An Integrated Approach Using Ontologies, Knowledge Graphs, Machine Learning, and Rules Models for Synthetic Financial Time Series Generation

Laurentiu Vasiliu[1*], Radu Prodan[2], Ahmet Soylu[3], and Dumitru Roman[3,4]

[1] *Peracton Ltd. DHKN Galway Financial Services Centre, Moneenageisha Rd, Galway, H91 V2R6,*
[2] *Institute of Information Technology, University of Klagenfurt, Universitätsstraße 65-67, A-9020 Klagenfurt am Wörthersee, Austria*
[3] *Kristiania University College, Oslo, Norway*
[4] *SINTEF AS, Forskningsveien 1, 0373 Oslo, Norway*

Abstract

In the Graph-Massivizer EU project, the financial use case is focused on generating synthetic financial time series in extreme volumes (PB) for advanced testing and training of financial (investment and trading) algorithms. Our key approach integrates ontology-based, graph-based, and rule-based models, leveraging the strengths of all three technologies. Ontologies are employed to capture the detailed properties of financial time series data, graph models are used to generate synthetic data, and financial-related rules are applied to ensure the desired quality and statistical properties of the synthetic data.

Keywords
Ontologies, synthetic data, financial time series, machine learning

1. Introduction

Synthetic data refers to artificially generated datasets that are specifically designed to replicate the characteristics of real-world data—in this case, financial time series. It has become a viable solution for powering quantitative analysis and back-testing of financial models, serving as a good alternative to historical data. The demand for synthetic data has increased due to the growing complexity of financial models and algorithms, driven by data-intensive machine learning (ML) models. These models often face limitations with real historical datasets, such as capped volumes, incomplete data, high costs or irrelevance when dealing with much older data. The key advantage of synthetic data lies in its ability to capture the statistical properties of real-world markets while maintaining a completely artificial nature, enabling intensive testing before financial models and algorithms are validated on real-time financial data and with live money.

The Graph-Massivizer (G-M) project [1] is developing a software platform capable of processing extreme volumes of data. One of the project's use cases is focused on generating synthetic data in extreme volumes that closely match the quality and characteristics of historical data samples of stocks and futures commodities. The approach is centered around knowledge graphs (KGs) [3], chosen for their ability to capture, store, and represent historical financial time series. The

RuleML+RR'24: Companion Proceedings of the 8th International Joint Conference on Rules and Reasoning, September 16--22, 2024, Bucharest, Romania
* Corresponding author.
laurentiu.vasiliu@peracton.com (L. Vasiliu); radu.prodan@aau.at (R. Prodan); ahmet.soylu@kristiania.no (A. Soylu) dumitru.roman@sintef.no (D. Roman)
0009-0000-9791-2759 (L. Vasiliu); 0000-0002-8247-5426 (R. Prodan); 0000-0001-6034-4137 (A. Soylu); 0000-0001-6397-3705 (D. Roman)

technologies employed are designed to create, process, store, and generate these KGs (knowledge graphs), which can be further enhanced by using ontologies to represent entities and their relationships.

2. Generating synthetic time series

The financial use case within the G-M project [2] (as described in Figure 1) aims to enhance algorithmic investment and trading performance in green-focused investments, targeting improvements such as a 2-4% increase in performance, a Sharpe ratio above 5, and a 1-2% boost in alpha. This is achieved by utilizing extreme volumes (in petabytes) of synthetic data for testing and training pre-production financial algorithms. The G-M platform enables the creation of realistic and cost-effective synthetic financial datasets, unlimited in size and accessibility, while mitigating issues such as biases, overfitting, and indirect contamination that often accompany historical data testing.

