tokens (NFTs),[21] ensuring data integrity and supporting transparent, tamper-proof transactions. Users can log in to the marketplace using their Web3 wallets, such as MetaMask,[22] facilitating secure and decentralized transactions.

Users within an organization can upload datasets using the shared management User Interface (UI), which is designed to facilitate inter-organizational collaboration, as showcased in Figure 3. To upload a dataset, a user navigates to the management interface and utilizes the provided upload feature to select their dataset. Once the dataset is uploaded, it is sent to the private IPFS network (which, being a decentralized storage solution, ensures that the data are securely distributed and accessible) and a Content Identifier (CID) is generated. This CID is a unique hash that serves as a pointer to the dataset within the IPFS network, ensuring both the integrity and ease of retrieval of the data. The generated CID is then made available within the management UI, allowing all users

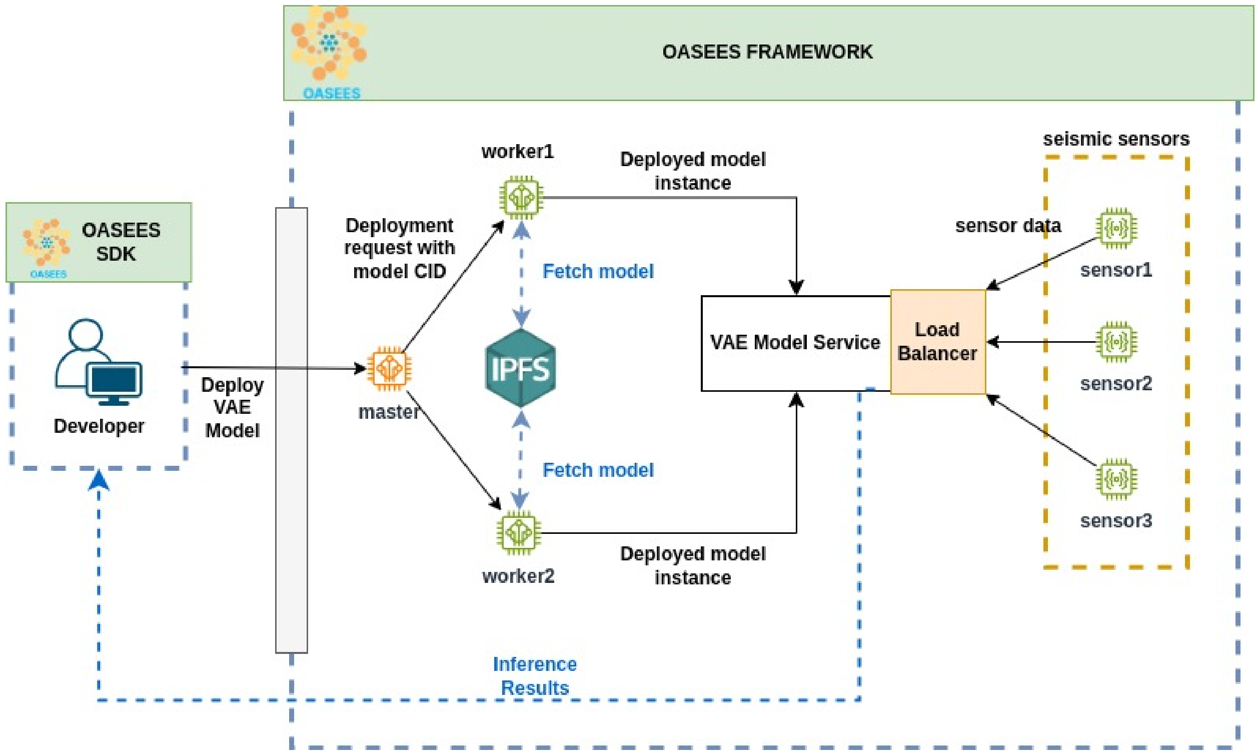

Fig 4: Integration of a VAE model into OASEES.

within the organization to access and use the dataset. This seamless process ensures efficient data sharing and collaboration, backed by the security and reliability of the IPFS .

