Country-Wide Digital Twin for Humanitarian AI: The Case Study of Cambodian Mine Action

Gürkan Solmaz[1], **Flavio Cirillo**[1], **Sébastien Pierrel**[1,2]

[1]NEC Laboratories Europe, [2]Kristianstad University

guerkan.solmaz@neclab.eu, sebastien.pierrel@gmail.com, flavio.cirillo@neclab.eu

Abstract

Landmine contamination poses a significant threat in post-conflict regions, endangering civilian populations even decades after hostilities end. Demining efforts are resource-intensive, and prioritizing areas for clearance is particularly challenging in regions with widespread contamination. This paper presents a *country-wide Digital Twin* for landmine risk assessment, addressing key challenges: (a) modeling of a country-wide Digital Twin, (b) data integration, enrichment and augmentation, and (c) efficient machine learning (ML) training. The system is deployed in Cambodia premises, where it streamlines the development of ML workflows and achieves up to 91% accuracy in assessing risks of landmine areas.

1 Introduction

Landmines put the population of many countries at risk decades after conflicts take place. The effort to rehabilitate land contaminated by *explosive remnants of war (ERW)* is a slow process. Large portions of land are contaminated, and defining priorities on which area to clear first is difficult. The existing landmine assessment tool Desk-AId (Anonymous 2024) implements a ML pipeline for landmine risk assessment using landmine data sources combined with Geographic Information System (GIS) technologies to achieve high-accuracy prioritization. However, the existing tools require manual preparation of the data through a QGIS user interface. This makes it difficult to integrate with new data sources, hindering the scalability and reproducibility in other contexts.

Nowadays intelligent systems adopt Digital Twin (DT), which is a virtual representation of a real-world asset in the digital world. The DT approach, originally adopted for machinery management, is applied in new domains such as smart cities (Mandal 2024; Mandal and O'Connor 2024; Jafari et al. 2023), healthcare (Sun, He, and Li 2023; Mahmud et al.

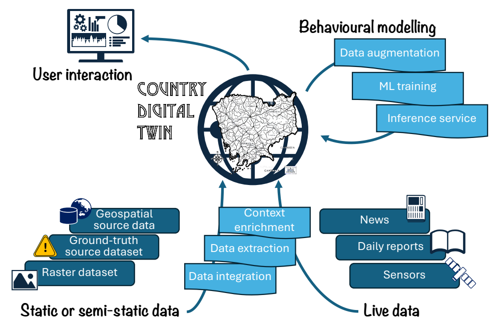

Figure 1: The concept of country-wide DT with data processing and behavioral modeling.

2025), data centers (Sarkar et al. 2024) and networks (Almasan et al. 2022). A DT puts together multiple views of the same real asset from multiple domains, thus, allowing an holistic approach and better intelligence. We embrace the DT for replicating a geographical area such as a country (see Fig. 1).

The realization of a DT is a complex task since it involves the integration of heterogeneous data, different types of analytics processes and the orchestration of the data processing flow. The DT allows users to specify *what* needs to be achieved, rather than *how* to achieve it. It requires minimal input from the user, as the system leverages contextual information from DT data to determine the appropriate operations. This language is interpreted by an analytics orchestration system driven by context. We use and adopt FogFlow (FIWARE 2024a; Cheng et al. 2017), a FIWARE serverless computing programming model and open-source framework.

For the use case of mine action, a country-wide DT includes many datasets such as geographical data (e.g., elevation, water ways, coastline) and socio-economic data (population density, buildings, facilities, national borders). Starting from these data, there can be a set of behavioural models (such as landmine risk assessment model) in various configurations (different sets of features and data augmentation techniques). Finding and configuring an ML model in earth intelligence is tricky and often requires extensive experimentation and comparison (Sun et al. 2022). Therefore, it is of great help to base the development of the Digital Twin on a low-code development platform (Dalibor et al. 2022; Michael and Wortmann 2021). The contributions of this paper are listed as follows.

- Design of a country-wide DT application with data and behavioural models. The data model is based on the object-centric NGSI-LD (ETSI CIM 2024).
- Integration, enrichment, and augmentation of the geographic datasets together with a ground-truth of confirmed landmine areas.
- Extensive ML training and experiments through a wide range of datasets and ML models. The models perform up to 91% accuracy.
- The application is deployed and used by the leading mine agency, Cambodian Mine Action Center (CMAC). We share datasets and code for the reproducibility by our research community.

2 Related Work

The term DT was coined in the early 2000s to describe a system made of a physical object, its virtual counterpart, and the relationship between the two. DTs are often referred in the context of smart cities, with a few instances of larger regions in the literature: (Sottet and Pruski 2023) presents an aggregation of smart cities' DTs to a Nation-wide DT, (Jamil et al. 2023) describes the use of a nation-wide (Cyprus) DT for environmental risk for real estate. The country-wide DT is used to describe the implementation of a Metaverse and Virtual Reality versions of a country such as in (Logothetis, Mari, and Vidakis 2023). Yet, this type of DT does not offer ML functionality.