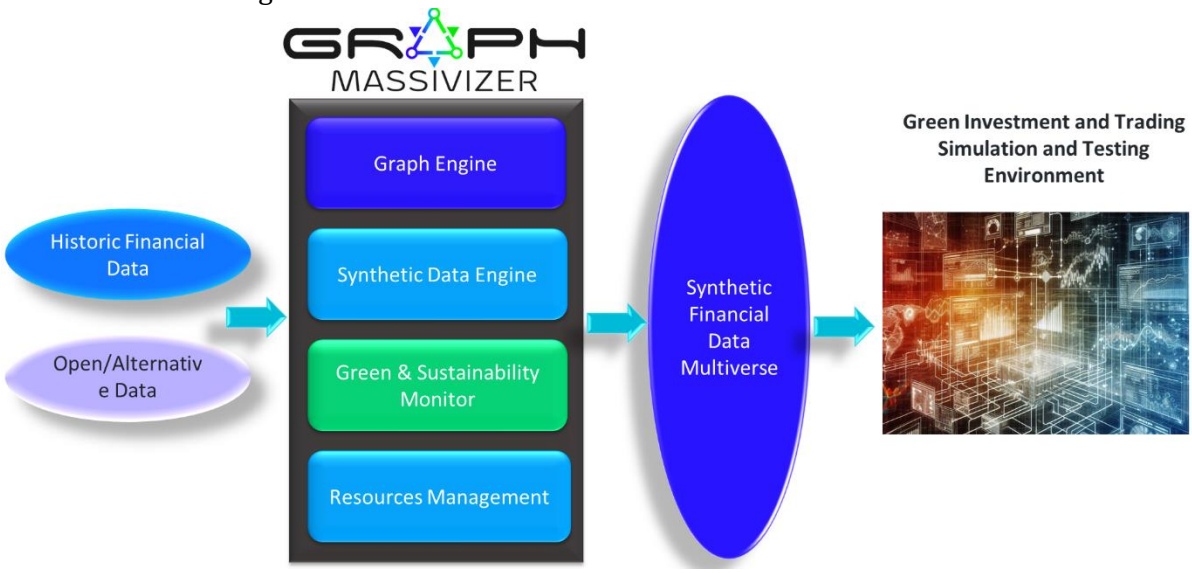

Figure 1: Financial Use Case, Graph-Massivizer EU project [1]

To generate synthetic data at this scale, the G-M approach involves several steps. Initially, batches of historical data (totaling 10 terabytes) from company stocks and futures commodities contracts are mapped to a financial massive graph (F-MG) through a time-series-to-graph transformation. Next, a synthetic financial massive graph (SF-GM) is created using a generative model. Finally, this SF-GM is used to produce synthetic financial data in batches, ready to be used in financial testing and simulations, with a total target output of 1 to 5 petabytes of synthetic data. The G-M Toolkit, currently under development, is an integrated platform comprising five specialized tools: Graph-Inceptor, Graph-Scrutinizer, Graph-Optimizer, Graph-Greenifier, and Graph-Choreographer (Figure 2). These tools perform distinct and critical functions for massive graph processing, including graph creation, analytics and probabilistic analysis, efficient execution of operations, energy consumption evaluation, and serverless deployment for on-demand resource utilization.

3. Challenges

Generating meaningful synthetic financial time series [4] involves several key challenges. First, accurately modeling the original historical financial data is essential. Given the complexities of financial markets, this requires careful consideration of various critical aspects, such as relevant financial variables, data clustering, fat tails, and noise, as well as how relationships between

different data types can be extracted using ontologies and reasoning. Additionally, we must address the heterogeneity of time series data, accounting for their changing statistical properties—such as mean, variance, and covariance—which are needed for calculating risk and performance metrics for the targeted financial assets. Second, ensuring the quality of the synthetic data is very important so it is indistinguishable from the original historical data in terms of its components, statistical properties, values, and patterns. To achieve this, quality must be enforced at the moment of data generation through well-defined rules. In the G-M financial use case, we have chosen to build a proprietary rule engine that encapsulates all relevant quality rules—covering statistical aspects, patterns, and correlations—to ensure that the synthetic data possesses the desired properties from the outset.

