A Synthetic Data Generation System based on the Variational-Autoencoder Technique and the Linked Data Paradigm

Ricardo Dos Santos[1], Jose Aguilar[1,2,3]

[1]CEMISID, Universidad de Los Andes, Mérida, Venezuela
[2]GIDITIC, Universidad EAFIT, Medellin, Colombia
[3]IMDEA Networks Institute, Madrid, Spain

Abstract: Currently, the generation of synthetic data has become very fashionable, either due to the need to create data in certain specific contexts or to study unknown scenarios among other reasons. Additionally, synthetic data is a critical component in training machine learning models in the presence of little data. This work proposes a Synthetic Data Generation System (SDGS) architecture to allow synthetic data generation to be fully automated. SDGS is based on the Variational AutoEncoders (VAE) learning technique, and has three main capabilities. The first is related to the ability to extract data samples from multiple sources using the Linked Data (LD) paradigm. The second is linked to the ability to merge data sets to increase the amount of information that can be provided to the VAE-based synthetic data generator. The last one is related to having a Feature Engineering layer to create new features by generating or extracting information from the dataset and then selecting the features that provide the best information for the VAE model. A case study is described in detail to show the new functionalities of the SDGS, such as dataset extraction from different sources using LD, dataset merging using pivots, and the application of different feature engineering methods. Finally, two metrics are used to evaluate the quality of the generated datasets in different case studies. The first one is the accuracy to analyze the performance of the models generated with the new SDGS functionalities, obtaining results above 90%. The second one is the two-Sample Hotelling's T-Squared Test to determine the quality of the synthetic data generated by the system, obtaining synthetic datasets very similar to the original datasets.

Keywords: Synthetic Data Generator, Linked Data, Variational Autoencoders,

I. INTRODUCCIÓN

A great current technological challenge is being able to take advantage of multiple datasets to generate synthetic data in specific contexts. This requires identification processes of said data sources, for which the LD paradigm can be used, as well as subsequently merging these multiple datasets into a single sample dataset for each given context [22], [23], [24]. Now, this set of data samples obtained from multiple sources requires a process of analysis of its characteristics (feature engineering) that allows optimizing the quality of the data, before going to a synthetic data generation process [19], [25], [26].

In this work, an architecture is proposed, called SDGS, which allows the generation of synthetic data considering those three aspects mentioned above, which uses the VAE learning technique to generate synthetic data. Specifically, SDGS uses the LD paradigm to identify data sources (used as an identifier of datasets for a specific context) [1]. Subsequently, it performs the fusion of the different identified datasets and a feature engineering process to identify the relevant variables. In this way, it builds a dataset for a given context. Finally, SDGS uses a VAE-based model to generate new data similar to the newly constructed dataset. In this way, it creates larger data sets from samples of small data sets [22], [23].

A. Relate works

Some similar research on dataset extraction and fusion with LD are as follows: Avazpour et al. [2] described an architecture that incorporates complex data aggregation from multiple sources, mapping and data transformation. The architecture is made up of a set of multiple components: i. The dataset components collect all the datasets transformed into CSV that come from different sources (CSV, RDF, API, among others); ii. The aggregator components allow extracting partial information from each specific CSV; III. The mapping components allow the imported data to be assigned to generate the data model of the architecture. In the work [3], the authors defined a keyword search system on federated RDF datasets. The process begins when the Mediator component receives the set of keywords specified by the user. The Mediator component uses the storage component to find the data and metadata that match the keywords. The Federated Schema Component is then used to find outer joins between the subqueries computed by each dataset. Finally, the Mediator executes the federated SPARQL query and returns the dataset with the composition of the data requested by the user from the different sources of the federated RDF datasets.

On the other hand, Rao et al. [4] presented a method for fusing linked open data from multiple sources for querying relationships between drugs and genetic disorders. Generally, the information about genes, drugs and disorders was stored in different places and in different formats such as RDF/XML, SQL and relational, among others. In this method, biomedical datasets are converted into RDF triples, normalizing the vocabulary and data URIs, and then stored as merged data. After merging, the system can be queried with SPARQL to understand the relationships between multiple entities from different datasets and extract datasets for a more specific context. Similarly, in [5], Chen proposed an LD fusion method based on the calculation of similarity and k-nearest neighbor (KNN), allowing to solve problems of entity conflicts between data sources. The LD similarity calculation effectively integrates URI nodes and blank nodes into linked data. On the other hand, the fusion of literal type nodes based on the KNN classification allows automating the fusion independently of the data sources. The KNN classifier generates a model that learns from common assignment strategies for resolving conflicts between literal nodes.

In the context of VAE, there is some research that applies feature engineering. For example, Nishimaki et al. [6] described the Localized Variational-Autoencoder (Loc-VAE) that provides neuroanatomically interpretable low-dimensional representation from 3D brain MR images. In this research, they take advantage of VAE models for feature extraction (using the latent vector Z of an image where the vector represents the extracted features). Then, they generate perturbations to each element of the latent vector and pass through the decoder, obtaining new images with the result of the perturbation. On the other hand, a feature selection method, called AVAE (Attention of Variational Autoencoder) is proposed by Van Dao et al. [7], which emphasizes the importance of the attention mechanism in the selection and evaluation of weights in the latent space. AVAE consists of CBAM (Convolutional Block Attention Module) coding layers, and can learn what and where to emphasize or suppress. CBAM is a lightweight CNN with only two convolutional layers: Channel Attention Module (CAM) and Spatial Attention Module (SAM).

In [8], Hadipour et al. developed a molecular embedding learning approach that combines PCA (Principal Component Analysis) and VAE to integrate global and local molecular features. The approach begins by collecting the molecule data, extracting the global (molecular descriptors) and local (atomic and bond) features for each molecule. The PCA method is used to reduce the atomic feature matrix and the bond feature matrix to a PCA-based feature vector, respectively. Then, it concatenates the global and local features and it filters out the columns with zero variance. Finally, VAE is used to incorporate the global

chemical properties and the local atom and bond features. Also, Akkem et al. [21] proposed the use of VAE and Generative Adversarial Networks (GAN) to generate synthetic data for crop recommendation. This research focuses on exploring the effectiveness of VAE and GAN in producing high-quality synthetic data, facilitating improved training and evaluation of recommender systems. In addition, they performed extensive qualitative analysis on the reliability of synthetic data in various experiments, including visual comparisons such as heatmaps, scatter plots, cumulative sum per feature plots, and distribution per feature plots. The work of Marco et al. [27] explored the use of conditional variational autoencoder (CVAE) and inverse normalization transformation for data augmentation and synthetic data generation in software engineering. Eleven datasets from sources like PROMISE and ISBSG were used. The CVAE-INT model created synthetic data, showing significant results with a mean p-value above 0.90 in the Mann-Whitney test. The model achieved lower mean absolute error and root mean squared error across various datasets.