There are various studies that address the DT management in different domains. In Industrie 4.0, where the cyber-physical system approach is well defined in standards and extensively applied, systems such as (Redeker et al. 2022) focus on continuous deployment management and data infrastructure, aiming to reduce the effort of the design, storage and deployment of DT. Intent-based approach is applied on network management (Leivadeas and Falkner 2022) to simplify the operations through iteratively expanding a human readable command (an intent) to composite multiple commands into the complex network system. For intent-based application execution lifecycle, (Dzeparoska et al. 2023) proposes executing applications among computing nodes where commands written in natural language are interpreted and processed by an LLM engine. Low-code development platforms (Dalibor et al. 2022; Talasila et al. 2023) allow abstracting the DT development where a developer composes the system using library of components. However, the research in this area is still at the early stage (Pinho, Aguiar, and Amaral 2023). The recent study (Avishahar-Zeira and Lorenz 2023) defines how individuals with domain-knowledge but low-programming skills can develop smart city applications. LLMs can also be utilized for smart city and geospatial knowledge graphs (Mandal and O'Connor 2024; Zhang et al. 2025). These studies address the development of application logic from available data (such as sensor data) however they do not facilitate the development of ML workflows handling dataset sampling, augmentation and enrichment, especially tricky for earth intelligence (Sun et al. 2022).

Several recent works explore the feasibility of risk assessment for supporting the on-field landmine clearance operations (Dulce Rubio et al. 2024; Anonymous 2024; Rafique et al. 2019; Schultz et al. 2016). These studies focus only on the performance of a specific ML model, mainly on offline analytics as ad-hoc data science solutions. Training a new model is a tedious process, involving expertise of GIS tools such as ArcGIS or QGIS. Lacking a DT data infrastructure hinders new data integration or easy application to new regions. We propose a DT system to support easy-development and deployment of earth intelligence (Sun et al. 2022) and apply it in the Cambodian Mine Action case study.

3 The Digital Twin System

We design a country-wide DT system for its application for landmine risk assessment. The goals of the system are to: 1) generate many datasets through automated data integration, enrichment, and augmentation; 2) train and deploy specialized ML models; 3) support local mine action agency with the generated data-centric insights (Reuters 2025) such as probabilities of landmine presence in given areas.

The implementation of a DT application requires several steps. The workflow starts from learning the application domain, and then the acquisition of relevant datasets. The datasets are heterogeneous and available from different sources such as industrial databases, open data, or sensor deployments. The discovery of the data is crucial for an optimal end result. The next stage is the modeling of the DT such as integrating the heterogeneous data, modeling the required preprocessing and the services to realize the DT application. Once the DT model and application are ready, the developer implements the functions to generate the behavioural models. A detailed workflow execution plan with concrete functions instances (there might be more instances of the same function for different elements of the DT) and in-

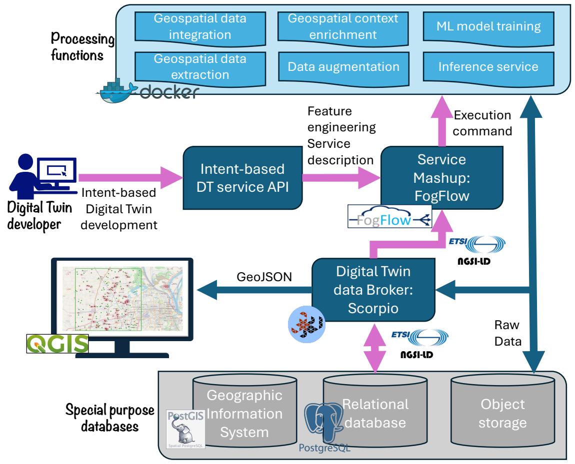

Figure 2: Implementation of the DT system using open source software (FIWARE Scorpio Broker and FogFlow), open standards (NGSI-LD) and execution environment (Docker).

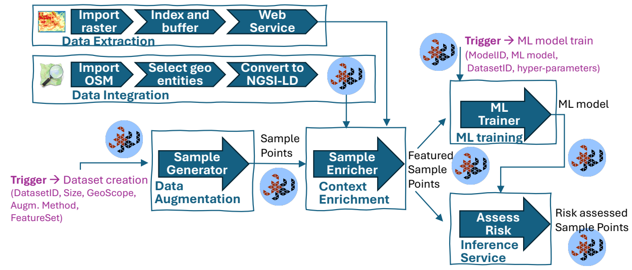

Figure 3: Overview of the automated data workflow and the role of the processing functions managed by FogFlow.

put/output workflow among the functions define the actual application deployment. Finally, the full service is deployed and its lifecycle is managed to ensure correctness and continuous service operation. In our approach, we automatize many of these steps including the data discovery, data integration, data preprocessing and behavioural model generation.

