A Practical Framework for Generating and Validating Synthetic Databases: Application to Freight Transport

Liviu-Ioan ZECHERU, Cristian-Eugen CIUREA
Bucharest University of Economic Studies
liviu.zecheru@net.ase.ro, cristian.ciurea@ie.ase.ro

Synthetic databases are increasingly used in research and industry to support testing, training, and analysis without exposing sensitive information. This paper proposes a practical framework for generating and validating synthetic databases, structured around a pipeline that ensures structural consistency, business relevance, and reproducibility. The framework is illustrated through a case study on freight transport in Romania, where a relational model was designed to capture entities such as clients, trains, conductors, and transported goods. A Python-based generator was developed to populate the database with realistic values under domain-specific constraints (e.g., valid national identifiers, capacity limits, distinct departure/arrival stations). Validation is focused on structural integrity, query performance, and privacy preservation. The results show that the generated dataset is both realistic and safe for academic or enterprise use, while the methodology is transferable to other economic and business contexts.

Keywords: *Synthetic Databases, Data Generation, Freight Transport, Data Validation*
DOI: 10.24818/issn14531305/29.3.2025.01

1 Introduction

Modern data-driven projects require access to large and diverse datasets, yet obtaining real-world data often poses challenges due to privacy restrictions and scarcity [1]. *Synthetic data* offers a compelling solution to this problem. Synthetic data is exactly what it sounds like: information artificially created – usually via algorithms, statistical models, or generative AI – rather than collected from real-world events. By analyzing patterns in source data and recreating them, synthetic datasets can mimic the statistical properties of real data without exposing any actual sensitive records. This approach enables researchers and businesses to perform analysis, understand customer behaviors, develop new products or even generate new revenue using *fake data generated from real data* – all while sidestepping many privacy issues inherent to using true personal data.

The use of synthetic data has grown rapidly in recent years and is becoming mainstream in industry. Analysts estimated that by 2024, 60% of data used for analytics and AI development will be synthetically generated [2]. This trend reflects the considerable advantages synthetic data provides in terms of speed and scale. In contrast to traditional data collection, generating data artificially can be a lower-cost, faster way to obtain vast quantities of training or test data. In fact, synthetic data has the potential to *"turbocharge the data-driven transformation of every industry"* by serving as the foundation for training machine learning models and AI systems. Organizations across domains – from finance and healthcare to retail and technology – are already exploring synthetic data to fuel innovation. For example, companies have used synthetic datasets to improve multilingual speech recognition for voice assistants and to create shareable, anonymized healthcare records for research, thereby enabling new analyses that would be infeasible with strictly real data. Synthetic data's flexibility allows it to fill gaps where real data is unavailable or insufficient, amplifying opportunities for business insights and AI development.

One of the most celebrated benefits of synthetic data is its ability to preserve privacy and enhance data security. By design, a well-generated synthetic dataset contains *no real personal identifiers*, yet maintains the statistical realism of the original data. Thus, *"synthetic data's most obvious benefit is that it* *eliminates the risk of exposing critical data and compromising the privacy and security of companies and customers"* [3]. In other words, synthetic data can give a clear impression of real-world phenomena without ever revealing the underlying sensitive information. This makes it invaluable for organizations dealing with strict data protection regulations. Furthermore, synthetic data allows machine learning models to be trained on large-scale examples far more quickly and safely. With synthetic data, a company can rapidly train and test models on massive datasets, accelerating the development and deployment of AI solutions. The result is faster iteration of analytics projects, unhindered by the delays and compliance checks associated with using confidential real datasets. In summary, synthetic data enables both privacy preservation and technical progress hand-in-hand – a combination highly attractive to modern businesses aiming to be data-driven yet compliant.

Beyond conventional structured data (e.g. tables of transactions or customer profiles), synthetic data generation now extends to unstructured domains such as images and audio. Advances in generative models, particularly Generative Adversarial Networks (GANs), have enabled the creation of photorealistic synthetic media that is often indistinguishable from real data. A striking example is the website *"This Person Does Not Exist"*, an AI face generator powered by Nvidia's StyleGAN algorithm. Each refresh of that site produces a completely new human face that looks authentic despite belonging to no real person. Such technology showcases how synthetic data can be used to generate realistic portraits for fictional individuals, providing visual anonymity while preserving realism. In general, the same techniques can synthesize any complex data modality – from voice recordings to sensor data – broadening the scope of synthetic data's applications. In business settings, these innovations mean organizations can create entire fictional yet plausible datasets (including images or multimedia) for testing and development purposes. For instance, banks could generate synthetic customer profiles complete with profile pictures, or automotive companies could simulate video footage of driving scenarios – all without involving any real people or events. This versatility of synthetic data generation opens the door to new use-cases and safer sharing of data across departments or with partners.

It is important to note, however, that synthetic data is not a silver bullet and must be used with care. The process of generating high-fidelity synthetic data can be complex, and poor implementation can introduce new risks. Researchers caution that without proper safeguards, synthetic data could inadvertently reveal patterns traceable to real individuals, thus undermining the privacy benefits is purports to offer. For example, if the generation process is naively done, a synthetic dataset might retain identifiable details from the original data, leading to potential privacy violations [4]. There are also concerns about bias and fidelity: if the source data is biased or limited, the synthetic data will reflect those issues, possibly amplifying unfairness or misrepresenting rare cases. Moreover, recent studies even suggest that over-reliance on AI-generated data can cause feedback loops (e.g. *model collapse* in AI models repeatedly trained on their own synthetic outputs). These security and quality implications underscore the need for rigorous validation of synthetic data. In practice, organizations should combine synthetic data generation with robust privacy checks (such as ensuring no record in synthetic data maps uniquely to a real individual) and with careful evaluation of the synthetic data's statistical accuracy and fairness. When done correctly, synthetic data can indeed be, as one industry toolkit describes a *"safer, smarter solution"* for data-driven development – but responsible AI principles must guide its use [5].