Panfilo et al. [28] introduced a data synthesis technique for both supervised and unsupervised learning on single-table and relational datasets. Utilizing generative deep learning models, the technique includes three variants: standard VAE, β-VAEs, and Introspective VAEs. The effectiveness of these variants is experimentally evaluated to determine how well they meet the quality requirements for generated data. Various performance indexes are used to capture different aspects of data quality, demonstrating the applicability of these models to relevant business cases. Kuo et al. [29] proposed enhancing the classic GAN framework with a VAE and an external memory mechanism to generate synthetic datasets that accurately reflect imbalanced class distributions in clinical variables. The method was tested on data related to antiretroviral therapy for HIV. Results show the method effectively prevents mode collapse, ensures low patient disclosure risk (0.095%), and maintains high utility for machine learning applications in healthcare. Eigenschink et al. [30] introduced a data-driven evaluation framework for generative models that produce synthetic sequential data. The framework assesses models based on five criteria: representativeness, novelty, realism, diversity, and coherence, independent of the models' internal structures. These criteria address various domain-specific requirements, allowing users to evaluate synthetic data quality across models. A review of generative models for sequential data shows that realism and coherence are crucial for natural language, speech, and audio processing, while novelty and representativeness are vital for healthcare and mobility data. Representativeness is typically measured using statistical metrics, realism through human judgment, and novelty with privacy tests. Finally, an initial SDGS has been proposed in [1], which only carries out an extraction of data from only one source to generate new data. A summary of the literature review with the articles closest to ours is presented in Table 1, showing the advantages and limitations with respect to our approach.

TABLE 1: LITERATURE REVIEW RELATED TO OUR RESEARCH.

Work	Advantages	Limitations
[1]	Propose a synthetic data generator that uses the	Data extraction is performed from a single source and does not

	advantages of the LD paradigm for automatic detection and extraction of data samples for smart grids, and the learning capability of VAE algorithms.	perform a feature engineering process to improve the generation of synthetic data.
[2]	It describes an architecture that incorporates complex data aggregation from multiple sources.	All sources must be transformed to CSV before being used by the architecture, and it does not have the ability to extract datasets scattered on the web as Linked Open Data.
[3]	It presents a system for searching federated RDF datasets using SPARQL queries.	It needs to manually create UNION clauses to combine the results of queries
[4]	It presents a method of merging linked open data from multiple sources converting them into RDF triples.	It is quite difficult to build datasets with this granularity and still maintain the relationship between the features of different datasets.
[5]	It proposes a fusion method based on similarity calculation and KNN, which allows solving problems of entity conflicts between data sources.	The difficulty in queries remains due to the granularity of the information.
[6]	VAE models are used to extract features from brain images, observing the effect generated by perturbations to the latent Z vector.	The data generated by the VAE model are used to find the explainability of the extracted features.
[7]	It proposes a feature selection method that helps to acquire important features in the latent space of the VAE model.	It emphasizes the importance of the weight selection and evaluation mechanism for the VAE.
[8]	VAE is used to incorporate the global chemical properties and the local atom and bond features.	VAE is not used to generate synthetic data but as a feature fusion component.
[21]	It proposes the use of VAE and GAN to generate synthetic data for crop recommendation.	It's not defined methods that allow the datasets to be obtained automatically, no feature engineering has been applied to improve the original data.

The main difference between our approach with previous works is that we define an entire architecture for the generation of synthetic data, where aspects such as the acquisition, preparation and optimization of the dataset are covered to improve the generation of synthetic data.

B. Contributions

The main difference between the works described above with this research is that our work, before generating data, seeks to build a robust dataset with the extraction and fusion of all data from different sources. In our case, the datasets already constructed in the study area are identified using the LD paradigm. The selected datasets are then merged based on the relationships that exist between them. Subsequently, a Feature Engineering process is carried out to build a robust dataset before passing it on to the VAE, this allows us to offer new and more information from the data. Finally, our VAE is used to generate synthetic data from the created dataset. In general, the main contributions of this research are:

- It defines an SDGS architecture for the automatic generation of synthetic data.
- It specifies an automatic mechanism for dataset identification from multi-sources based on the LD paradigm.
- It specifies an automatic mechanism for the creation of a robust dataset based on multi-datasets.
- It defines a Feature Engineering module that merges, extracts and selects features.

This article is organized as follows: Section 2 shows the theoretical framework around SDGS; Section 3 describes the SDGS components for the multi-sources and multi-datasets management processes, and the feature engineering module; Section 4 describes a case study where the SDGS is tested; Section 5 presents the experiments with the SDGS in different case studies and a comparison with other architectures. Finally, Section 6 presents the conclusions of this research.

II. SDGS

The SDGS architecture generates synthetic data using the LD paradigm to identify and extract data from the Internet and the VAE technique to train a model with this sample so that this model can later be used to generate synthetic data. This architecture is composed of the following modules (see Fig. 1) [1]:

- **DataSet Acquisition (DSA):** The objective of this module is to find data samples through LD-based search mechanisms, taking advantage of Open Data Sources (ODS) or endpoints.
- **Data Preparation (DP):** The objective of this module is to optimize the data sample. It normalizes numerical attributes with high variance, and processes textual or numerical data representing a specific finite set of categories or classes.
- **Synthetic Data Generation (SDG):** The objective of this module is to generate synthetic data by training a VAE-based knowledge model, which automatically extracts and learns the features of the optimized data sample for a given context.

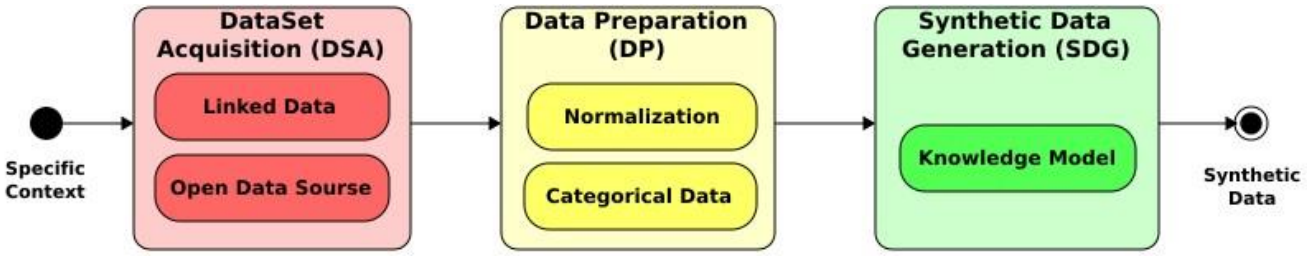

Fig. 1: SYNTHETIC DATA GENERATION ARCHITECTURE [1].

In this work, SDGS is extended with new components in order to implement its Synthetic Data Generation functionality.

III. EXTENSIONS TO THE SDGS

A. Architecture

This research extends the approach proposed in [1] (see Fig. 2). Firstly, by adding two processes to the DSA module, a first process focused on the use of multiple data sources and a second process focused on the fusion of multiple datasets. In addition, a new module, called Feature Engineering (FE), is added to analyze the characteristics of the sample datasets to be used by the SDG, allowing Feature Fusion, Feature Extraction and Feature Selection. Finally, the SDG is implemented using the VAE technique as the data generator.