Our proposed system is composed of few components to address the DT development stages in an automated fashion. At the core of the system there are two elements: 1) A data handling component, namely a DT data broker, and 2) a service mashup component that encompasses the data processing and behavioural model execution. The data model of the DT is the open standard NGSI-LD (ETSI CIM 2024) specified by ETSI. NGSI-LD is an object-oriented information model. The information model includes: *entity* as a unique id that refers to the real-world object, *attribute* as a value referred to the object (either a direct observation or generated inference), and *relationship* that links DT instances between each other. Being based on JSON-LD, NGSI-LD includes *context* element that backs up the information with dictionaries, improving universal meaningfulness of the data in the system.

The *DT data broker* offers capabilities to discover and provide data with different filtering options such as geographic scoping, value matching on linked entities, or type of data. We integrate the open-source Scorpio Broker (FIWARE 2024b) as the DT data broker. The *service mashup* interprets high level DT application language and generates detailed execution plan of the application. A capability of the service mashup is to discover the available DT entities and prescribe the correct number of processing function instances that covers all entities. Further, the function instances are connected to each other based on the workflows of processing pipelines. The service mashup also issues commands to execute the functions into processing nodes. The processing functions are containerized functions which get the data either from the Digital Twin data broker or from other processing functions. The service mashup establishes the correct input/output data flows of the processing functions. Processing functions include integrating data, data enrichment/augmentation, and ML model training. We utilize the serverless computing of FogFlow (FIWARE 2024a; Cheng et al. 2019) to realize the service mashup.

The user can instruct the system using an intent-based, high-level language. This language triggers operations such as dataset selection, data integration, enrichment, augmentation, and ML training and inference routines. The system interprets these commands using contextual information from the Digital Twin, including data availability, geographic scope, and targeted entities. Using the API, the user can programmatically select original data to be integrated, feature generation routines, dataset sampling methodology, and ML parameters configurations. Lastly, graphical user interfaces visualize the results of the DT application using the Scorpio Broker geodata export function as GeoJSON.

Fig. 3 shows four classes of DT functions automatically triggered (through the service mashup interpretation) by a specific type of entity (available at the DT data broker): **Sample Generator:** NGSI-LD *SampleSet* entities trigger the generation of SamplePoints according to the SampleSet attributes (sampling method, number of points, targeted area). **Enricher:** NGSI-LD *SamplePoint* entities trigger the enrichment function to compute the features necessary for the ML training and inference process. **ML-trainer:** NGSI-LD *RequestForModelTraining* entities trigger the service pipeline to train the ML models (preprocessing, training, testing, evaluating) and storing them. This operator also creates an entity describing the ML model including hyper-parameters and performance indicators. **AssessRisk:** NGSI-LD SampleSet entities with the 'risk_assessment' attribute triggers an instance of the appropriate ML model. The result is a new dataset of points containing the landmine risk at their location.

4 Digital Twin Data Processing

4.1 Data integration and enrichment

The Cambodia Mine Action and Victim Assistance Authority (CMAA) conducted a ground truth survey for landmine and ERW between 2009 and 2014. The UN Office for the Coordination of Humanitarian Affairs (OCHA) (Open Development Cambodia 2020) published this survey data. This dataset contains geospatial information on the contamination of 16,224 locations that are considered as "positive" samples in the ML training. Due to the nature of the use case, no negative data are available; however our DT system offers the augmentation of data by generating "negative" samples through hard negative data mining. Our system allows registering and applying several algorithms as docker containers for the data augmentation.

A key aspect for earth intelligence (Sun et al. 2022) is the *context enrichment*, which provides additional features to the positive and negative samples. We select geographic, topographic, and social data to enable the ML models infer military strategies and situational patterns. **Geospatial information:** OpenStreetMap (OSM) provides a wide range of data: natural features (e.g., water bodies), man-made infrastructure (e.g., roads, buildings), socio-economic points of interests such as education, health and financial facilities, as well as administrative boundaries (national borders, province and districts) and localities. The thematic features relevant to the ML are extracted and transformed in 19 NGSI-LD entity types as listed in Table 1. Once the data is indexed in the DT data broker, the context enrichment processing function calculates the necessary features for each sample. **Topographic data:** We use two raster datasets from the Shuttle Radar Topography Mission (SRTM) (Laboratory 2013) for the elevation and slope. Both datasets have a spatial resolution of ∼30m. **Population density:** Another relevant socio-economic factor is integrated from Gridded Population of the World project (CIESIN 2018). The raster resolution is 30 arc-seconds or ∼912m in Cambodia. All geo-data in this system is stored in WGS84. However, when distances and/or surfaces are calculated, the system will project the data into UTM zone 48N ('EPSG:32648') for Cambodia.

Table 1: Geospatial entity types on the data broker.