Following dataset upload, the OASEES Python software development kit (SDK) can be employed to develop and train ML/DL models. The SDK streamlines the model development process, making it easy to implement ML/DL algorithms. Once a model is developed and training is completed, it is stored on IPFS and is automatically made available and accessible via the management UI. The model can then be deployed through the OASEES SDK (Figure 4), either to a specific worker node or across multiple nodes for scalability. Upon deployment, the model is retrieved from IPFS and instantly deployed as a service, capable of processing incoming sensor data and providing inference results. Additionally, users can easily retrain the model using new data, again through the management UI.

This process involves uploading new datasets and employing the OASEES Python SDK once

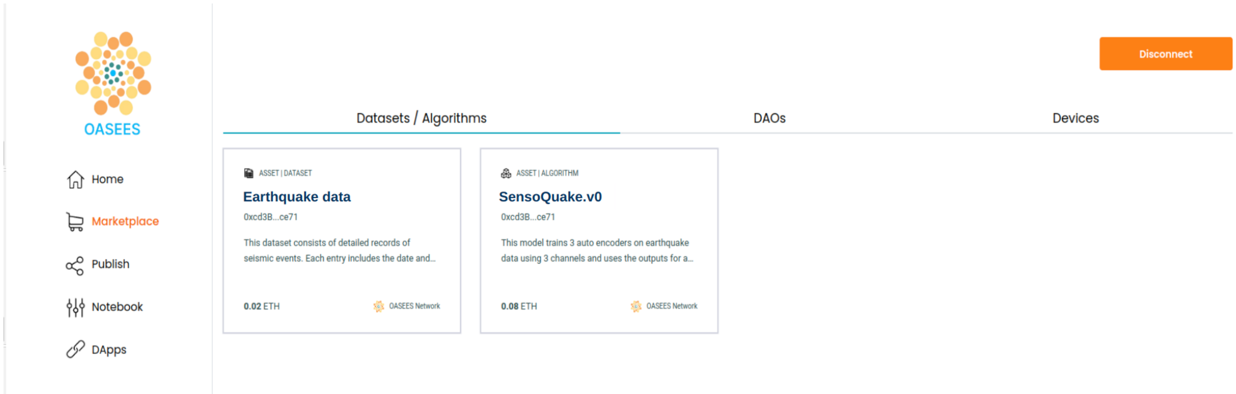

Fig 5: Trained model and datasets as products.

again, to refine the model. This capability allows for continuous improvement and adaptation of the model, based on updated data, fostering an environment of ongoing development and innovation within the organization. Each trained model is also accompanied with all the necessary files in order to be deployable, including model parameters, source files that load the model and enable its inference mode and finally a Dockerfile that allows the building of the model as a containerized application.

If a user within an organization wishes to publish a model or a dataset, they can use the OASEES marketplace. To begin, the user logs in with their Web3 wallet, such as MetaMask,[22] providing a secure and decentralized authentication method. Once logged in, the user can upload the assets they wish to sell to the OASEES IPFS network. After uploading the assets, the user can mint an NFT that contains the CID of the asset, which ensures that it is uniquely identified and can be securely accessed. The minted NFT is then listed on the OASEES marketplace, where it becomes available for purchase by other users. Furthermore, users of the OASEES marketplace can browse and buy these assets (Figure 5). For example, trained models, once purchased, are ready to be deployed by the buyer. Furthermore, if the purchaser possesses similar data, he or she can retrain the purchased model to improve its performance or adapt it to his/her specific needs.