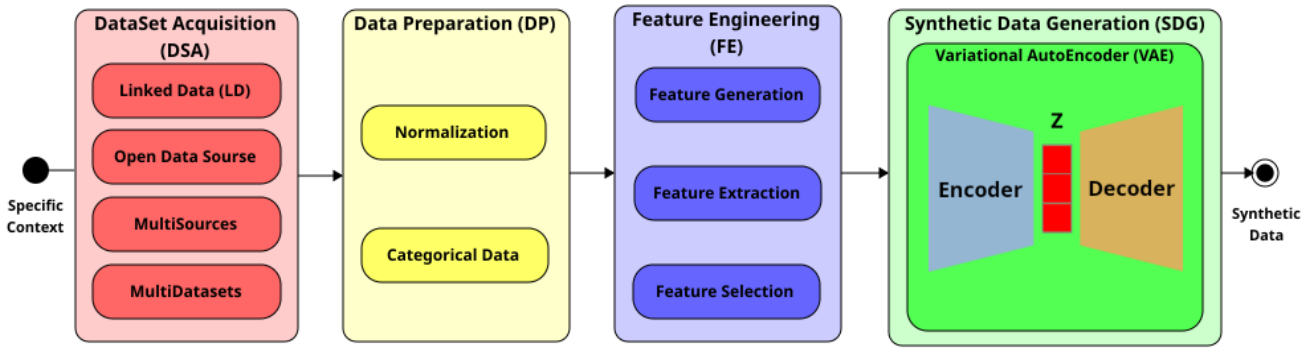

Fig. 2: EXPANSION OF SDGS

B. DSA Module

The objective of this module is to find data samples for a given context using the LD paradigm. Table 2 shows the macro-algorithm of the DSA module. This module begins by analyzing the context of the required data samples, obtaining the keywords to search the data samples (Step 1). Then, the Multi-sources process is invoked to search the datasets using the keywords obtained in the previous step (Step 2). Finally, if it is decided to merge the obtained datasets with another dataset from an associated context (Step 3), then the multiple dataset process is invoked (Step 3.1).

TABLE 2: MACRO-ALGORITHM OF THE DSA MODULE.

Input: Specific Context
Procedure: 1. The context of the required data is analyzed in order to obtain the search keywords. 2. Invoke the Multi-sources process to find a Data Sample using the search keywords 3. If merging the Data Sample: 3.1 Invoke the Multi-dataset process to merge the Data Sample with the dataset from the associated context.
Output: New Dataset

B.1. Multi-sources Process in DSA

The objective of this process is to find data samples from different dataset sources. Specifically, the data samples are obtained using search mechanisms based on LD, taking advantage of Open Data Sources (ODS), searching each dataset source registered in the SDGS and selecting sample datasets that best match the specific context required. These dataset sources are officially provided by countries/regions such as Europe (https://data.europa.eu), Spain (https://datos.gob.es/), Canada (https://open.canada.ca), United States (https://www.data.gov/), among others. The metadata published by these sources follows the CKAN standard (https://ckan.org/), which allows queries using the SPARQL language. Table 3 shows the macro-algorithm of the Multi-sources Process in DSA. This process begins by preparing the search queries for each dataset source (Step 1). Then, each query is executed in each dataset source, and the list of possible datasets is obtained (Step 2). Finally, the list is ordered according to the degree of coincidence with the required context (Step 3).

TABLE 3: MACRO-ALGORITHM OF MULTISOURCES PROCESS TO SEARCH SAMPLES OF DATA.

Input: Search Keywords

Procedure: 1. Prepare the search queries with the search keywords for each dataset source based on the LD paradigm. 2. The search is executed for each dataset source and added to the list of possible data samples. 3. The data samples that best fit the search are sorted and selected.
Output: Data Sample

B.2. Multi-datasets Process in DSA

The objective of this process is to build a data sample that combines information from different datasets. Specifically, having a dataset from the main context required, another dataset belonging to a context associated with the main context is searched. Then, it searches for relations between the features of the datasets; the matches are used as pivots for the merging of both datasets. Table 4 shows the macro-algorithm of the Multi-datasets process in DSA. This process begins by invoking the Multi-sources process to find a Data Sample from the Associated Context (Step 1). Step 2 searches for the similarities of both Data Samples; this similarity is based on the name and type of each feature of the Main Data Sample and the Associated Context Data Sample. Finally, it merges the Main Data Sample and the Associated Context Data Sample using the similarities as a pivot (Step 3), generating a Merged Data Sample.

TABLE 4: MACRO-ALGORITHM OF MULTIDATASETS PROCESS TO FUSION SAMPLES OF DATA.

Input: Main Data Sample, Associated Context Search Keywords
Procedure: 1. Invoke the Multi-sources process to find a Data Sample using Associated Context Search Keywords. 2. Search for possible similarities between the characteristics of the Main Data Sample and the Associated Context Data Sample. 3. Merge both Data Samples using the similarities as a pivot.
Output: Merged Data Sample

C. DP Module

The objective of this module is to transform the sample dataset into an optimal representation for the VAE model, knowing that this type of model works optimally with data ranging between [0 and 1] or [-1 and 1], either binary (digital) or continuous (analog) data. Some of the tasks this module does is convert textual or numerical data into a specific finite set of categories (categorical data), or normalize numerical data with high variance, among other things. Table 5 shows the DP macro-algorithm, which begins by analyzing the dataset to determine the processes that will be required for each column of the data sample (Step 1). For columns with numeric data with many different values, it proceeds to normalize them (Step 2). For columns with textual or numeric data that can be represented in categories, it proceeds to categorize them (Step 3).

TABLE 5: MACRO-ALGORITHM OF THE DP MODULE TO OPTIMIZE THE SAMPLE OF DATA.

Input: Sample Dataset

Procedure:
1. The sample of data is analyzed.
2. The attributes with numeric data and high variance are normalized.
3. The attributes with textual and numeric data with finite values are categorized.
Output: Preprocessed Data Sample

D. FE Module

The objective of this module is to analyze the characteristics of the sample dataset. It specifically analyses the Preprocessed Data Sample generated by the DP module, obtaining new information from the dataset features and selecting the features that offer more information for the SDG module. Table 6 shows the macro-algorithm of the FE module. This process begins with the analysis of the dataset (Step 1). Then, in step 1.1, information is aggregated by applying Feature Generation techniques such as Interaction Feature, Polynomial Feature, Trigonometry Feature, Create Clusters, or Combine Rare Levels. In Step 1.2, information is added by applying Feature Extraction techniques such as Media-Based Feature, Median-based Feature and Quartiles-Based Feature. Finally, in step 1.3, it selects the features that offer the most information, applying Feature Selection techniques such as Permutation Feature Importance, Remove Multicollinearity, Ignore Low Variance, and Genetic Algorithm. In the next section, we will explain in detail the Feature Generation, Feature Extraction and Feature Selection techniques used in the case study to evaluate the behavior of our SDGS.

TABLE 6: MACRO-ALGORITHM OF FE TO ENHANCE THE SAMPLE OF DATA.

Input: Preprocessed Data Sample
Procedure:
1. Analyses the Preprocessed Data Sample:
1.1. Add new information generated from its characteristics.
1.2. Add new information extracted from its characteristics.
1.3. Select the features that provide the most information.
Output: Enhanced Data Sample

E. SDG Module

The objective of this module is to generate the synthetic data from the data sample optimized in the previous module. In this process, a knowledge model is built and trained that automatically extracts and learns the characteristics of the sample of data using VAE. This knowledge model is then used to generate the synthetic data. Table 7 shows the SDG macro-algorithm, the process begins by configuring and building the knowledge model that will learn the latent characteristics in the sample of data (Step 1). Step 2 consists of training the knowledge model using VAE and the data sample. Finally, the synthetic dataset is generated using the previously created and trained knowledge model (Step 3).