Entity Type	Count	Area (km^2)/ Length (km)	Country Area Coverage (%)
Airport Facility	14	38 km^2	0.02
Building	410,739	63 km^2	0.03
City	28	-	-
Coastline	124	1,294 km	-
Commune	25	1,519 km^2	0.75
Commune Border	62	714 km	-
Country	1	203,375 km^2	100.00
District	74	49,890 km^2	24.53
District Border	96	1,296 km	-
ERW	16,224	<points>	-
Education Facility	2,332	27 km^2	0.01
Financial Facility	1,178	-	-
Health Facility	1,381	1 km^2	0.00
Hamlet	619	-	-
Military	149	82 km^2	0.04
Residential Landuse	2,476.	870 km^2	0.43
Road	193,229	122,365 km	-
Town	192	17 km^2	0.01
Village	799	5 km^2	0.00
Water	21,101	5,487 km^2 32,763 km	2.70
Wetland	660	2,966 km	-

4.2 Data augmentation

Our DT system enables vast data augmentation for many ML trainings in an easily-reproducible manner. For the data augmentation, we apply three different techniques of data sample generation, to generate any number of negative samples based on the needs. For the experimental study, we generate same number of negative points (16,224), such that any random prediction would produce 50% accuracy as a starting benchmark for the ML models.

The first augmentation strategy is "uniform random" data point generation for negative samples, representing areas without marked landmines. The data points are distributed over the whole country. Uniform random negatives help models to learn all different patterns of geographies and contexts. On the other hand, it does not provide a high-resolution, such that the positive and negative samples may be very far apart. To provide a high-resolution dataset for later ML training, we use a second sampling strategy based on the "hard negative data mining" technique. Hard negative mining is recently applied in areas such as computer vision (Jiang et al. 2024; Robinson et al. 2020) and information retrieval (Formal et al. 2022), in the past called as bootstrapping (Sung 1996). Hard negatives are based on "hard" distance given to the model geometrically, such that a point is selected over the circular area over all positive samples with a radius equal to the given distance. Examples to such circles are shown in 4-top around the positive samples. The training based on hard negative data sampling forces the ML models to learn how to distinguish patterns of the areas which are close to each other. As the third strategy ("mixed sampling"), DT utilizes both uni-

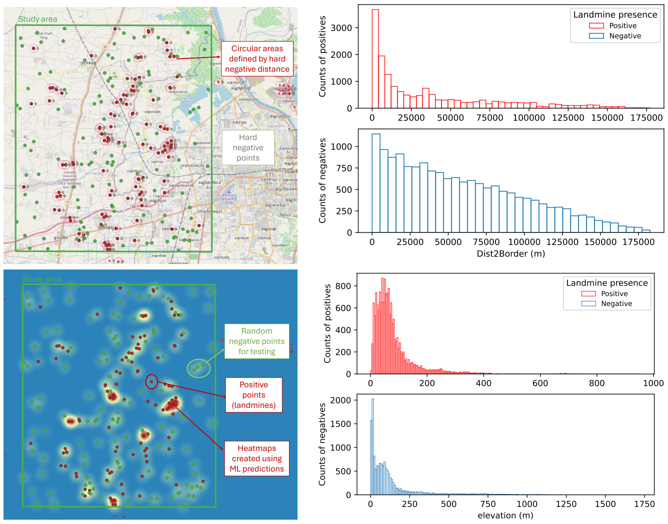

Figure 4: Left: An example study area to visualize training data augmentation and inference. Right: Distribution of positive and negative samples based on their distance to Cambodian borders (top two figures) and geographic elevation (bottom figures).

form random sampling and hard negative sampling on a given proportion parameter. We set the parameter as 0.5, such that half of the negatives (8,112) are generated uniformly and rest are based on the hard negatives. The intuition is to present the ML models points with different characteristics, augmenting with both close and far-away points from landmines.

Fig. 4 shows an example visualization of the application for a selected study area (top) using hard-negative data creation for training. The test data points (green points) follow uniform random distribution over the study area and they are utilized only on the inference phase. On the bottom, the positives (red points) and negatives (green) are predicted by the inference service and heatmaps are created based on the ML predictions.

Initial analysis. We analyze the difference of the positive and negative data samples based on their feature values. This analyses run over all available data features and helps developers understand data characteristics for any given dataset. As an example, Fig. 4 (top-right) is generated from the uniform random dataset. This figure shows the distances of positive and negative samples to the Cambodia country borders. The positives values are skewed towards shorter distances (e.g., distances shorter than 5km), whereas the negatives in this dataset are geometrically distributed. Similarly, Fig. 4 (bottom-right) shows the positive and negative samples based on their geographical feature of elevation from the sea level. Although in the case of Cambodia this feature is initially expected to be agnostic of the landmine presence, the data for positives and negatives follow different distributions, such that the negatives are skewed more towards lower elevation levels. In addition to these, each data feature are compared against each other based on their distributions of the positive (landmine) areas. The generated insights are stored as part of the DT and visualized statistics figures and as heatmaps on the QGIS platform.

Table 2: ML training parameters and specifications.