Finally, after data integration and product creation, the Gaia-X[20] onboarding process starts. OASEES hosts these resources on IPFS and provides a management API to make access and control easier. This critical step involves minting verifiable credentials for all participants, including the legal entity, the data product provider and transforming the product itself into a Gaia-X service offering. This meticulous process ensures that all facets are defined, documented and ready for consumption. Once defined, the data product is catalogued and made available for consumption. The Eclipse Dataspace Connector (EDC)[23] infrastructure safeguards data integrity and policy compliance, ensuring secure data exchange. To participate in this secure data exchange, consumers must also utilize an EDC connector. Once the storage process is completed and the physical URL of the resource offered within the data product is known, the next step involves defining the Gaia-X data product and publishing it as an available asset within the EDC. This process involves two steps; (i) obtain an SSL certificate, which must be valid for one of the Trust Anchors defined in Gaia-X and (ii) utilize the Gaia-X Wizard to create self-descriptions for entities, participants, service offerings, and data resources.[24] After the data products have been registered in the ecosystem, they may be accessed and utilized using an EDC connector that interacts with the OASEES EDC connector. In order to simplify this process, OASEES includes an EDC consumer connector and a user interface that enables server-side rendering for fast and user-friendly interaction with the data products. In detail, the steps to consume the data from the catalogue include i) retrieving the Data Product Identifier, ii) negotiating a contract for data usage, iii) obtaining the identifier for the agreed contract and iv) initiating the data transfer process. The complete procedures of data product storage, Gaia-X catalog registration, and consumption are depicted in Figure 6.

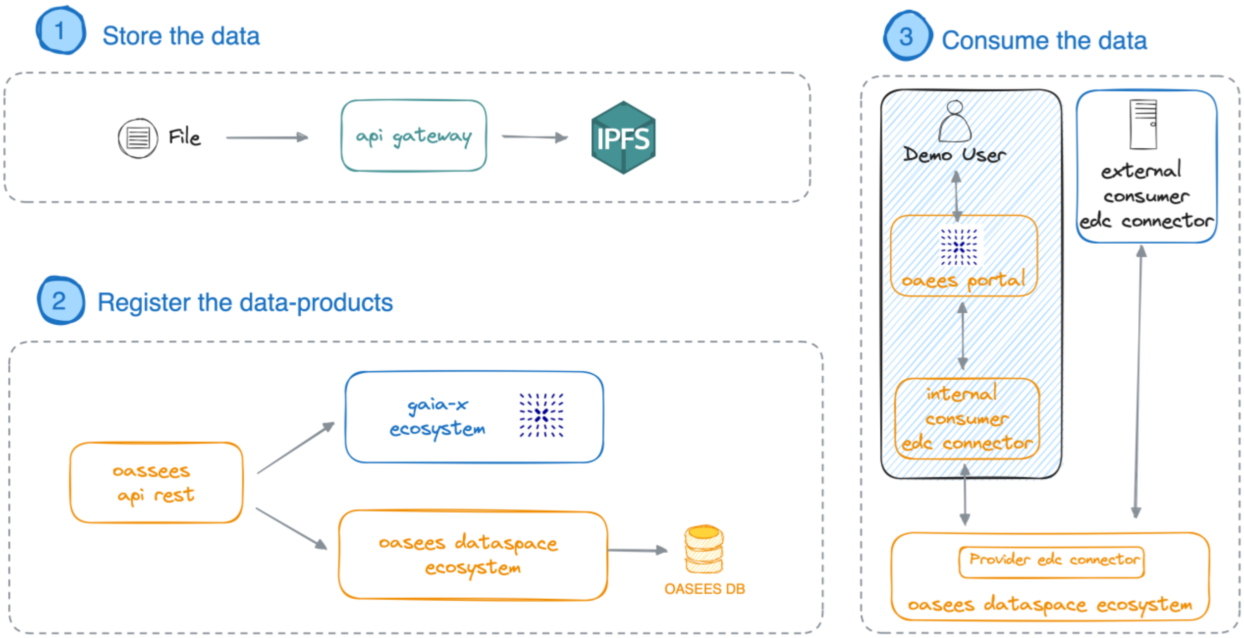

Fig 6: High-level steps for enabling data product sharing with Gaia-X.

3 Methodology

In this work, a publicly available dataset of accelerometer recordings from the Groningen region in Northern Netherlands has been utilized, an area prone to small earthquakes induced by the deep underground gas extraction activities.[25] The dataset was selected for two primary reasons; i) the small magnitude earthquakes in Groningen, producing low amplitude recordings that can easily be confused with the noise or other types of vibrations, and ii) the region's dense seismic network providing high-resolution data due to the gas extracting company's liabilities and public interest.