TABLE 7: MACRO-ALGORITHM OF SDG FOR THE GENERATION OF SYNTHETIC DATA.

Input: Enhanced Data Sample
Procedure:
1. The knowledge model with the desired configuration is built.
2. The knowledge model representing the sample of data is trained.
3. The synthetic dataset is generated with the knowledge model.
Output: Synthetic Dataset

IV. DESCRIPTION OF SDGS IN A CASE STUDY

This section presents a case study, which shows the new capabilities of the architecture.

A. Experimental Context

The objective of this case study is to show a functional version of the modules of the proposed system for a smart grid context, allowing the automated extraction of a data sample and the optimization of the information provided by the dataset. In general, this architecture can be deployed in any context for automatic synthetic data generation. Specifically, the modules will be deployed to search for datasets in the context of Energy Management, with an associated context on Territorial Indicators that will allow us to show the fusion of datasets. Additionally, in order to show the ability to identify and extract data from multiple sources, it will use three sources of datasets:

- Canada Open Government (https://open.canada.ca): it offers more than 40,000 datasets with information from Canadian provinces, territories and municipalities on government activities such as health, science and technology, economy and industry, and education, among many others.
- European Open Data (https://data.europa.eu): it offers more than 1.5 million datasets about 33 European countries with information on energy, agriculture, transport, regions and cities, etc., and
- Open Data of the Government of Spain (http://datos.gob.es): it offers more than 66 thousand datasets with information on demography, urban planning and infrastructure, rural environment, and employment, among others.

On the other hand, in this case study, different techniques were chosen for each type of feature engineering that the system has, with the intention of showing how the FE module works. Each one of the selected techniques is described below:

Regarding the Feature Generation technique, the following technique was used:

- Trigonometric features generate new features by applying the trigonometric functions tangent, sine and cosine. This process generates three features for each numerical feature in the dataset.

Also, the next Feature Extraction techniques were used [11], [13]:

- Media-Based Feature, it extracts the mean of the feature and constructs a new feature with the distance of each value of the feature with respect to its mean. This process is repeated for each numerical feature in the dataset.
- Median-Based Feature, it extracts the median of the feature and constructs a new feature with the distance of each value of the feature related to its median. This process is repeated for each numerical feature in the dataset.
- Quartiles-Based Feature extracts the quartiles of the feature (25%, 50%, 75%, 100%), where the first quartile is all values that are less than 25% of the data for that feature, the second quartile is all values greater than 25% and less than 50% of the data, the third quartile is all values greater than 50% and less than 75%, the fourth quartile is all values greater than 75% and less than 100%. With this information, it builds a new feature that indicates the quartile of each value of that feature. This process is repeated for each numerical feature in the dataset.

Finally, the next Feature Selection technique was used [13], [14]:

- Genetic Algorithm, it selects the features that provide the most information using the fitness individual's chromosome. A chromosome of an individual is represented by all the features in the dataset and it is coded in binary, 1 if present and 0 if not present in that individual. To calculate the quality of an individual, a classification model is used as the fitness function. Particularly, to build

the classification model, the Logistic Regression technique was used. The fitness of an individual is obtained based on the quality of the results of the classifier using the dataset only with the features presented in this individual. This dataset is divided into two parts, the first part is used to train the classifier and the other part is used to measure its performance. In addition, in each generation, it creates a new population, which will serve to obtain the parents for the new individuals. At this point, it applies two genetic operations: crossovers and mutations. In crossover, it copies all the features from one parent up to the crossover point, and the rest of the features are copied from a second parent, the default probability for crossover is 60%. In mutation, it selects the features to which mutation will be applied, based on a probability; the default value is 10%. At the end of this process, it calculates the fitness of the population of individuals. All these processes are repeated until the maximum number of generations is reached, the default values are 50 generations. In the last generation, it selects the individual fittest. This individual indicates which features provide the most important information.

B. Multi-sources Process

The responsibility of this process is to search for a data sample from the various data sources registered in the system. This process begins when it receives the keywords of the problem context that requires a dataset. For our case study, this information would be:

Keywords: “energy| electricity| consumption| Spain”.
Endpoints registered in the system:
* EU: https://data.europa.eu/sparql
* ES: http://datos.gob.es/virtuoso/sparql
* CA: https://open.canada.ca/sparql

It prepares queries for each endpoint. For each endpoint, the queries are similar and are specified in SPARQL language:

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
SELECT DISTINCT ?id ?idtype ?type ?url ?title
WHERE{
        ?id a dcat:Distribution.
        ?id dct:title ?title.
        FILTER regex(lang(?title), “en”, "i").
        FILTER regex(?title, Keywords, "i").
        ?id dct:format ?idtype.
        ?idtype rdfs:label ?type.
        ?id dcat:accessURL ?url.
} LIMIT 100
```

Once the keywords and queries are available, they are executed on their respective endpoints. The information obtained is structured with the following data: id, idtype, type, url, title, endpoint, and similarity ratio. To calculate this ratio, we use the Levenshtein distance metric to compare the differences between our keywords and the description or title of the datasets [12]. Table 8 shows the search results and similarity ratios. Finally, the selected dataset is the one with the highest similarity ratio.

TABLE 8: RESULT OF THE SEARCH OF SAMPLES OF DATA.

Title	Endpoint	Ratio

Electricity consumption in municipalities and sectors of Catalonia	EU	79.75
Electricity consumption in municipalities and sectors of Catalonia	ES	79.75
Weekly electricity consumption in Castilla y León	EU	74.6
Hourly electricity consumption in the hospitals of Castile-Leon	EU	73.6
Electricity generation by type of energy	EU	72.8
Electricity Generation	CA	72.4
Installation of electrical energy production. Individualized data	ES	72.25
Electricity Interchange 2016 Update	CA	71.8
net-zero-electricity-provincial-generation-2021	CA	71.4
Hourly electricity demand in Catalonia per MWh	ES	71.0

C. Multi-Datasets Process

The responsibility of this process is to merge datasets to generate a better sample dataset. This process begins when it receives the main dataset selected in the previous phase, and the keywords of the associated context that requires a merged dataset. For our case study, this information would be:

Keywords: “Territorial| Indicators| Spain”.
Dataset: MainDataSample.csv
Similarity Threshold: 85%

Then, it requests a dataset for the Multi-sources process using the keywords of the context associated (see section IV.B). The AssociatedDataSample.csv file obtained will be merged with the MainDataSample.csv file.

To merge both datasets is necessary to identify the columns that relate them. To do that, the first step is to extract and verify the similarity of the titles and data types of the columns (variables) between both datasets. The similarity ratio of the titles is calculated and those that exceed the Similarity Threshold are selected as possible pivot columns. Then, it verifies if the possible pivot columns are of the same data type, and if they match, it selects them as pivots. Table 9 shows the pivots automatically selected by the system for this case study.

TABLE 9: SELECTED PIVOTS.

Nº	**Main Data Sample**		**Associated Data Sample**		**Ratio**
	Title	**Datatype**	**Title**	**Datatype**	
1	Any	integer	Any	integer	100
2	Comarca	string	Comarca	string	100
3	Cod Municipi	integer	Codi Municipi Ine	integer	87

Finally, it merges the datasets using the pivots as a relationship mechanism between the datasets, obtaining the FusionatedDataSample.csv file.

D. FE Module

The responsibility of this module is to generate new features by analyzing the new dataset and selecting the features that provide the most information. This process starts by generating new features. For this case, Table 10 shows the Feature Generation obtained with Trigonometric Features. E.g., tangent, sine and cosine of Consumption kWh. These new features with trigonometric behaviors provide the VAE model with additional information at the time of the construction of the knowledge model.

TABLE 10: GENERATION OF NEW VARIABLES USING TRIGONOMETRY FEATURES

Consumption kWh	**Trigonometric Feature**		
	tan(Consumption kWh)	**sin(Consumption kWh)**	**cos(Consumption kWh)**
0.0000009695	0.0000009695	0.0000009695	1
0.0169895240	0.0169911588	0.0169887066	0.9998556815
0.3876446800	0.4083042838	0.3780089057	0.9258019589
0.8060319000	1.0421430222	0.7215454805	0.6923670409
0.9999999400	1.5574075191	0.8414709523	0.5403023563

Then, new features are extracted. Table 11 shows these new features, the Mean-Based Feature, the Median-Based Feature and the Quartiles-Based Feature. In this specific case, the Mean-Based Feature and Median-Based Feature calculate the distance of "Consumption kWh" with respect to its Mean (0.002746) and Median (0.000151). These two features provide the model with information on the dispersion of values in terms of the central value, either with respect to the mean and the median. The mean is usually better when the values follow a symmetric distribution and the median is better when the values are outliers, as these values distort the mean. The Quartiles-Based Feature determines the quartile to which "Consumption kWh" belongs by dividing the dataset into four equal parts; the first quartile is 25%, ranging from 0 to 0.0000341155. The second quartile is 50%, ranging from 0.0000341155 to 0.0001509117. The third quartile is 75%, ranging from 0.0001509117 to 0.0008732833. The fourth quartile is all values greater than 0.0008732833. This new feature provides information to the model on the behavior of the variables by classifying it into four groups.

TABLE 11: FEATURE EXTRACTION USING MEAN-BASED, MEDIAN-BASED AND QUARTILE-BASED FEATURES

Consumption kWh	**Mean-Based Feature**	**Median-Based Feature**	**Quartiles-Based Feature**
	mean(Consumption kWh)	**median(Consumption kWh)**	**quartiles(Consumption kWh)**
0.0000009695	0.0027454629	0.0001499422	1
0.0169895240	-0.0142430915	-0.0168386122	4
0.3876446800	-0.3848982475	-0.3874937682	4
0.8060319000	-0.8032854675	-0.8058809882	4
0.9999999400	-0.9972535075	-0.9998490282	4

Finally, Table 12 shows a summary of the resulting dataset, with some of the features selected using the Genetic Algorithm. At the beginning of the process, there were 864 features in the data set, and when this method was run, 422 features were selected. Furthermore, it can be observed that most of the selected features are binary, which is an advantage for the VAE model as it works very well with this type of value. Another interesting fact is that in the feature selection for this dataset, the method selected only the features that represent the year 2018 (Any_2018), since there is a lot of information around this date.

TABLE 12: PARTIAL FEATURES SELECTED USING GENETIC ALGORITHM

	Codi Municipi	Any_2018	Provincia_BARCELONA	Codi Sector_5	Codi Sector_6	DescripcionSector_CONSTRUCCIO I OBRES PUBLIQUES
0	0.0	0.0	1.0	0.0	0.0	0.0
1	0.0	0.0	1.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0	0.0	1.0
4	0.0	0.0	1.0	0.0	1.0	0.0
5	0.0	0.0	1.0	0.0	0.0	0.0
…	….	…	…	…	…	…
35975	1.0	0.0	0.0	0.0	0.0	0.0
35976	1.0	0.0	0.0	0.0	0.0	0.0
35977	1.0	0.0	0.0	0.0	0.0	1.0
35978	1.0	0.0	0.0	0.0	1.0	0.0
35979	1.0	0.0	0.0	0.0	0.0	0.0

V. ANALYSIS OF RESULTS

This section is composed of two parts, the first part focuses on the development of experiments with our system in different case studies, and the second part focuses on comparing our work with other approaches.

A. Analysis of SDGS

The aim of this section is to analyze the performance of the SDGS in different case studies that require the generation of synthetic data. In particular, it will generate synthetic data using data samples from Energy, COVID-19, and Industry 4.0 contexts.

A.1. Context of analysis

We considered 3 contexts where an initial dataset exists. The contexts are the following:

- **Energy:** The original dataset is composed of information on electricity consumption in the different municipalities and sectors of Catalonia. In addition, it was merged with other territorial indicators such as population density, economic level, among others. This dataset was obtained thanks to the DSA Module using the LD sources that our system has (more information in III.B, IV.B and IV.C).
- **COVID-19:** The original dataset in this context is composed of information on COVID-19: confirmed cases, deaths, hospitalizations, tests, among many other variables [14]. This dataset was extracted from https://github.com/owid/covid-19-data/tree/master/public/data.