In this context, our work explores a practical approach to generating complex synthetic dataset while addressing the above opportunities and challenges. We focus on the creation of a multi-table synthetic database modeled after a real-world scenario, in which one table (termed **"Conductors"**) contains fictional person records. Each synthetic person entry includes various attributes (such as name, contact information, etc.) as well as a photograph, demonstrating how both structured data and images can be generated artificially. To ensure realism, we leveraged a combination of techniques: web scraping of public information to gather authentic-looking data points, programmatic data synthesis using libraries and custom scripts, and incremental refinement of synthetic profile images using multiple generative services. In particular, the photo field for each "Conductor" was populated by experimenting with several AI image-generation tools and selecting the most convincing results – ultimately using the StyleGAN-based *ThisPersonDoesNotExist* generator to obtain high-fidelity human faces that have no real identity. We document the methodology of this process, including the challenges encountered (such as balancing realism with anonymity and automating web data extraction), and how they were overcome. The remainder of this article is organized as follows: **Section 2** reviews the relevant literature and background on synthetic data generation and its applications. **Section 3** details the research methodology, including the data sourcing (web scraping), synthetic data generation pipeline, and the design of the synthetic database (with an outline of its schema and entity-relationship structure). **Section 4** discusses potential future improvements and extensions of this work, and **Section 5** concludes with final observations and the key takeaways of our study.

2 Literature review

2.1 AI and Digital Transformation in the Modern Economy

Over the past decade, artificial intelligence has rapidly moved from research labs to mainstream business operations. Surveys show that AI adoption is now widespread in enterprises: more than three-quarters of organizations report using AI in at least one business function [6]. In fact, AI is increasingly seen as a cornerstone of innovation in the business domain, enabling a transition toward smarter and more sustainable practices. This trend is underpinned by a booming industry – the global AI market was estimated at around USD 196.6 billion in 2023, with projections of a staggering 36.6% compound annual growth through 2030 [7]. Such growth reflects the high expectations that companies place on AI technologies to drive efficiency, productivity, and competitive advantage in the modern economy.

In parallel, organizations are embracing a broader wave of digital transformation often termed *Industry 4.0*. This concept denotes the convergence of AI with other advanced technologies – including the Internet of Things (IoT), cloud computing, big data analytics, and robotics – to create smart, data-driven enterprises. Manufacturers and service providers alike are integrating these technologies throughout their operations, enabling real-time decision-making and automation on an unprecedented scale. For example, IoT sensors combined with AI-driven analytics allow firms to perform predictive maintenance on equipment, reducing downtime and improving operational efficiency. Similarly, AI-powered data analysis across previously siloed business systems (from supply chain logistics to customer service) yields new insights that inform better strategic decisions. Industry 4.0 thus represents a fusion of modern technologies in economic activities, bringing increased agility, productivity, and innovation across sectors. In summary, the contemporary literature highlights that businesses which leverage AI and related technologies can "revolutionize the way [they] operate" through automation, self-optimization and enhanced data visibility, ultimately achieving levels of efficiency and responsiveness not previously possible [8].

2.2 Synthetic Data Generation and Application

As AI becomes ubiquitous in business, the availability of high-quality data emerges as a critical enabler for model training, testing, and validation. However, real-world data can often be scarce, expensive, or sensitive (e.g. subject to privacy regulations), especially in fields like finance or healthcare. In this context, synthetic data has rapidly gained prominence as a viable solution. Forecasts also underline this trend: as shown in Fig. 1, the synthetic data market is expected to grow steadily between 2023 and 2030, confirming its adoption across industries. Synthetic data refers to artificially generated datasets that mimic the statistical properties of real data while omitting any real personal or confidential information.

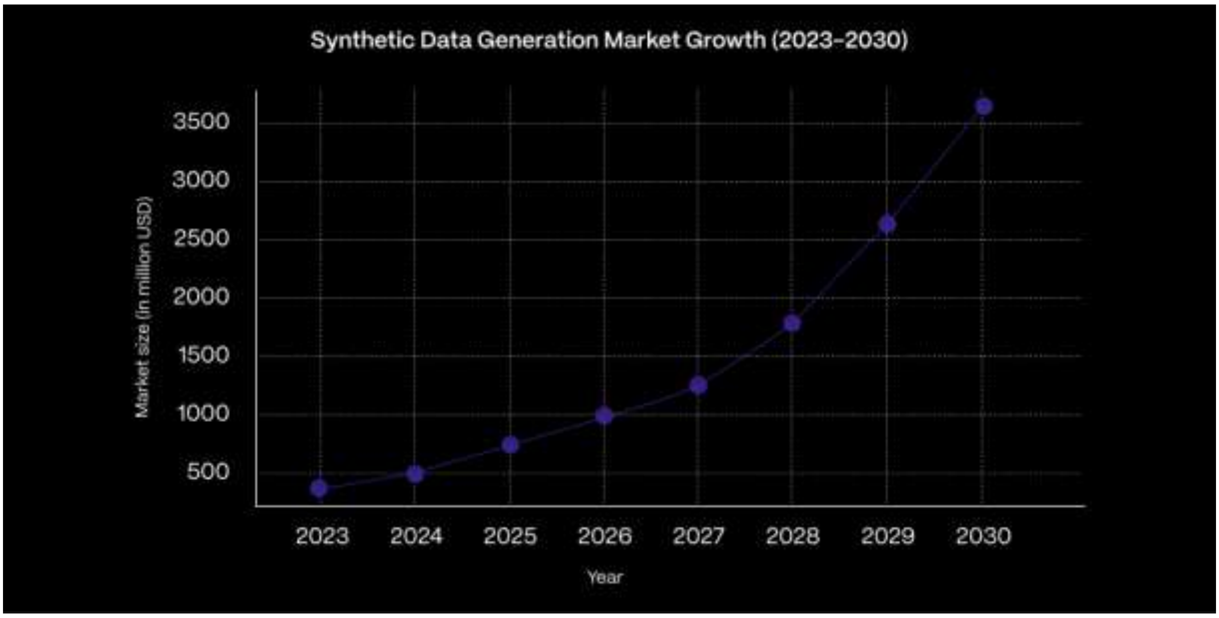


Fig. 1. Synthetic Data Generation Market Growth (2023-2030) **[9]**

Recent research underscores the surge of interest in this approach: major tech organizations such as OpenAI, Apple, Microsoft, Google, Meta, and IBM have all reported using synthetic data in their AI development pipelines, highlighting its growing importance in practice. The appeal lies in its ability to address key data challenges. By replacing or augmenting real datasets with realistic simulated data, organizations can sidestep privacy risks and compliance barriers, enabling data sharing and AI experimentation without exposing sensitive records. For instance, in the financial sector – where strict regulations and proprietary data silos limit information sharing – there is a “growing need for effective synthetic data generation” to produce datasets that preserve the statistical patterns of original customer data while protecting privacy [10]. In short, synthetic data offers a pathway to harness AI capabilities in domains where real data is constrained, by generating realistic yet safe proxies of the information needed.