The dataset is comprised of 3 channel readings (East, North, and Up) from approximately 100 surface accelerometers, resulting in about 300 channels of data per event. Four event types are defined, earthquakes (including small shallow earthquakes, occurring at the depth of approximately 3km, with a magnitude of $2 - 3.5$ Richter), strong storms, rush hour (around 17:00PM) and silent hour (around 03:00AM). Each event is represented by 3-minute recordings, totaling $1,297$ files with roughly 300 files per event, ensuring a balanced class distribution.

In this study, VAEs with a residual network architecture have been employed for earthquake detection, due to their efficiency in capturing complex patterns in data. VAEs are comprised of an encoder and a decoder network, both equipped with residual blocks that enhance the learning capability and feature representation. More specifically, the encoder network $q(z|x)$ maps input data x to a latent representation space z. Encoder architecture includes a series of one-dimensional convolutional layers, residual blocks and fully connected (FC) layers. Each residual block consists of two convolutional layers with batch normalization and skip connections to facilitate the training of the deep networks. The encoder ends with two FC layers that output the mean μ and log-variance $\log \sigma^2$ of the latent variables. The latent variable z is sampled using the reparameterization trick of Equation 1

$$z = \mu + \exp\left(\frac{1}{2}\log \sigma^2\right) \cdot \epsilon \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, I)$ represents standard normal noise.

Afterwards, the decoder network $p(x|z)$ utilizes a series of transposed convolutional layers and residual blocks to reconstruct the input data from the latent space. It begins with a FC layer that transforms the latent variables into a high-dimensional feature map. This is followed by a series of transposed convolutions to upsample and reconstruct the original data dimensions, passed through a softmax function. The VAE is trained by minimizing the objective function of Equation 2

$$\text{Loss} = \text{BCE}(x, \widetilde{x}) + \text{KL}(q(z|x) \parallel p(z)) \tag{2}$$

where $\text{BCE}(x, \widetilde{x})$ is the binary cross-entropy loss between the input x and the reconstruction $\widetilde{x}$ and $\text{KL}(q(z|x) \parallel p(z))$ is the Kullback-Leibler divergence between the learned distribution $q(z|x)$ and the prior $p(z)$. The VAE was trained for 20 epochs, with a learning rate of 3×10^{-4}, using the Adam optimizer and a batch size of 32.

After VAE training, the encoder part is utilized to extract features from the input data, which are subsequently processed by a classification network for earthquake detection. The encoder component, whose weights are now frozen, processes input data from three channels (ch_1, ch_2, and ch_3), each with a shape of $\mathbb{R}^{32\times1\times24000}$, producing three latent representations z_1, z_2 and z_3 for each channel, respectively. These representations are the concatenated into a single latent vector z, to be subsequently processed by a series of two FC layers with ReLU activation functions. Finally, the network's output layer uses the softmax function, producing class probability distributions.

The classification network was trained with a learning rate of 3×10^{-4} using the Adam optimizer for 20 epochs, with a batch size of 32, using BCE with logits as the loss function. To effectively train and test the proposed model architecture, the dataset was split intro two subsets (train & test), over an $80\%/20\%$ ratio. Finally, to evaluate the performance of our classification model, we utilized standard information retrieval metrics such as accuracy, precision, recall and F1-score for each class. The architecture of the proposed model, which integrates both the VAE and the classifier, is illustrated in Figure 7.

4 Experimental Results

Simulation and algorithmic procedures were performed on a computer equipped with an AMD Ryzen 5 5600H 3.30 GHz processor and a NVIDIA GeForce RTX 3050 GPU with 16 GB of RAM. The model described in the previous Section was implemented in PyTorch.

Table 1 provides a thorough assessment of the models performance in seismic detection. Specifically, the model scored a total accuracy of 97.24%, suggesting a high-level ability in classifying