- **Industry 4.0:** This case study corresponds to a production process for the manufacturing of sandwich bread, where the client can customize the wrapper (logo, name, etc.), the quantity, the type of bread (grain bread, white bread, with sesame, with raisins, etc.), among other things. The production process involves devices like the smart conveyor belt that routes the bread to the least busy device wrapping machine to pack the bread using the correct wrapper, smart slicers that slice the bread, the smart printers, and the rest of the devices. The original dataset was composed of information on the status of each process of a production line.

In each context, 4 experiments/cases were realized,

- Two experiments without using the new FE module: One case was using a small data sample (case A) and the other case with a large data sample (case B).
- Two experiments used the new FE module: One case was using Feature Generation and Feature Extraction (case C), and the other used, additionally, Feature Selection (case D).

Regarding the metrics, we used two metrics to analyze the quality of our SDGS. The first metric was the Accuracy of the VAE model built in the different experiments (Section V.A.2). The accuracy measures the performance of the built model based on the total percentage of elements classified correctly. Specifically, the accuracy of VAE is measured by comparing each input with respect to its output, verifying how similar they are, since the learning process of this model is based on passing that input through an Encoder and Decoder, to see if the model is able to reconstruct the input. The second metric was the Two-Sample Hotelling's T-Squared Test, which allows comparing the multivariate means of different populations (datasets with multiple variables). This test allowed performing a hypothesis test to verify the existence of significant differences between two datasets. In our case, it verifies the differences between the training dataset of the VAE model and the synthetic dataset generated by the VAE model in the different experiments (Section V.A.3).

A.2. Performance of the generated VAE models

Table 13 shows the results achieved by the VAE models built with the different datasets. The accuracy obtained using the new FE module offers better results than the models built without using FE. It should also be noted that using the three types of Function Engineering the results are better, as seen in the Energy dataset with a precision of 0.953, in COVID-19 with 0.963, and in Industry 4.0 with 0.961.

TABLE 13: RESULT OF THE ACCURACY OF VAE MODELS

Dataset	Case	Sample	Features	Accuracy
Energy	A	500	95	0.903
	B	29508	95	0.912
	C	29508	864	**0.947**
	D	29508	422	**0.953**
COVID19	A	356	26	0.925
	B	10000	26	0.935
	C	10000	67	**0.957**
	D	10000	59	**0.963**
Industry 4.0	A	250	3	0.918
	B	10000	3	0.946
	C	10000	14	**0.952**
	D	10000	10	**0.961**

A.3. Quality of the Datasets generated

The Two-Sample Hotelling's T-Squared Test checks for significant differences between the dataset used in the training of the VAE model and the dataset with synthetic data generated by the VAE model. This test proposes the following hypotheses,

- Null hypothesis (H_0): the two datasets come from populations with the same features.
- Alternative hypothesis (H_1): the two datasets come from populations with different features.

In addition, the significance level was normally 5% (alpha = 0.05), knowing that the inverse distribution function of F is (1 - alpha) = 0.95. This value allowed determining when H_0 was rejected, i.e. when the probability of H_0< 0.05 or when the calculated value of F was greater than the critical value of the F distribution table when transforming Hotelling's T2 statistic into an F statistic. Rejecting H_0 indicated that at least one of the variables, or a combination of one or more variables working together were significantly different between the datasets.

Finally, we compared the two datasets (training and generated) using the Two-Samples T-Squared Test that has the Python library pingouin[1]. This library executes all the calculations associated with the test and returns the following information:

- T2: Hotelling's T-squared value.
- df1: First degree of freedom.
- df2: Second degree of freedom.
- F: F-distribution value with df1 and df2[2].
- P-value: Probability of the null hypothesis (H_0).

Table 14 shows the results obtained in the different experiments. In this case, we are going to detail the test analysis process for the first experiment since the other experiments are similar. The value 110.02386 in T2 was the result obtained in the different matrix operations of the test with both datasets. This value when is transformed the Hotelling T2 statistic into an F statistic, we obtained the value 1.04906202. Having the F value, we had two methods to determine if there were significant differences between the datasets:

- The first method required finding the critical value in the F distribution table with the degrees of freedom df1 = 95 and df2 = 904 and the default value of alpha 0.05. In this case, we used the calculator offered at https://datatab.net/tutorial/f-distribution, where it gave an F-critical of 1.268. Now, if F > F-critical then the H_0 is rejected. In this case, it was not rejected; therefore, there were no significant differences between the datasets.
- The second method required calculating the P-value. In this case, the library gave us this value (0.36056018). Now, if P-value < 0.05 then the H_0 was rejected. In this case, it was not rejected, so there were no significant differences between the datasets.

TABLE 14: TWO-SAMPLE HOTELLING'S T-SQUARED TEST RESULT

Dataset	Case	Features	Two-Sample Hotelling's T-Squared Test

[1] https://pingouin-stats.org/build/html/generated/pingouin.multivariate_ttest.html
[2] https://datatab.net/tutorial/f-distribution

			T2	F	df1	df2	P-value	Hypothesis
Energy	A	95	110.02386	1.04906202	95	904	0.36056018	H_0
	B	95	101.02386	1.06171521	95	58920	0.32111891	H_0
	C	864	895.56456	1.02137516	864	58151	0.32508918	H_0
	D	422	425.45645	1.00099831	422	58593	**0.48518434**	H_0
COVID-19	A	26	22.367299	0,82998916	26	685	**0.70947055**	H_0
	B	26	24.9877204	0.95986472	26	19973	0.52147351	H_0
	C	67	68.346573	1.01673144	67	19932	0.43904006	H_0
	D	59	57.346573	0.96915681	59	19940	**0.54289117**	H_0
Industry 4.0	A	3	1.3564567	0.45033636	3	496	0.71717036	H_0
	B	3	1.8544546	0.61808971	3	19996	0.60320568	H_0
	C	14	10.425738	0.74421147	14	19995	0.73092497	H_0
	D	10	6.725738	0.67227112	10	19989	**0.75131788**	H_0

When analyzing the experiments knowing that when P-values are less than 0.05 (tend to zero), H_0 is rejected, it is observed that the results using the FE module with the three Feature Engineering methods tend to obtain P-values closer to 1. For example, Energy with 0.48518434, COVID-19 with 0.54289117 and Industry 4.0 with 0.75131788. Only in COVID-19, the first experiment of this dataset with fewer samples performed better (0.70947055). In general, the P-value results indicated that the FE module contributes to improve the quality of the synthetic data, since SDGS was able to generate synthetic data very close to the data used to train the VAE model of the system, allowing the training of knowledge models in cases where there was a problem of insufficient data.

On the other hand, another test was carried out to evaluate the quality of the synthetic data generated. In particular, its use in the construction of classification models is verified. This test consisted of training and validating a Random Forest-based Classification model using the synthetic data, and then testing the quality of the model using the original data, to verify if it is capable of classifying them correctly. Table 15 shows the results achieved in this test using different datasets. In general, all models obtain excellent Accuracy and F1 score, both in training with the synthetic data and in the tests carried out with the original dataset. It is consistently observed that the models created with the synthetic data generated through the different processes of Feature Engineering (cases C and D) obtain better results, as seen for Energy with values of 0.936 and 0.883, in COVID-19 with 0.952 and 0.923, and in Industry 4.0 with 0.959 and 0.928, respectively. Thus, it is observed that the models with the synthetic data generated by the FE module offer better results than the models that do not use FE.