Underlying the rise of synthetic data are advances in generative AI models. Techniques such as Generative Adversarial Networks (GANs) and variational autoencoders have dramatically improved the fidelity of synthetic data over the past decade. These models learn the complex patterns of real datasets and can produce entirely new data points (images, text, numeric records, etc.) that are statistically credible. As a result, synthetic data is now being applied in a wide range of fields. Researchers report successful applications in robotics, autonomous driving, finance and healthcare, among others. A key benefit is the ability to create or augment data for scenarios that are rare or hard to observe. For example, in machine learning for image recognition, generating additional synthetic samples of rare defects or uncommon cases can help alleviate class imbalance and improve model robustness. Likewise, to combat bias, one can enrich training datasets with simulated examples of underrepresented groups or conditions, thereby enhancing fairness in AI outcomes [11]. From an industry perspective, synthetic data is increasingly seen as a “controlled and scalable data source” for AI development, offering an endless supply of tailored data without the cost and delay of traditional data collection.

In an economic and business context, synthetic data generation opens new avenues for innovation and experimentation. Companies can use synthetic datasets to simulate realistic business scenarios and test "what-if" analyses without relying on proprietary or live data. A pertinent example is in supply chain and logistics management: organizations are building digital twin models of their operations (warehouses, transport routes, inventories) and then feeding them with AI-generated synthetic scenarios – such as sudden demand surges, machine failures, or transportation disruptions – to evaluate system resilience and optimize contingency plans. This approach allows firms to stress-test their strategies safely, anticipating problems and refining processes in a virtual environment before they occur in reality. Studies report that such AI-driven simulations can reveal bottlenecks and yield faster insights, ultimately leading to more robust and efficient supply chains. More broadly, synthetic data empowers organizations to develop and validate data-intensive applications (for example, customer behavior models, risk analysis tools, or AI-driven decision support systems) even when real datasets are limited or cannot be shared. The literature thus portrays synthetic data as a powerful catalyst for business analytics and AI development – one that complements the overall trend of digital transformation. By enabling richer training data and safe experimentation, synthetic data techniques help businesses fully leverage modern AI models and technologies in pursuit of innovation and competitive advantage. In summary, the confluence of advanced AI models and modern data-generation techniques is providing organizations with unprecedented capabilities to generate realistic synthetic data, which in turn accelerates research and development in various economic sectors while safeguarding privacy and integrity. This confluence is a recurring theme in recent literature and forms a foundation for the present work.

3 Research methodology

This section details the data sourcing, the synthetic data generation pipeline, and the design of the synthetic database underpinning our railway transport scenario. We also discuss the tools and technologies used, as well as steps taken to ensure the synthetic data's realism and integrity. Finally, we outline how one might validate and benchmark the resulting dataset. The approach follows state-of-the-art practices in synthetic data generation, combining web scraping, programmatic data synthesis, and generative AI techniques.

Data Sourcing and Collection

The first step was gathering real-world data to ground the synthetic dataset in reality. We employed web scraping techniques to collect reference data from multiple sources. For example, lists of personal names were obtained to generate realistic identities for train conductors. Using Python-based scraping, we collected extensive lists of Romanian first names (separate lists for males and females) and common surnames from online databases and public resources. This ensured that the synthetic identities would reflect authentic naming patterns (e.g., containing Romanian diacritics and culturally appropriate name frequencies). Additionally, we sourced geographical data for railway stations and routes. Key location names (major cities and train stations) were compiled, either from public transport websites or open data repositories, to serve as origin/destination points in the railway network. These real place names allow the creation of plausible routes (e.g. București-Brașov) rather than fictional locales.

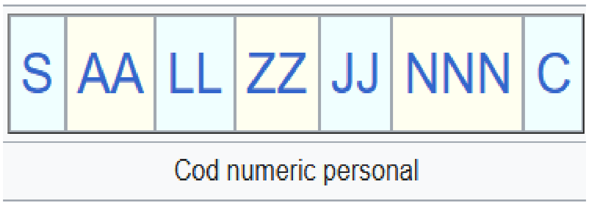
Another critical dataset was a catalog of railway freight goods. Instead of relying on limited or random examples, we took an innovative approach: leveraging data from the domain of transportation simulation games. We curated a list of cargo types from SCS Software's popular simulators *Euro Truck Simulator 2* and *American Truck Simulator*. These games offer rich and varied cargo lists (e.g. lumber, fuel, machinery, food products), which we scraped from a fan-maintained wiki [12]. By parsing these lists (with permission credited to the game developers and community), we obtained over a hundred distinct freight item names along with their unit measures. The descriptions of the goods were generated using the Selenium automation tool alongside ChatGPT and Python. At the time, ChatGPT was on its first steps, not having a publicly available API, so this idea crossed our minds. We prompted for accurate descriptions of the goods and retrieved the answers programmatically. This provided a realistic distribution of goods that might be transported via rail – from agricultural products to hazardous materials – greatly enhancing the authenticity of our freight transport data.

Data cleaning was performed on all scraped inputs. We removed any duplicate entries and standardized formatting (for example, ensuring consistent capitalization, and removing any special characters that could conflict with SQL syntax, such as apostrophes). The name lists were stripped of any extraneous metadata so that they contained one name per line. The cargo list from the game wiki was tabulated into a structured format (CSV), separating each item's name, a short description, and a typical unit of measurement (e.g. tons, liters, packages). These curated lists became the foundation for the next phase of synthetic data generation.

Synthetic Data Generation Pipeline

With source data in hand, we designed a multi-stage synthetic data generation pipeline to create the full database content. The pipeline was implemented in Python as a script (see our GitHub repository for reference [13]) that programmatically produces synthetic records and writes them as SQL insert statements. The generation process was broken into clear steps, executed in a logical sequence to satisfy all relational dependencies:

1. **Generation of Synthetic Identities**: We created lifelike personal profiles for train conductors (the railway staff). Each synthetic conductor was assigned a first name and last name by randomly drawing the real name lists (ensuring gender consistency, i.e. female first names for female profiles, etc.) To each profile we also assigned a birthdate and gender, from which we derived a Personal Numeric Code (CNP) – the Romanian national identification number. The CNP format encodes an individual's gender and date of birth within its digits, so we leveraged this structure to enhance authenticity. For each conductor, the script generates a random birth date (within a plausible range for an active railway employee, e.g. 24 to 60 years of age) and determines the CNP digits accordingly (including the correct gender digit and birth year/month/day components). A modulo-11 checksum is computed for the final CNP digit to ensure each code is formally valid. The structure of the Romanian CNP used in our synthetic data is summarized in Fig. 2. By imitating the official CNP scheme, our synthetic identities attain high fidelity – each fake conductor has an ID number that could pass as real, matching their age and gender.