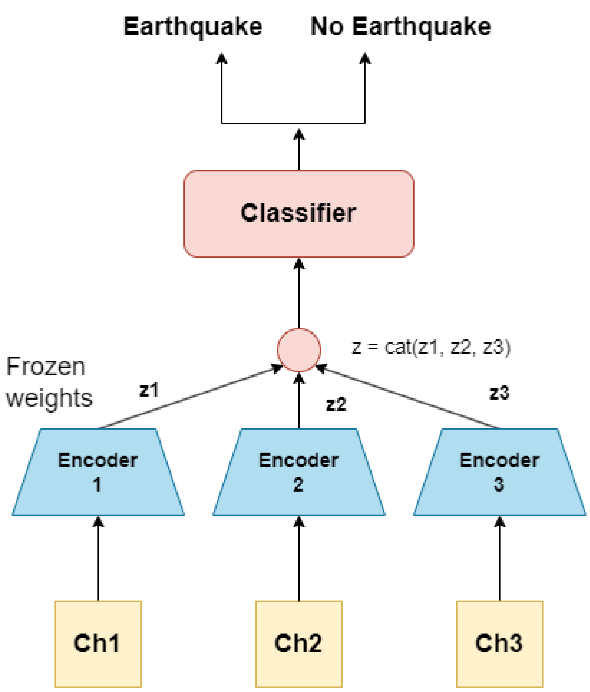

Fig 7: High-level architecture of the proposed model

Table 1: Performance metrics for each class.

Class	Precision	Recall	Accuracy	F1 Score
Earthquake	96.30%	91.23%	91.23%	93.69%
No Earthquake	97.50%	98.98%	98.98%	98.24%

earthquake and non-earthquake events. In addition, the F1-score takes into account both precision and recall, with the earthquake class having an F1-score of 93.69% and the non-earthquake class an F1-score of 98.24%, respectively. These findings are indicated of the model's robustness and ability in distinguishing earthquake from non-earthquake events.

5 Conclusions

In this study, a novel DL approach for earthquake detection, running on a distributed data architecture has been presented. A distinguishing feature of the proposed solution, is the use of VAEs along with a classification model, leveraging the latent space features coming from the VAE, to accurately classify seismic events and differentiate them from noise. In addition, the presented solution demonstrates the use of the OASEES framework for model deployment and IPFS for data and model storage.

The findings of the current work affirm the sought-out solution, showcasing the model's ability in accurately classifying both earthquake and non-earthquake events. Notably, the model's high precision is crucial for applications where missing an event could have significant consequences. Additionally, the slight imbalance in false predictions (i.e., due to higher number of false positives compared to false negatives), suggest a cautionary approach, by predicting earthquakes more readily, possibly due to the higher cost of missing an earthquake.

Regarding scalability, the OASEES framework can seamlessly adapt as new devices are onboarded using the OASEES SDK. Data processing latency across distributed environments can also be mitigated, by integrating sensors directly into worker nodes, introducing edge computing to process data closer to the source, minimizing the need for data transfers to central servers. To further enhance system robustness under varying network conditions, real-time network monitoring tools could be employed to evaluate factors like latency, packet loss, and jitter.

Future research should focus on several key areas to further enhance the system's performance and scalability. First, training models using larger operational datasets could improve temporal coverage and enhance model generalization ability. The proposed architecture could also be adapted for multiclass classification, enabling deeper insights not only for seismic events but for the different types of noise recorded. Additionally, incorporating decentralized AI training schemes, could further address data privacy issues of CML, while exploring dynamic client selection policies for worker nodes participation in training rounds, could optimize resource allocation, balancing computational load and reducing network bottlenecks, further improving the efficiency and scalability of the system.

Finally, the utilization of ML in earthquake detection holds significant potential and could revolutionize the use of relevant data, towards the minimization of response times for natural disaster management operations.

6 Acknowledgments

This research initiative is supported by the European Union's Horizon Europe Framework Programme for Research and Innovation, under the OASEES project (Grant Agreement No. 101092702). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union, which cannot be held responsible for them.

References

1 D. Muhammad, I. Ahmad, M. I. Khalil, *et al.*, "A generalized deep learning approach to seismic activity prediction," *Applied Sciences* **13**(3), 1598 (2023).

2 D. Vere-Jones, "Foundations of statistical seismology," *Pure and Applied Geophysics* **167**, 645–653 (2010).

3 S. Li, X. Yang, A. Cao, *et al.*, "Seist: A foundational deep learning model for earthquake monitoring tasks," *IEEE Transactions on Geoscience and Remote Sensing* (2024).