ML Parameters	Specifications
ML Models	XGB, LogR, SVC, RF, FNN
Number of geo-data points	32448
Sampling techniques	Un. random, hard neg., mixed
Hard negative dist.	{1, 2, 5, 10, 15, 20} (km)
Number of datasets	26
Number of trained ML models	130 (5 per dataset)
Proportion hard neg.	0.5 (for mixed sampling)
Number of features	22 non-categorical, 6 categorical
ML/Data aug. random seed	4
Train/test/validate (%)	80%/10%/10%
Number of epochs (NN)	250
Batch size (NN)	256
Dropout (NN)	0.05
Learning rate (NN)	0.0005
Evaluation metrics	F1, accuracy, recall, precision, training duration

4.3 End model training

DT generates two types of datasets for end model training: the first includes only features that are globally applicable (e.g., geographic features such as elevation); the second includes the local (categorical) features, such as names of provinces and districts in Cambodia. Thus, the latter datasets are aimed for specialized models only applicable in Cambodia. The popular one-hot encoding algorithm is applied to the categorical features. Although there are only 6 categorical features (Province, District, City, Town, Village, and Hamlet), the one-hot-encoding produces more than a thousand new features. Thus, the encoding vastly increases the dimensionality of the data, from 22 to 1221 dimensions (including the output labels; 1 for positive and 0 for negatives). Although the number of features seems significantly high, this enables to test the performances of ML models over high dimensionality. In addition, each dataset goes through preprocessing steps for the missing feature values and through scaling/normalization, shuffling, and splitting. We include validation results (over the 10% of the dataset).

The 5 different ML models, namely, XGBoost (XGB), Logistic Regression (LogR), Support Vector Classifier (SVC), RandomForest (RF), and Feedforward Neural Network (FNN), are chosen for training and the experimental study. These models are applied to the 26 datasets, resulting in 130 trained models. Each model parameters as well as outputs (e.g., performance results) are stored in the DT, and the DT can choose and deploy the models. All parameters of the ML training, including specifications of the data and ML architectures, are listed in Table 2.

5 Experimental Study

In our experiments, we instantiate a DT to support CMAC operations. DT allows us to generate training and explore practicability of various ML models.

5.1 System evaluation

We conduct the experiments on a GPU server with 4x NVIDIA RTX A5000 (24GB VRAM), 512 GB of DDR4-3200 MHz, 2x 1.6TB NVMe SSD using ZFS on a Ubuntu 22.04 LTS operating system. Further, the software setup is as follow: Python 3.12.5, pyTorch 2.4.1, scikit-learn 1.5.2, CUDA 12.2, Docker 27.1.2, FogFlow 3.2.8, Scorpio 5.0.3, PostGIS 3.4.2.

The system evaluation focuses on the generation of the datasets and the enrichment process. FogFlow instantiates a new processing function for each trigger, thus twinning each sample point. However, this would create tens of thousands of Docker instances. In one experiment, the Zettabyte File System snapshot mechanism overloaded the system at kernel level. To remedy this, we design the enrichment function to run as a singleton. The advantage is to support a country-wide DT with limited resources.

FogFlow's publish-subscribe mechanism takes about 4 min to supply all the newly generated SamplePoints to the geospatial enrichment function. This duration is negligible compared to ~5h required to enrich the 32k sample points. The singleton enrichment instance is configured to process as many as 60 points simultaneously. A first type of enrichment is through value extraction from raster files that result in three features (elevation, slope, and population density). We use a REST enrichment service to extract the data at given coordinates. Since some raster files are large (>1GB), we cache them at the service start. The execution time of raster lookup requests is very fast: 40ms (cache) vs 1.67s (no-cache). This approach is relatively hungry in memory (2.5GB used) but saves ~3sec by sample, totaling to 30h of processing time on a single core for a full SampleSet. Other features are based on a nearest neighbour calculation. The implementation of our enrichment processing function takes advantage of the GIS extension (PostGIS) of the Scorpio data broker. Leveraging PostGIS' ST_DISTANCE function and casting the geometry to geography, a single SQL query can find the nearest neighbour and calculate the distance to the sample point being processed. The overall enrichment time is variable, as shown in Fig. 5-top,

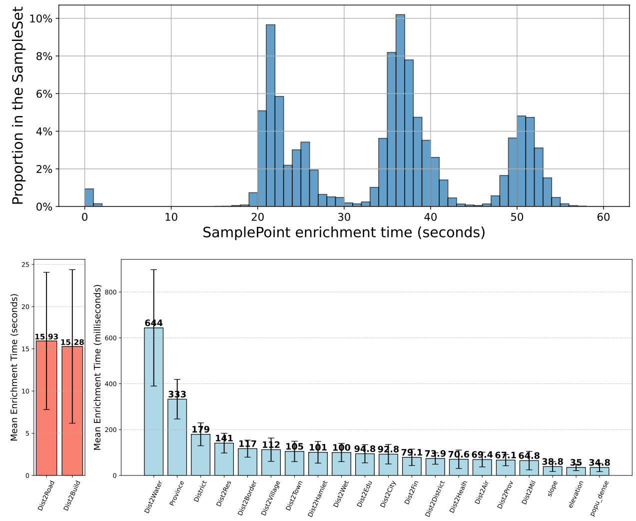

Figure 5: Top: Distribution of enrichment time for a dataset of 32,448 points. Bottom: Average enrichment time per data feature.

the vast majority spanning between 20 and 55 sec. We observe in Fig. 5-bottom that the Dist2Build and Dist2Road take considerably more time than other features. This is due to the distance computations on higher numbers of buildings and roads.