S	AA	LL	ZZ	JJ	NNN	C

Cod numeric personal

Fig. 2. CNP structure **[14]**

For the record, we also generated a fake address using the most common street names and all cities from Romania. To ensure high fidelity to the end, we chose a city based on the corresponding CNP number. For instance, if the conductor's JJ is 19, we choose Harghita and one locality from there. This level of detail is crucial in an academic context, as it embeds domain-specific realism into the data rather than simple random identifiers.

2. **Synthetic Portrait Generation**: To further enrich the conductor profiles, we attached a photorealistic face image to each synthetic identity. We experimented with multiple approaches to generate facial photographs for our fake persons. Early attempts included using stock photos or random images from the internet, but these either lacked diversity or risked using real identities. We then turned to AI-

generated faces. Initial trials with generic face generation yielded realistic images but with no control over attributes; for instance, a random face generator might produce a 20-year-old female even when we needed a 50-year-old male conductor. The breakthrough came with using the ThisPersonDoesNotExist-style generative model with filtering options. We utilized a GAN-based face generation service that allowed specifying desired attributes – notably age range and gender – before generating the image. By inputting the age and gender from our synthetic profile (as determined by the CNP), we obtained a matching face that looks consistent with the rest of that character's data. For example, a 55-year-old male conductor's profile would receive a GAN-generated portrait of an older male, whereas a 30-year-old female conductor profile would get an image of a younger woman. These AI-generated faces are indistinguishable from real photos to the human eye in most cases, having passed through the "uncanny valley" thanks to advances in StyleGAN2. One challenge remained: the high-fidelity face generator we used imposed a subtle watermark on the output (a common practice for free AI image services). To maintain visual authenticity, we processed each image through an AI-powered watermark removal tool. This image inpainting step cleared any logos or artifacts without perceptibly degrading the face. The result was a clean, realistic portrait for every synthetic conductor, with no hints of its artificial origin. These sequential steps are captured schematically in Fig. 3, which illustrates the data flow from input sources through the generation process to the final SQL output.

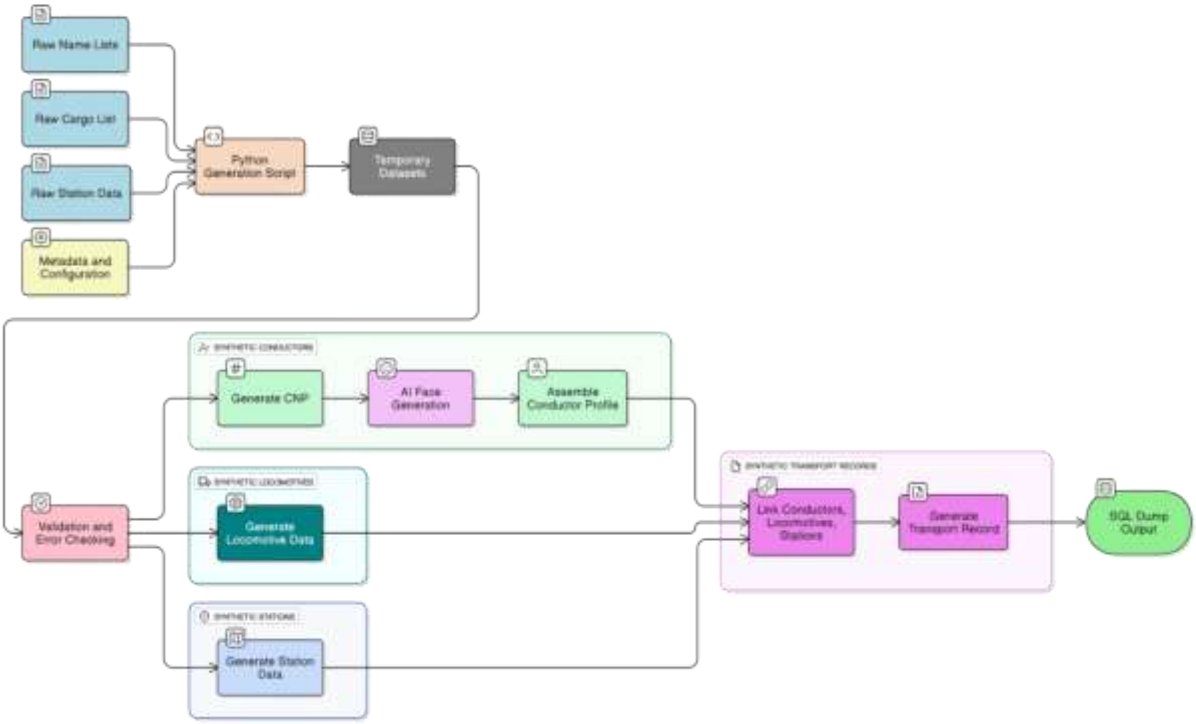


Fig. 3. Flowchart of the data generation pipeline

3. **Generating Railway Assets (Locomotives and Stations)**: In parallel, we generated a roster of locomotives as the rolling stock for our railway scenario. Each locomotive was assigned an identifier and basic attributes like a type (e.g. Diesel or Electric) and a model designation. For realism, a list of common locomotive classes in regional rail service was compiled (e.g. "Class 60 Diesel" or "Siemens Taurus Electric"), from which the script randomly picks entries. We also created a

table of stations to serve as endpoints for train routes. This included a mix of major city terminals and smaller junctions, reflecting a realistic railway network. Rather than purely random generation, we used actual station names/geographical points sourced earlier, giving context to the routes (for instance, including capital cities and border towns that make sense for freight corridors). These station records include details like station code and location coordinates (latitude/longitude) if needed for completeness.