4 Z. E. Ross, M.-A. Meier, E. Hauksson, *et al.*, "Generalized seismic phase detection with deep learning," *Bulletin of the Seismological Society of America* **108**(5A), 2894–2901 (2018).

5 C. Chai, J. Kintner, K. M. Cleveland, *et al.*, "Automatic waveform quality control for surface waves using machine learning," *Seismological Society of America* **93**(3), 1683–1694 (2022).

6 A. B. Rodriguez, C. Benítez, L. Zuccarello, *et al.*, "Bayesian monitoring of seismo-volcanic dynamics," *IEEE Transactions on Geoscience and Remote Sensing* **60**, 1–14 (2021).

7 A. M. Thomas, A. Inbal, J. Searcy, *et al.*, "Identification of low-frequency earthquakes on the san andreas fault with deep learning," *Geophysical Research Letters* **48**(13), e2021GL093157 (2021).

8 K. H. Chen, H.-Y. Chiu, K. Obara, *et al.*, "Segmentation characteristics of deep, low-frequency tremors in shikoku, japan using machine learning approaches," *Earth, Planets and Space* **75**(1), 32 (2023).

9 T. Wang, Y. Bian, Y. Zhang, *et al.*, "Using artificial intelligence methods to classify different seismic events," *Seismological Society of America* **94**(1), 1–16 (2023).

10 H. Kubo, M. Naoi, and M. Kano, "Recent advances in earthquake seismology using machine learning," *Earth, Planets and Space* **76**(1), 36 (2024).

11 A. Sarlas, A. Kalafatelis, G. Alexandridis, *et al.*, "Exploring federated learning for speech-based parkinson's disease detection," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, *ARES '23*, Association for Computing Machinery, (New York, NY, USA) (2023).

12 A. Giannopoulos, P. Gkonis, P. Bithas, *et al.*, "Federated learning for maritime environments: Use cases, experimental results, and open issues," *Journal of Marine Science and Engineering* **12**(6) (2024).

13 W. Shi, J. Cao, Q. Zhang, *et al.*, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal* **3**(5), 637–646 (2016).

14 W. John, C. Sargor, R. Szabo, *et al.*, "The future of cloud computing highly distributed with heterogeneous hardware," *Ericsson Technology Review* **2020**(5), 2–13 (2020).

15 P. D'Ostilio, A. Fontalvo, D. Iglesias, *et al.*, "Oasees d2.1 overall architecture and use case definition," techreport, A.S.M. Terni S.p.A. (2023).

16 "Azure stack edge." online (2024).

17 M. A. Kourtis, N. Tcholtchev, I.-D. Gheorghe-Pop, *et al.*, "Towards continuous development for quantum programming in decentralized iot environments," *Procedia Computer Science* **238**, 7–14 (2024).

18 I. P. Chochliouros, M. A. Kourtis, G. Xilouris, *et al.*, "Oasees: An innovative scope for a dao-based programmable swarm solution, for decentralizing ai applications close to data generation locations," in *Artificial Intelligence Applications and Innovations. AIAI 2023 IFIP WG 12.5 International Workshops*, I. Maglogiannis, L. Iliadis, A. Papaleonidas, *et al.*, Eds., 91–105, Springer Nature Switzerland, (Cham) (2023).

19 Y. Psaras and D. Dias, "The interplanetary file system and the filecoin network," in *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, 80–80 (2020).

20 A. Braud, G. Fromentoux, B. Radier, *et al.*, "The road to european digital sovereignty with gaia-x and idsa," *IEEE Network* **35**(2), 4–5 (2021).

21 B. Hammi, S. Zeadally, and A. J. Perez, "Non-fungible tokens: A review," *IEEE Internet of Things Magazine* **6**(1), 46–50 (2023).

22 W.-M. Lee, *Using the MetaMask Crypto-Wallet*, 111–144. Apress, Berkeley, CA (2023).

23 V. Siska, V. Karagiannis, M. Drobics, *et al.*, "Building a dataspace: Technical overview," *Gaia-X Hub Austria* (2023).

24 GAIA-X, "Gaia-x self-description creation wizard."

25 I. E. Bal, “Earthquake wavelets,” Hugging Face (2024).