5.2 ML evaluation

At the end of each ML model training, the analytics results of accuracy, F1-score, recall, precision over the datasets, as well as the training duration are stored in the DT along with the corresponding trained ML model, so that for each model that is ready-to-deploy, DT can check those accuracy metrics. Furthermore, confusion matrices (1 confusion matrix per training, in total 130 confusion matrices) are created as separate plots for the developers. There are two types of performance analysis for the 5 metrics. The first analysis compares ML model performances by including or excluding categorical features. This type of analysis resulted in 60 figures. The second analysis compares model performances based on different sampling strategies and hard negative distances (different datasets). This analysis resulted in only 10 figures, since each figure contains results from multiple ML models and datasets. Due to the space limitation, we highlight figures that showcase the effects of certain parameters over the model performances, where we observe similar patterns across all figures.

Fig. 6-left shows the performances of ML models based on their F1-score accuracy metric in the cases of global models without categorical features compared to the Cambodia-specific models with cat-

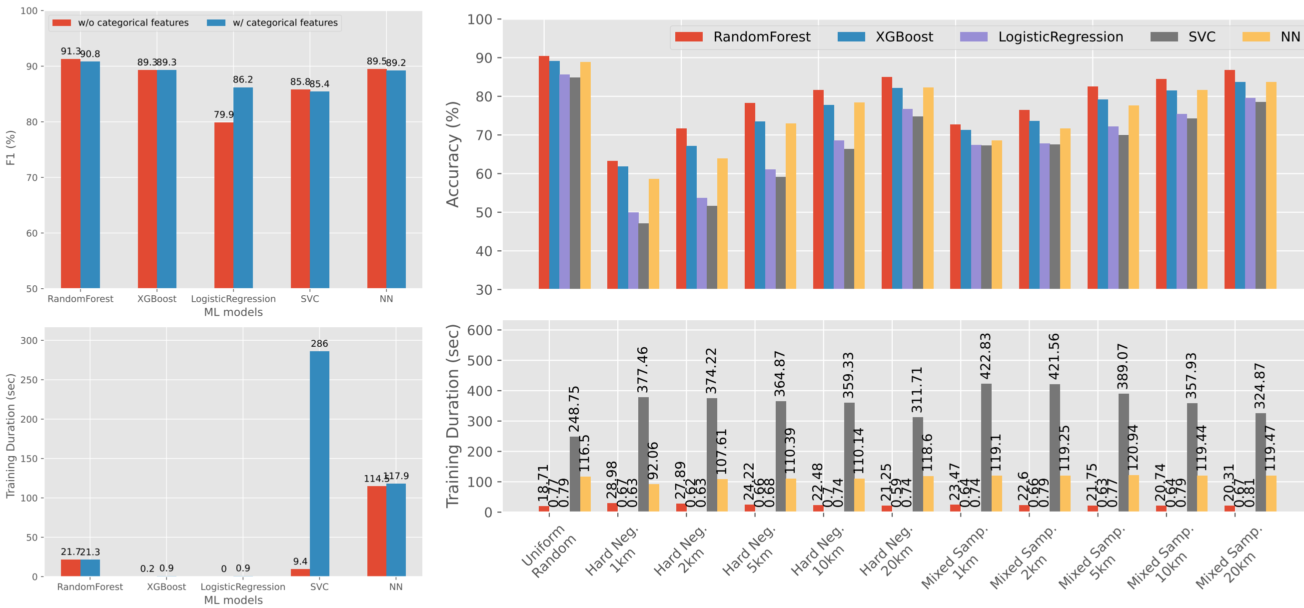

Figure 6: Left: F1 scores and training durations of the ML models on the uniform random data w/ and w/o categorical features. Right: Classification performance through data augmentation strategies: Uniform random, hard negative and mixed sampling with different hard distances.

egorical features. First, the top model (RF) performs 91.3%, showing that the model can successfully distinguish landmine areas from the non-mine areas. A very similar performance of 89.5% is obtained from the neural networks. The performances of the models are not significantly affected by the existence of categorical features, except the LogR model with a $\sim$6 point gain in F1-score thanks to the categorical features. The bottom-left figure shows the training duration of the same models. The training duration differ significantly, such that the XGB and LogR provide the fastest training both of them with less than only 1 sec. On the other hand, RF takes $\sim$22 sec, and NN $\sim$115 sec. These four models perform the training in a similar duration in case of inclusion of categorical features. On the other hand, SVC performance on training duration highly degrades with the categorical feature with a 30-fold increase in the training duration. We consider the high number of increase in the dimensions ($\sim$60-fold, from 21 to 1220) caused this degradation due to the implementation of the kernel functions based on similarities. For tree-based models such as NN or RF, addition of more features do not make a real difference. For instance, NN model takes about 3 sec longer time due to the number of input nodes increasing to 1220.