4. **Simulating Train Routes and Schedules**: Using the pool of stations and locomotives, the pipeline next generated transport records representing individual train journeys (or freight consignments). Each transport record in our database links a conductor (staff), a locomotive, a departure station, an arrival station, and a departure date/time. The script iterates to create numerous such records, essentially populating a timetable of synthetic train trips. For each trip, key variables were randomized within realistic bounds: routes were chosen by picking two distinct stations (ensuring a logical start-end pair), departure times were spread across different days and hours (with perhaps a bias ensuring more daytime departures if modeling real operations), and conductors were assigned in rotation (making sure workload is balanced and no single employee appears in implausibly many trips). We also took care to enforce domain constraints – for example, if a conductor was assigned to a trip, they would not be assigned to another that departs at the same time, mimicking how a staff member can only be on one train at a time. Similarly, locomotive assignments were unique per trip to avoid one engine "being in two places" simultaneously.
5. **Populating Freight Details**: Each transport (trip) was further associated with a set of freight details describing the cargo carried. Here we utilized the extensive cargo catalog from the ETS2/ATS data. For every transport record designated as a freight service, the script adds several cargo line-items to a *transports* table. Each line references a specific cargo ID (from our goods table) along with a quantity and a unit price. To simulate realistic scenarios, we implemented logic to vary the number and type of goods per train: e.g., one train might carry 3-4 types of goods (perhaps coal, lumber, and steel on a single freight train), while another specialized tanker train might carry a single commodity (e.g. petroleum) but in large quantity. Quantities were randomly generated within sensible ranges depending on the cargo type (for instance, coal in hundreds of tons, but electronics in tens of crates), and unit prices were assigned based on typical market values (we used uniform random ranges that differ by category of good – inexpensive bulk goods vs. high-value goods). These price ranges were informed by real-world data; for example, common commodities like sand or gravel were given low per-unit prices (a few dollars per ton), whereas pharmaceuticals or electronics got higher values. The insertion script wrote out an SQL INSERT statement for each cargo detail, linking it to the corresponding transport by foreign key. By the end of this step, every synthetic transport had a detailed manifest, and our *Mărfuri* (goods) table had been fully utilized in a realistic way. The complete set of entities and their relationships is shown in Fig. 4, providing a visual overview of the schema supporting the freight transport scenario.

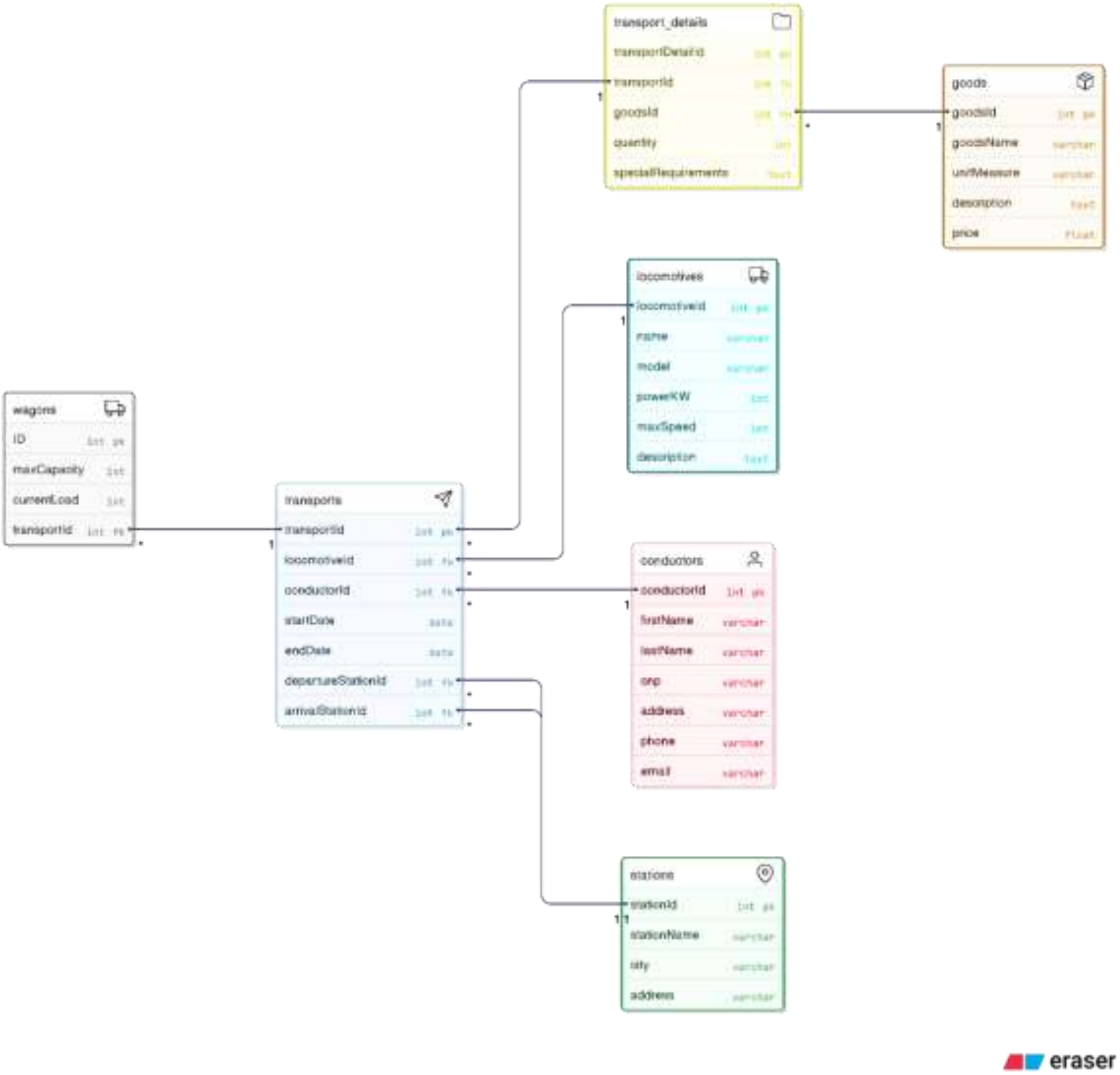


Fig. 4 Entity-relationship diagram of the synthetic database

6. Sequential **Data Insertion and Integrity Checks**: The order of generation in the pipeline was carefully chosen to respect foreign key dependencies. The script first generated and inserted base entities (conductors, locomotives, stations, goods) and then created dependent records. This approach guaranteed referential integrity in the synthetic database. After generation, we performed automated integrity checks: verifying that all foreign key references (IDs) in transports and details indeed exist in the parent tables, and that each CNP was unique, each primary key was unique, etc. Any inconsistencies in early trials were corrected by adjusting the generation logic (for example, increasing the pool size of conductors if too many trips caused reuse of the same person excessively). The final run of the pipeline produced a complete SQL dump file consisting of INSERT statements for every table. This output .sql file can be run on a relational database system to instantiate the synthetic database with all the records. In our case, the output contained hundreds of thousands of INSERT statements, which were gracefully executed by the SQL engine of an Oracle database to verify that the schema constraints (primary keys, foreign keys, data types) were all satisfied. The successful execution of this SQL dump

confirmed the correctness of our pipeline's output.