TABLE 15: RESULT OF THE ACCURACY OF CLASSIFICATION MODELS

Dataset	Case	Sample	Features	Training	Testing

				Accuracy	F1	Accuracy	F1
Energy	A	500	95	0.894	0.869	0.833	0.795
	B	29508	95	0.902	0.885	0.856	0.803
	C	29508	864	**0.923**	**0.893**	**0.877**	**0.827**
	D	29508	422	**0.936**	**0.912**	**0.883**	**0.831**
COVID19	A	356	26	0.905	0.888	0.872	0.844
	B	10000	26	0.924	0.901	0.895	0.858
	C	10000	67	**0.941**	**0.914**	**0.908**	**0.869**
	D	10000	59	**0.952**	**0.932**	**0.923**	**0.881**
Industry 4.0	A	250	3	0.903	0.892	0.864	0.815
	B	10000	3	0.927	0.903	0.889	0.862
	C	10000	14	**0.944**	**0.927**	**0.916**	**0.878**
	D	10000	10	**0.959**	**0.940**	**0.928**	**0.890**

B. Comparative analysis

This section carries out a comparison with other similar works in the literature. For that, two types of comparisons were carried out.

B.1. Qualitative comparison

In this paper are identified the following four evaluation criteria for a qualitative comparison with other works (Table 16):

- C1: Sample data acquisition mechanism.
- C2: Sample data preparation methods.
- C3: Feature engineering techniques.
- C4: Metrics used to measure the quality of synthetic dataset generation.

TABLE 16: QUALITATIVE ANALYSIS WITH PREVIOUS WORKS

Work	C1	C2	C3	C4
[9]	Dataset	Not indicated	Not indicated	Mean and Standard Deviation of

				Minutiae, False Acceptance Rate (FAR) and True Acceptance Rate (TAR)
[10]	Dataset	Not indicated	Not indicated	K-mean clustering algorithm and Area Under the Curve (AUC)
[15]	Dataset	Data Normalization and Data Scaling	Not indicated	Dice loss, IoU score, F-score, Accuracy, Recall, and Precision
[16]	Ontology	Not indicated	Not indicated	Recall, Precision, F-score, Average Precision and IuO score
[17]	Dataset	Not indicated	Not indicated	Accuracy
[18]	Dataset	Not indicated	Not indicated	Validity, Novelty, Uniqueness, Reconstruction and Score
[20]	Dataset	Not indicated	Not indicated	Discriminative score and Predictive scores
[21]	Dataset	Data Normalization and Categorical Data	Not applied	Data heatmap, Correlation, Charts by features and Accuracy
Our approach	Automatic extraction of datasets using LD and REST API	Data Normalization and Categorical Data	Feature Generation (5 techniques), Feature Extraction (3 techniques) and Feature Selection (4 techniques)	Two-Sample Hotelling's T-Squared, F1, and Accuracy

Regarding criterion C1, most of them used datasets with data samples prepared for a specific study. However, in [16], the authors used an ontology that stores all the knowledge and from there, the sample dataset was extracted for the generation of synthetic data. In our work, the datasets were automatically extracted from external sources using LD. In addition, our dataset extraction mechanism allows the fusion of several datasets for a single data sample, allowing extending the features of the synthetic dataset. With respect to C2, most of the works do not indicate what preprocessing is done with the dataset, although, due to the generation model they use, all of them should perform a normalization of the data. Additionally, in [15] was carried out a pyramidal rescaling of the input data, obtaining information from different levels of precision. In [21] and our work, several transformations were carried out.

Regarding criterion C3, in [21], they do not apply feature engineering techniques to improve the generation of synthetic data, and in the rest of the works, they do not indicate if they use any technique. In our case, the system has several methods for feature extraction, generation and selection (see sections III.D and IV.D). Concerning criterion C4, the works [9], [10], [18] and [21] used specific metrics for the type of problem to be solved. In our case, we used the two-sample Hotelling T-squared to determine in general terms the degree of similarity of the synthetic dataset with respect to the sample dataset used to train the VAE model, and the accuracy to evaluate the quality of the knowledge model. Some papers used the same metrics (e.g., [15], [17], [21]), and [15] and [16] use Recall, Precision and F-score.

B.2. Quantitative comparison

This section focuses on a quantitative comparison with other works that used the same metric as ours. We have used the same context of study in each case to carry out this test. Table 17 presents the metrics obtained in these works.

TABLE 17: QUANTITATIVE ANALYSIS WITH PREVIOUS WORKS

Work	Accuracy
[15]	96%
[17]	95%
[21]	No indicate
Our approach	92% for dataset from [15] 96% for dataset from [17]

Regarding the works [15], the results were a little lower, and that difference perhaps came from the generation model used in each case (in [15], a generative adversarial network, and in our case, a VAE technique). In [17] and our work, the values of the accuracy metric were higher than 95%, i.e., they presented very good results. In general, in our work, the results were presented in a range of values ranging from 92% to 96%, these results depended on the characteristics of each dataset used to train our synthetic data generator. Although [21] uses the same metric, the value achieved is not presented in the paper.

VI. CONCLUSIONS

The extended SDGS provides an automated architecture for the generation of synthetic data using the LD paradigm as an identifier and extract of data source, and VAE as a generator. In this work, three extensions were implemented. The multisource extension allows extracting sample data automatically from different data sources. In the case of our architecture, it has implemented an extraction mechanism based on LD queries, where queries were built using semantic tags that allow extracting datasets with the features of the context of the problem. In addition, this architecture integrates a mechanism that allows datasets to be merged, which favors the expansion of the features of a dataset. This multi-dataset extension was based on the automatic search for pivots that serve as a common point for the union of the datasets. Finally, the third extension adds to the SDGS architecture a new module with Feature Engineering processes that allowed generating and extracting of new features by applying different techniques to the dataset. Also, it allows selecting the features that offer more information to the VAE model. This module included five feature generation techniques, three feature extraction techniques and four feature selection techniques.

A case study was described in detail to show the new capabilities of the SDGS architecture using the three expansions specified in this research. Finally, two metrics were used, the first one was the accuracy for analyzing the performance of the models generated with the new SDGS functionalities, obtaining results above 90%. The second one was the Two-Sample Hotelling's T-Squared Test to determine the quality of the synthetic data generated by the system, obtaining synthetic datasets very similar to the datasets used to train the VAE models. Also, a comparative analysis was carried out with related works of generation of synthetic data, detailing the different characteristics present in each of them.

SDGS has some limitations, for example, it only feeds on datasets from repositories obtained using the LD paradigm. The second limitation is in its pivot selection mechanism for merging datasets. In addition to checking that the columns were of the same data type, only the column names were compared using the syntactic similarity of the two strings. The third limitation is that the architecture only uses default values in its different configuration parameters (e.g., for the VAE technique).

Future works will test other types of data sources for the multi-source process of the SDGS architecture, allowing the construction of sample data from databases, and knowledge graphs, among other current technologies associated with dataset repositories. For the multi-dataset process, it will test other pivot selection mechanisms with innovative techniques in text interpretation to allow semantic analysis of text strings, use of synonyms, translations, among others. Likewise, it will use the LD paradigm for the construction of a new intelligent layer, called Meta learning, which will allow the automatic adjustment of the architecture's parameters to optimize the generation of synthetic datasets according to the desired context and the knowledge model to be built. Finally, other future work should evaluate the quality of synthetic data obtained in contexts other than classification tasks, such as, for example, to build predictive or prescriptive models (in this particular area, there is very little data).