Fig. 6-right includes the experimental accuracy results across the datasets created through different data augmentation strategies. The RF models provide the best accuracy in every dataset, showing the success of the model on dealing with 1220 data features. The best performance is followed by XGB as another ensemble learning algorithm that combines multiple decision-trees. The bottom-right figure compares the training durations of these models, showing SVC having a significantly longer training durations due to the high-dimensionality of the data with categorical features. XGB provides a good trade-off with the second highest performance and very fast training times in the range of $[0.6, 0.7]$sec.

This section highlights the results from the ML experiments, whereas there are many other data-centric insights that could be generated using the country-wide DT system. The enabling of training of many ML models by the DT allows the developers run vast number of experiments and create additional insights. Finally, the inference phase where the models are deployed and exposed as a service by the DT, enables testing these models against any given arbitrary study area. The DT system and application are deployed in the CMAC premises using the propriety datasets.

6 Conclusion

This paper introduced a country-wide DT system that improves landmine risk assessment for demining efforts. By streamlining data processes and using high-accuracy ML models, the system supports the mine action project for prioritizing and planning activities in Cambodia.

References

Almasan, P.; Ferriol-Galmés, M.; Paillisse, J.; Suárez-Varela, J.; Perino, D.; López, D.; Perales, A. A. P.; Harvey, P.; Ciavaglia, L.; Wong, L.; et al. 2022. Network digital twin: Context, enabling technologies, and opportunities. *IEEE Communications Magazine*, 60(11): 22–27.

Anonymous. 2024. [Alternative title for anonymity] Machine learning for Geospatial AI Predictions. *Anonymous journal*.

Avishahar-Zeira, A.; and Lorenz, D. H. 2023. Could No-Code Be Code? Toward a No-Code Programming Language for Citizen Developers. In *Proceedings of the ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, 103–119.

Cheng, B.; Fuerst, J.; Solmaz, G.; and Sanada, T. 2019. Fog function: Serverless fog computing for data intensive iot services. In *2019 IEEE International Conference on Services Computing (SCC)*, 28–35. IEEE.

Cheng, B.; Solmaz, G.; Cirillo, F.; Kovacs, E.; Terasawa, K.; and Kitazawa, A. 2017. FogFlow: Easy programming of IoT services over cloud and edges for smart cities. *IEEE Internet of Things journal*, 5(2): 696–707.

CIESIN, C. U. 2018. Gridded Population of the World, Version 4 (GPWv4): Population Count. Accessed: 2024-06-09.

Dalibor, M.; Heithoff, M.; Michael, J.; Netz, L.; Pfeiffer, J.; Rumpe, B.; Varga, S.; and Wortmann, A. 2022. Generating customized low-code development platforms for digital twins. *Journal of Computer Languages*, 70: 101117.

Dulce Rubio, M.; Zeng, S.; Wang, Q.; Alvarado, D.; Moreno Rivera, F.; Heidari, H.; and Fang, F. 2024. RELand: Risk Estimation of Landmines via Interpretable Invariant Risk Minimization. *ACM Journal on Computing and Sustainable Societies*, 2(2): 1–29.

Dzeparoska, K.; Lin, J.; Tizghadam, A.; and Leon-Garcia, A. 2023. LLM-based policy generation for intent-based management of applications. In *International Conference on Network and Service Management (CNSM)*, 1–7. IEEE.

ETSI CIM. 2024. Context Information Management (CIM); NGSI-LD API. Technical Report ETSI GS CIM 009 V1.8.1, European Telecommunications Standards Institute (ETSI).

FIWARE. 2024a. FogFlow. https://fogflow.readthedocs.io/en/latest/.

FIWARE. 2024b. Scorpio Broker. https://scorpio.readthedocs.io/en/latest/.

Formal, T.; Lassance, C.; Piwowarski, B.; and Clinchant, S. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2353–2359.

Jafari, M.; Kavousi-Fard, A.; Chen, T.; and Karimi, M. 2023. A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future. *IEEE Access*, 11: 17471–17484.

Jamil, A.; Padubidri, C.; Karatsiolis, S.; Kalita, I.; Guley, A.; and Kamilaris, A. 2023. GAEA: A Country-Scale Geospatial Environmental Modelling Tool: Towards a Digital Twin for Real Estate. In *Environmental Informatics*, 177–199. Springer.

Jiang, R.; Nguyen, T.; Ishwar, P.; and Aeron, S. 2024. Supervised contrastive learning with hard negative samples. In *International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Laboratory, N. J. P. 2013. NASA Shuttle Radar Topography Mission Global 1 arc second [Data set]. Accessed: 2023-12-09.

Leivadeas, A.; and Falkner, M. 2022. A survey on intent-based networking. *IEEE Communications Surveys & Tutorials*, 25(1): 625–655.