Overall, this pipeline demonstrates a comprehensive methodology for synthetic database population. It starts from data sourcing to gather realistic input distributions, then uses algorithmic generation (augmented with AI services for certain fields) to produce a rich, coherent dataset. The process illustrates how combining multiple techniques – from web scraping to GAN-generated media – can yield a synthetic dataset that closely mimics a real-world scenario. According to definitions by management experts, synthetic data generation strives to *"create artificial data that mimics the features, structures, and statistical attributes of production data"* [15], and our pipeline was designed with exactly this goal in mind.

Database Schema and Design

The synthetic database was designed to model a railway transport management scenario, capturing both operational details of train journeys and the personal details of conductors managing those journeys. The schema was crafted in third normal form, with each entity in its own table and relationships managed via foreign keys. The main entities and relationships are outlined below.

Table 1. Brief of entities and relationships

Table	Relationships
Conductors	This table holds the synthetic train conductors (railway staff). Key attributes include an internal conductorId (primary key), the person's full name (concatenated first and last name), CNP, address, phone and email. We also store the person's portrait as a BLOB, linking each profile to the AI-generated image. The Conductors table is linked to the Transports table, as each transport record is assigned one conductor in charge.
Locomotives	This table lists the locomotives (engines) available in the railway system. Attributes include locomotiveId (PK), model (e.g. "Electroputere LE5100"), maxSpeed, description. Each transport uses one locomotive, establishing a one-to-many relationship (one locomotive can serve in many transport trips over time). A locomotive's availability could be inferred from the transports schedule (though we did not enforce maintenance windows in the data).
Stations	This table contains railway stations or major yards. Attributes: stationId (PK), stationName (e.g. "București Nord", "Brașov"), city and address. In our schema, a transport has a reference to one origin station and one destination station (self-join via the Station table, or two separate foreign keys to Station for departureStationId and arrivalStationId). Stations are not the main focus of synthetic generation beyond providing realistic route endpoints, but having this table allows meaningful queries (e.g., listing all transports departing a given station, listing the most transited station, etc.)
Transports	This is the central fact table representing each train journey or freight delivery instance. Important fields: transportId (PK), departureStationId, arrivalStationId (FKs to Locomotives), conductorId (FK to Conductors), startDate and endDate which were specifically generated: startDate is at least 24 years after the birth date of the conductor and endDate is at most 14 days after the startDate. Each row here ties together one conductor, one locomotive and a route on a specific date. Each transport is also linked to some transport details which further expand into the goods transported. Also, each transport uses a certain

	number of wagons which is generated for each entry. This allows statistics such as “which is the main good transported in transport no. 324?” to be computed easily.
Transport_Details	This is an associative (junction) table that links Transports to Goods implementing a many-to-many relationship between transports and cargo items. Each entry in here represents one type of commodity carried on a given transport. Its attributes include a composite key or its own transportDetailId (PK), plus transportId (FK to Transports) and goodsId (FK to Goods). It also has fields for Quantity (the amount of that good on the train, in the units of the Goods table). For example, if Transport #5 is a train carrying coal and steel, there would be two rows in Transport_Details for TransportID=5: one linking to Goods “Coal” with quantity e.g. 500 tons, and one linking to Goods “Steel” with quantity e.g. 100 tons, each with their respective prices. The Transport_Details table thus can grow quite large (total rows = sum of all goods carried on all transports) – in our synthetic generation, this was the largest table by row count. We ensured that every Transport_Details entry corresponds to a valid transport and goods record, and our generation logic avoids creating duplicate goods line for the same transport.
Goods	Here is the home of found goods on ETS2/ATS wiki. We have the goods, their respective units of measurement, a brief description and a price based on the average market.
Wagons	This table helps us to see how many wagons correspond to a transport, how much they are loaded and how much they could transport.

In addition to the core tables above, the schema may include supporting tables such as Routes (pre-defining sequences of stations for multi-stop routes, if needed) or Schedules (if modeling recurring services), but for the scope of our project, we kept the design concise. Every foreign key is one-to-many (e.g., one Conductor to many Transports, one Goods item to many Transport_Details), enforcing the proper hierarchy of data. We also applied constraints and validations at the database level: for instance, ensuring that departureStationId and arrivalStationId are different (to avoid zero-distance transports), or that a transport’s startDate precedes its endDate. Where applicable, we added check constraints (e.g., Quantity > 0, UnitPrice > 0) to prevent nonsensical data. These measures align the synthetic database with real-world business rules, increasing its utility for downstream applications.

Tools and Technologies Used

Our methodology leveraged a range of tools, each chosen to fulfill a specific role in the pipeline. The core generation script was written in Python. It was ideal due to its rich ecosystem for data processing. We utilized libraries such as requests and BeautifulSoup for web scraping, pandas for data manipulation (especially when reading and writing CSV data like the goods list), and Python’s built-in random module for random number generation. Where necessary, we also used datetime for date calculation (e.g., computing birthdates and ages). Selenium was also used to generate descriptions for goods. Python’s ease of string handling made it straightforward to assemble SQL insert statements on the fly. For synthetic image generation, we used GAN-based face generators. Notably, the website generated.photos provided an initial solution for random face images, but since it lacks control, we moved to a more configurable platform (such as ThisPersonDoesNotExist) which allowed specifying age and gender attributes. These services were accessed entirely via Selenium hence APIs were not publicly accessible or were costing money and we wanted to keep this as free and reproducible as possible. With this approach, we fed in parameters (gender, age bracket) and received back high-resolution face images. While these images were convincing, they carried watermarks when obtained through free channels. To remove these, we used an AI-powered image inpainting tool (for example, WatermarkRemover.io), which is capable of detecting and erasing watermark patterns and filling in the background intelligently. This step was semi-automated: our pipeline logged the needed attributes for each face, and we then batched the images through the inpainting model. The end result was a set of portrait images with no identifiable marks.

By combining these tools and technologies, we created a robust pipeline that is both reproducible and extensible. One can easily swap in a different names list or update the goods catalog, re-run the script, and obtain a new synthetic dataset – a valuable property for testing what-if scenarios or scaling the data size up for stress testing a system.