FUNDING

Jose Aguilar was partially supported by grant 22-STIC-06 (HAMADI 4.0 project) funded by the STIC-AmSud regional program.

REFERENCES

[1] Dos Santos, R., Aguilar, J., & R-Moreno, M. D. (2022). A synthetic Data Generator for Smart Grids based on the Variational-Autoencoder Technique and Linked Data Paradigm. In 2022 XVLIII Latin American Computer Conference (CLEI) https://doi.org/10.1109/CLEI56649.2022.9959918

[2] Avazpour, I., Grundy, J., & Zhu, L. (2019). Engineering complex data integration, harmonization and visualization systems. Journal of Industrial Information Integration (Vol. 16, pp. 100103). https://doi.org/10.1016/j.jii.2019.08.001

[3] Izquierdo, Y., Casanova, M. A., García, G., Dartayre, F., & Levy, C. H. (2017). Keyword Search over Federated RDF Datasets. In ER Forum/Demos (pp. 86-99). https://dblp.org/rec/conf/er/IzquierdoCGDL17

[4] Rao, G., Zhang, L., Zhang, X., Li, W., Li, F., & Tao, C. (2019). A Multi-Source Linked Open Data Fusion Method for Gene Disorder Drug Relationship Querying. In SEPDA@ ISWC (pp. 31-35). https://dblp.org/rec/conf/semweb/RaoZZLLT19

[5] Chen, Y. (2022, May). Linked Data Fusion Based on Similarity Calculation and K-Nearest Neighbor. In Journal of Physics: Conference Series (Vol. 2221, No. 1, pp. 012043). IOP Publishing. https://doi.org/10.1088/1742-6596/2221/1/012043

[6] Nishimaki, K., Ikuta, K., Onga, Y., Iyatomi, H., & Oishi, K. (2022, October). Loc-VAE: Learning Structurally Localized Representation from 3D Brain MR Images for Content-Based Image Retrieval. In 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 2433-2438). IEEE. https://doi.org/10.1109/SMC53654.2022.9945411

[7] Van Dao, T., Sato, H., & Kubo, M. (2022). An Attention Mechanism for Combination of CNN and VAE for Image-Based Malware Classification. IEEE Access (Vol. 10, pp. 85127-85136). https://doi.org/10.1109/ACCESS.2022.3198072

[8] Hadipour, H., Liu, C., Davis, R., Cardona, S. T., & Hu, P. (2022). Deep clustering of small molecules at large-scale via variational autoencoder embedding and K-means. BMC bioinformatics (Vol. 23, No. 4, pp. 1-22). https://doi.org/10.1186/s12859-022-04667-1

[9] Engelsma, J. J., Grosz, S. A., & Jain, A. K. (2022). PrintsGAN: Synthetic fingerprint generator. IEEE Transactions on Pattern Analysis and Machine Intelligence (pp. 1-14). https://doi.org/10.1109/TPAMI.2022.3204591

[10] Shah, P., Ullah, H., Ullah, R., Shah, D., Wang, Y., Islam, S., Gani A. Rodrigues, J. J. (2022). DC-GAN-based synthetic X-ray images augmentation for increasing the performance of EfficientNet for COVID-19 detection. Expert Systems (Vol. 39, No. 3, pp. e12823). https://doi.org/10.1111/exsy.12823

[11] Aguilar J, Jerez M, Exposito E., Villemur T, (2015) CARMiCLOC: Context Awareness Middleware in Cloud Computing, In *2015 Latin American Computing Conference (CLEI)*, doi: 10.1109/CLEI.2015.7360013.

[12] Morales, L., Ouedraogo, C., Aguilar, J., Chassot C, Medjiah S., Drira K. (2019) Experimental comparison of the diagnostic capabilities of classification and clustering algorithms for the QoS management in an autonomic IoT platform. Service Oriented Computing and Applications (Vol. 13, pp. 199–219)

[13] Aguilar, J.; Salazar, C.; Velasco, H.; Monsalve-Pulido, J.; Montoya, E. (2020) Comparison and Evaluation of Different Methods for the Feature Extraction from Educational Contents. Computation,(vol 8) https://doi.org/10.3390/computation8020030

[14] Quintero Y, Ardila D., Camargo E., Rivas F, Aguilar J. (2021) Machine learning models for the prediction of the SEIRD variables for the COVID-19 pandemic based on a deep dependence analysis of variables, Computers in Biology and Medicine (Vol. 134). https://doi.org/10.1016/j.compbiomed.2021.104500.

[15] Thambawita, V., Salehi, P., Sheshkal, S., Hicks, S., Hammer, L., Parasa, S., deLange T., Halvorsen P., Riegler, M. (2022). SinGAN-Seg: Synthetic training data generation for medical image segmentation. PloS one (Vol. 17, No. 5, p. e0267976). https://doi.org/10.1371/journal.pone.0267976

[16] Hoeser, T., & Kuenzer, C. (2022). SyntEO: Synthetic dataset generation for earth observation and deep learning–Demonstrated for offshore wind farm detection. ISPRS Journal of Photogrammetry and Remote Sensing (Vol. 189, pp. 163-184). https://doi.org/10.1016/j.isprsjprs.2022.04.029

[17] Pfitzner, B., & Arnrich, B. (2022). DPD-fVAE: Synthetic Data Generation Using Federated Variational Autoencoders With Differentially-Private Decoder. arXiv preprint arXiv:2211.11591. https://doi.org/10.48550/arXiv.2211.11591

[18] Ma, C., & Zhang, X. (2021, October). GF-VAE: a flow-based variational autoencoder for molecule generation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 1181-1190). https://doi.org/10.1145/3459637.3482260

[19] Morales L., Aguilar J., Garcés-Jiménez A., Gutierrez De Mesa J., Gomez-Pulido J. (2020), "Advanced Fuzzy-Logic-Based Context-Driven Control for HVAC Management Systems in Buildings, *IEEE Access* (vol. 8, pp. 16111-16126) doi: 10.1109/ACCESS.2020.2966545.

[20] Desai, A., Freeman, C., Wang, Z., & Beaver, I. (2021). Timevae: A variational auto-encoder for multivariate time series generation. arXiv preprint arXiv:2111.08095. https://doi.org/10.48550/arXiv.2111.08095

[21] Akkem, Y., Biswas, S. K., & Varanasi, A. (2024). A comprehensive review of synthetic data generation in smart farming by using variational autoencoder and generative adversarial network. Engineering Applications of Artificial Intelligence (Vol. 131, pp. 107881). https://doi.org/10.1016/j.engappai.2024.107881

[22] Aref, S., J. Shortle, L. Sherry. (2024). Generating synthetic flight tracks for collision risk safety analysis: Variational autoencoders with a single seed track. In Proceedings of the Integrated Communications, Navigation, and Surveillance Conference, Herndon, VA.

[23] Hubert, N., Monnin, P., D'aquin, M., Monticolo, D., & Brun, A. (2024, May). PyGraft: Configurable Generation of Synthetic Schemas and Knowledge Graphs at Your Fingertips. In Semantic Web-21st International Conference, ESWC 2024. https://doi.org/10.5281/zenodo.10243209

[24] Aguilar J., Garcès-Jimènez A., Gallego-Salvador N., De Mesa J., Gomez-Pulido J., Garcìa-Tejedor A. (2919) Autonomic Management Architecture for Multi-HVAC Systems in Smart Buildings, *IEEE Access*, (vol. 7, pp. 123402-123415), 10.1109/ACCESS.2019.2937639

[25] Hoseini, S., Theissen-Lipp, J., & Quix, C. (2024). A survey on semantic data management as intersection of ontology-based data access, semantic modeling and data lakes. Journal of Web Semantics, 100819. https://doi.org/10.1016/j.websem.2024.100819

[26] Gourabpasi, A. H., & Nik-Bakht, M. (2024). BIM-based automated fault detection and diagnostics of HVAC systems in commercial buildings. Journal of Building Engineering, 87, 109022. https://doi.org/10.1016/j.jobe.2024.109022

[27] Marco R., Sakinah S. Ahmad S. (2022) Conditional Variational Autoencoder with Inverse Normalization Transformation on Synthetic Data Augmentation in Software Effort Estimation, International Journal of Intelligent Engineering and Systems, (Vol.15, No.3).

[28] Kuo N., Garcia F., Sönnerborg A., Böhm M, Kaiser R., Zazzi M., Polizzotto M., Jorm L., Barbieri S., (2023) Generating synthetic clinical data that capture class imbalanced distributions with generative adversarial networks: Example using antiretroviral therapy for HIV, Journal of Biomedical Informatics (Vol 144), https://doi.org/10.1016/j.jbi.2023.104436.

[29] Panfilo D. Boudewijn A.; Saccani S.; Coser A.; Svara B.; Rossi C. et al., (2023) A Deep Learning-Based Pipeline for the Generation of Synthetic Tabular Data," IEEE Access, (Vol. 11, pp. 63306-63323), doi: 10.1109/ACCESS.2023.3288336.

[30] Eigenschink P, Reutterer T, Vamosi S, Vamosi R., Sun C. Kalcher K. (2023) Deep Generative Models for Synthetic Data: A Survey, IEEE Access (vol. 11, pp. 47304-47320, doi: 10.1109/ACCESS.2023.3275134.