Logothetis, I.; Mari, I.; and Vidakis, N. 2023. Towards a digital twin implementation of eastern Crete: an educational approach. In *International Conference on Extended Reality*, 255–268. Springer.

Mahmud, S. T.; Ma, W.; Thomsen, T.; Chen, C.; Rex, R.; Lai, A.; Sohn, L. L.; and Randles, A. 2025. Microfluidic Digital Twin for Enhanced Single-Cell Analysis. In *International Conference on Computational Science*, 283–297. Springer.

Mandal, S. 2024. A privacy preserving federated learning (PPFL) based cognitive digital twin (CDT) framework for smart cities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 23399–23400.

Mandal, S.; and O'Connor, N. E. 2024. LLMasMMKG: LLM Assisted Synthetic Multi-Modal Knowledge Graph Creation For Smart City Cognitive Digital Twins. In *Proceedings of the AAAI Symposium Series*, volume 4, 210–221.

Michael, J.; and Wortmann, A. 2021. Towards Development Platforms for Digital Twins: A Model-Driven Low-Code Approach. In Dolgui, A.; Bernard, A.; Lemoine, D.; von Cieminski, G.; and Romero, D., eds., *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, 333–341. Springer International Publishing.

Open Development Cambodia. 2020. Baseline survey on mine/ERW (2009-2014). Accessed: June, 2024.

Pinho, D.; Aguiar, A.; and Amaral, V. 2023. What about the usability in low-code platforms? A systematic literature review. *Journal of Computer Languages*, 74: 101185.

Rafique, W.; Zheng, D.; Barras, J.; Joglekar, S.; and Kosmas, P. 2019. Predictive analysis of landmine risk. *IEEE Access*, 7: 107259–107269.

Redeker, M.; Weskamp, J. N.; Rössl, B.; and Pethig, F. 2022. A digital twin platform for Industrie 4.0. In *Data Spaces: Design, Deployment and Future Directions*, 173–200. Springer International Publishing Cham.

Reuters. 2025. A Path to Safety: Japan as a leader in removing landmines and supporting victims. https://www.reuters.com/plus/a-path-to-safety-japan-as-a-leader-in-removing-landmines-and-supporting-victims.

Robinson, J.; Chuang, C.-Y.; Sra, S.; and Jegelka, S. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*.

Sarkar, S.; Naug, A.; Guillen, A.; Luna, R.; Gundecha, V.; Babu, A. R.; and Mousavi, S. 2024. Sustainability of data center digital twins with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 23832–23834.

Schultz, C.; Alegría, A. C.; Cornelis, J.; and Sahli, H. 2016. Comparison of spatial and aspatial logistic regression models for landmine risk mapping. *Applied Geography*, 66: 52–63.

Sottet, J.-S.; and Pruski, C. 2023. Data and Model Harmonization Research Challenges in a Nation Wide Digital Twin. *Systems*, 11(2): 99.

Sun, T.; He, X.; and Li, Z. 2023. Digital twin in healthcare: Recent updates and challenges. *Digital Health*, 9: 20552076221149651.

Sun, Z.; Sandoval, L.; Crystal-Ornelas, R.; Mousavi, S. M.; Wang, J.; Lin, C.; Cristea, N.; Tong, D.; Carande, W. H.; Ma, X.; et al. 2022. A review of earth artificial intelligence. *Computers & Geosciences*, 159: 105034.

Sung, K. K. 1996. Learning and example selection for object and pattern detection.

Talasila, P.; Gomes, C.; Mikkelsen, P. H.; Arboleda, S. G.; Kamburjan, E.; and Larsen, P. G. 2023. Digital twin as a service (DTaaS): a platform for digital twin developers and users. In *IEEE Smart World Congress (SWC)*, 1–8. IEEE.

Zhang, J.; Zhu, J.; Guo, Z.; Wu, J.; Guo, Y.; Lai, J.; and Li, W. 2025. More intelligent knowledge graph: A large language model-driven method for knowledge representation in geospatial digital twins. *International Journal of Applied Earth Observation and Geoinformation*, 139: 104527.

Reproducibility Checklist

This paper:

Includes a conceptual outline and/or pseudocode description of AI methods introduced: **yes**

Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results: **yes**

Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper: **yes**

Does this paper make theoretical contributions? **no**

Does this paper rely on one or more datasets? **yes**

If yes, please complete the list below.

A motivation is given for why the experiments are conducted on the selected datasets **yes**

All novel datasets introduced in this paper are included in a data appendix. **NA**

All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **NA**

All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. **yes**

All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. **yes**

All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisfying. **NA**

Does this paper include computational experiments? **yes**

If yes, please complete the list below.

This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. **yes**

Any code required for pre-processing data is included in the appendix. **no, since the code is part of a bigger system and will not work as standalone.**

All source code required for conducting and analyzing the experiments is included in a code appendix. **yes**

All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **yes**

All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from **NA**

If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. **yes**

This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. **yes**

This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. **yes**

This paper states the number of algorithm runs used to compute each reported result. **yes**

Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. **yes**

The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). **yes**

This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. **yes**