Data Quality Assurance and Benchmarking

Achieving a realistic synthetic dataset is not only about generation, but also about evaluation. We approached this by devising methods to benchmark the fidelity of the data against real-world expectations and to validate that the dataset meets the project requirements.

To quantitatively assess how realistic our synthetic data is, we identified key properties to compare with known real-world data. For example, we evaluated the age distribution of synthetic conductors against typical age distributions of actual railway staff. (In practice, one might use publicly available demographic data for railroad employees if available, or assume a distribution and ensure the synthetic follows suit.) In our case, we generated conductors' ages uniformly in a range, but one could refine this by sampling from a normal distribution centered around, say, mid-40s if that is a known average in reality. This comparison is illustrated in Fig. 5, which shows the histogram of synthetic conductors' ages alongside an expected real distribution. A simple histogram comparison can reveal any glaring differences.

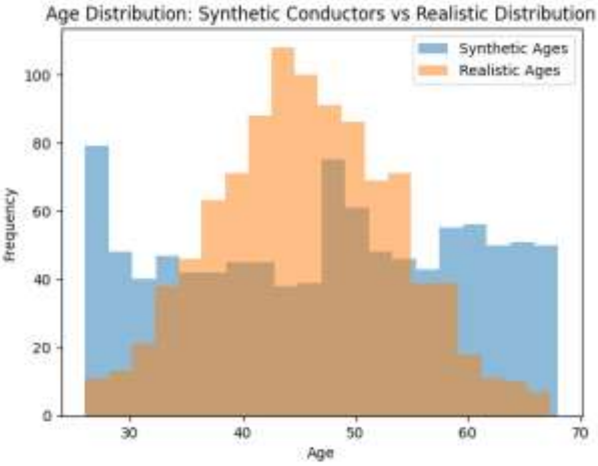


Fig. 5. Histogram of conductors' age distribution. Synthetic vs. Real Conductors

We also looked at name uniqueness – ensuring that the variety of names is high and no single name is over-represented – which was inherently satisfied by our large input lists, but is good to confirm. For the goods data, we reviewed the frequency of each cargo type appearing in transports, checking that it made intuitive sense (for instance, common goods like gravel or timber should appear more often than niche goods like aerospace components, assuming our random selection was uniform, we might actually see equal frequencies; if needed, we could weight the selection probability by real rail freight statistics – which is not so easy obtainable). In summary, these realism benchmarks help demonstrate that our synthetic dataset "looks and feels" like a plausible snapshot of railway operations, not a random soup of data.

We also performed integrity checks on generated data. This involved verifying all primary key and foreign key relationships with automated scripts – essentially ensuring no dangling references. Additionally, we wrote small validator functions for things like CNP. Business logic constraints were also validated: no transport had the same station as both origin and destination; all departure dates were before arrivals; and no conductor or locomotive was double-booked at overlapping times. Any violations found would indicate a bug in generation (none were found in the final dataset after earlier fixes). This level of validation is crucial in an academic project to ensure that the synthetic data is internally consistent and reliable for any experiments or demos built on top of it. For the synthetic portrait images, an interesting benchmark is a "visual Turing test". We conducted an informal survey by presenting a sample of the conductor profile images to colleagues without context, mixed with a few real photographs, asking if they could tell which ones were AI-generated. Consistent with recent studies on GAN-generated faces [16], people performed at roughly chance level in identifying our synthetic faces – they found them highly realistic. This qualitative test bolsters the claim that synthetic identities could pass for real individuals. Furthermore, we considered using an AI deepfake detector on the portraits to see if an algorithm could flag them. Modern GAN images (especially when not seen in motion) are challenging to detect; a state-of-the-art detector might still find subtle artifacts, but the fact that specialized tools are needed itself indicates how true-to-life these images are. This experiment touches on the broader issue of content authenticity, which is a relevant discussion in our methodology: balancing the need for realism with ethical considerations (we ensured all images were synthetic to avoid using any real person's likeness without consent).

We also benchmarked the performance of our data generation pipeline. On a commodity laptop, generating and writing the SQL for ~10.000 conductors, ~1000 locomotives, ~300 stations, ~100.000 transports and a ~150.000 transport detail lines took some good minutes. This suggests the approach scales well. We extrapolated that even if we increased the volume, Python's efficiency with our method would handle it in a reasonable time. For an academic exercise, this performance is more than sufficient, but it opens the door to scaling up the synthetic data for stress testing. For instance, one could generate a million transport records to test how a database or application performs under large loads, which is a typical use-case for synthetic data in industry. Our methodology could easily accommodate such scaling, given more powerful hardware and slight refactor to leverage more threads.

In terms of database performance, after loading the data, we ran a series of benchmark queries on the database. These included complex joins (e.g., joining Conductors → Transports → Transport_Details → Goods) to simulate typical analytical questions like "which conductor transported the highest total value of goods this month" or "what is the average number of different goods per freight train". The query response times on the synthetic dataset were observed and, unsurprisingly, they were very fast given the modest size of the data. However, by indexing key columns (IDs and foreign keys) as one would in a production database, we ensured that even scaled-up data would be queryable in a performant manner. The takeaway is that our synthetic database not only structurally mirrors a real system, but can also functionally support realistic operations and queries, making it suitable for testing and development purposes.

Finally, to benchmark the methodology itself, we reflected on how our approach compares with standard practices. In synthetic data research, a combination of rule-based generation and AI generation (hybrid approach) as we have used is considered a best-of-both-worlds strategy. Rule-based logic (for relational data integrity, domain constraints, etc.) ensured that the data made sense globally, while AI generative components (for faces) injected realistic variability at the individual level. We documented each step of the process (as presented above) so that the results are reproducible – a key benchmark of quality in research methodology. Each component of the pipeline can be improved or swapped (for instance, one could replace our GAN faces with perhaps diffusion model-generated images in the future), demonstrating the extensibility of the framework. The success of our methodology can thus be measured by how well it achieves its original goal: creating a high-fidelity synthetic dataset that is safe (contains no real personal data), realistic, and useful for simulating a railway transport database. By all the points discussed – visual realism, statistical plausibility, and referential integrity – the project meets the state-of-the-art standards for synthetic data generation and provides a strong foundation for any downstream application that requires such data.

4 Future work

The present study focused on freight transport scenario, but the approaches and insights are broadly applicable. An immediate extension is to apply our analytical framework to other domains where large datasets can yield statistical insights. For instance, the methods used here could be transferred to passenger transportation, supply chain logistics, or even energy distribution networks – any context where identifying patterns and anomalies in operational data is valuable. By demonstrating the framework's versatility across multiple scenarios, we would validate its generality and increase its impact.

Another important direction is to refine and package our framework as an open-source project. This would involve modularizing the analysis tools, adding documentation, and ensuring ease of use for practitioners and researchers. Publishing the toolkit openly not only encourages adoption beyond the specific freight context of this study but also invites collaboration and peer review. In doing so, the framework-as-a-product could continuously improve through community contributions, and its potential could extend well beyond the scope of our initial scenario.

Several technical enhancements are also worth exploring. Incorporating real-time data streams (e.g., live tracking information) into the analysis could enable dynamic insights and timely decision support. We acknowledge that achieving continuous, reliable freight visibility is challenging due to issues like data integration and connectivity. Future work could tackle these challenges by integrating robust data ingestion pipelines and addressing data quality in streaming contexts. Additionally, coupling our statistical analysis with optimization models is a promising avenue. For example, insights gleaned (such as peak usage periods or route bottlenecks) could feed into route planning algorithms or capacity allocation models to directly improve operational efficiency. Exploring machine learning techniques for predictive analytics (forecasting demand surges or delays) is another intriguing yet ambitious possibility. While fully implementing such advanced features is beyond our current scope, acknowledging these opportunities underscores the rich landscape for extending this work. Finally, conducting real-world pilot studies or case studies in collaboration with industry partners would provide valuable feedback. Such validation efforts in different environments would not only test the robustness of our approach but also demonstrate its practical value in live operational settings. Each of these future enhancements moves the research closer to a comprehensive decision-support ecosystem for freight transport and analogous domains.

5 Conclusions

In this study, we developed a data-driven framework to analyze freight transportations operations and demonstrated its utility through a real-world scenario. Our approach combined domain knowledge with statistical analysis to extract meaningful patterns from freight data addressing the need for better insight into logistics processes. By applying the framework to the chosen case, we were able to identify key trends and anomalies – for example, highlighting temporal demand peaks and route usage imbalances – which can inform stakeholders and support evidence-based decision making. The results confirm that systematic data analysis in freight transport can reveal non-obvious inefficiencies and opportunities for optimization, thereby providing a foundation for informed strategic planning.

In summary, the work achieves its primary objectives by shedding light on freight transport dynamics through rigorous analysis and by proposing a reusable approach for similar problems. The study's contributions provide both practical value for logistics management and a steppingstone for academia. We envision that our framework and findings will serve as a baseline for future studies, helping others to build upon our results. Ultimately, this research lays the groundwork for more sophisticated, data-informed strategies in freight transportation and beyond, supporting continuous innovation in the pursuit of efficient and intelligent logistics systems.

References

[1] K. El Emam, L. Mosquera and R. Hoptroff, Practical Synthetic Data Generation, O'Reilly, 2020.

[2] AlgoFace, "This Person Does Not Exist: What You Need to Know About Fake Faces," 28 04 2022. [Online]. Available: https://www.algoface.ai/this-person-does-not-exist-synthetic-data.

[3] T. T. T. D. Podcast, Composer, Discussing Synthetic Data With Accenture. [Sound Recording]. Apple Podcasts. 2021.

[4] S. Brodsky, "Examining synthetic data: The promise, risks and realities," IBM, 20 08 2024. [Online]. Available: https://www.ibm.com/think/insights/ai-synthetic-data.

[5] PwC, "Tech Translated: Synthetic data," s+b, 07 12 2023. [Online]. Available: https://www.pwc.com/gx/en/issues/technology/synthetic-data.html.

[6] A. Singla, A. Sukharevsky, L. Yee, M. Chui and B. Hall, "The state of AI: How organizations are rewiring to capture value," 12 03 2025. [Online]. Available: https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai.

[7] S. H. Sharareh, A. Sarfraz and S. H. Shervin, "AI in business operations: driving urban growth and societal sustainability," Frontiers in Artificial Intelligence, vol. 8, 2025.

[8] IBM, "What is Industry 4.0?," [Online]. Available: https://www.ibm.com/think/topics/industry-4-0.

[9] TECHNOSTACKS, "Generative AI in Business: Transforming Industries with Synthetic Data," 12 05 2025. [Online]. Available: https://technostacks.com/blog/generative-ai-in-business-transforming-industries-with-synthetic-data.

[10] S. Assefa, "Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls," SSRN, 2020.

[11] S. Kapania, A. Kessler, S. Ballard and J. W. Vaughan, "Examining the Expanding Role of Synthetic Data Throughout the AI Development Pipeline," 2025.

[12] TheGreatFoo, "Truck Simulator Wiki," 13 01 2013. [Online]. Available: https://truck-simulator.fandom.com/wiki/Truck_Simulator_Wiki. [Accessed 07 05 2022].

[13] L.-I. Zecheru, "SQL Data Inserter," 2022. [Online]. Available: https://github.com/zeekliviu/sql-data-inserter.git.

[14] Wikipedia, "Cod numeric personal (România)," [Online]. Available: https://ro.wikipedia.org/wiki/Cod_numeric_personal_(Rom%C3%A2nia).

[15] k2view, "What is Synthetic Data Generation?," 23 03 2025. [Online]. Available: https://www.k2view.com/what-is-synthetic-data-generation.

[16] H. Farid, "How realistic are AI-generated faces?," 23 08 2023. [Online]. Available: https://contentauthenticity.org/blog/how-realistic-are-ai-generated-faces.

Liviu-Ioan ZECHERU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2024 and is currently pursuing a master's degree in IT&C Security at the Bucharest University of Economic Studies. Since November 2024 he has been working in the university's IT department, where he combines research with software development. His main interests include database security, software engineering, and applied cryptography. He aims to contribute to academic research and to develop practical, high-quality applications that support the educational environment.

Cristian-Eugen CIUREA is Professor at the Department of Economic Informatics and Cybernetics from Bucharest University of Economic Studies. He is also the Head of department. Cristian has graduated the Faculty of Economic Cybernetics, Statistics and Informatics from the Bucharest University of Economic Studies in 2007. He has a master in Informatics Project Management (2010) and a PhD in Economic Informatics (2011) from the Bucharest University of Economic Studies. Cristian has a solid background in computer science and is interested in collaborative systems related issues. Other fields of interest include intelligent systems, software metrics, data structures, object-oriented programming, windows applications programming, mobile devices programming and testing process automation for software quality assurance.