4. Proof of concept and implementation

Our focus to date has been on several core aspects such as scalable data generation, efficient data storage and streaming, energy consumption monitoring, parallel processing, large memory capacity management, data security, and cost optimization to support the anticipated production demands on the G-M platform. The current financial use-case proof of concept is built on preliminary implementations of the main tools, as depicted in Figure 2 below, with their respective data flows highlighted. Their first versions have been uploaded to GitHub [6], where 17 repositories—both private and public—are available for access.

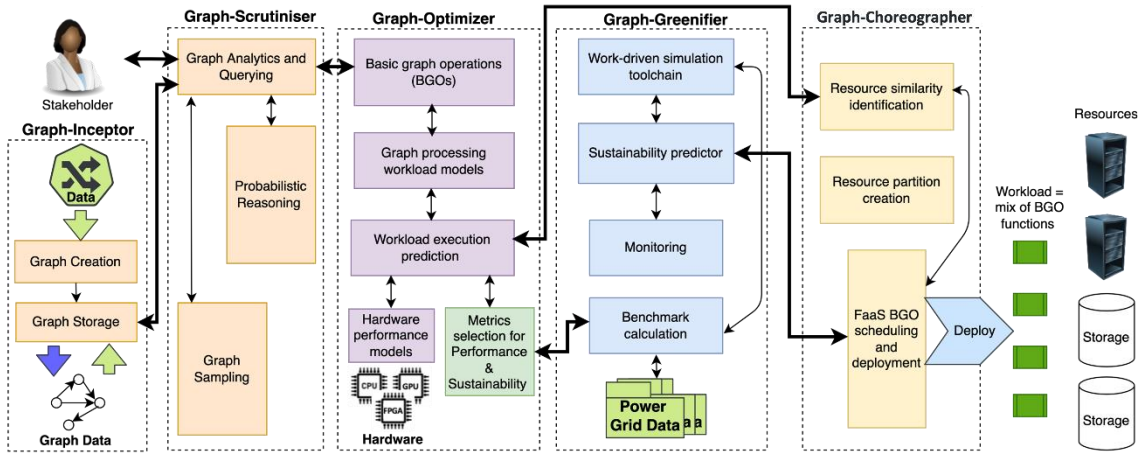

Figure 2: Graph-Massivizer Platform [7]

The Graph-Massivizer platform pipeline [7] involves five sequential steps, each corresponding to the invocation of a specific tool that has very specific functionalities:

1. **Graph-Inceptor**: This is the first tool in the pipeline, responsible for the historic financial data ingestion, initializing and managing the ingestion and storage of massive graphs. It supports three operations namely [7]:
    1. **Graph creation** – implement BGOs to support the ETL process of extraction, transformation and loading data.
    2. **Graph modelling** – administers graph generators, ontologies, and mapping rules for the financial use-case dataset.
    3. **Graph storage –** allows access to the storage layer through a virtual KG.
2. **Graph-Scrutinizer**: Utilizes the ingested graph data for in-depth analysis and reasoning. It is comprised of [7]:
    1. **Graph analytics –** focusing on higher-level operations acting on batch and streaming data.

2. **Graph algorithms and querying** – is concerned with BGOs used by the data scientist or the graph analytics group comprising both exact and approximated implementations.
3. **Graph distillation –** has the building blocks that prepare indices belonging to the massive graph to be used by the approximate graph algorithms and querying BGOs.

3. **Graph-Optimizer**: Maps BGOs (Basic Graph Operations) to the target computing units, while considering their properties and metrics collected from the massive graph. It has the following elements [7]:
    1. **System model** – is concerned with the composition of hardware models, computational units, and interconnections.
    2. **Data model** – it is coming from data reduction over the original dataset allowing for an accurate estimation of its processing properties.
    3. **Workload model** – it is composed of multiple BGOs, captures the actual graph processing metrics and properties.
    4. **Design-space exploration** – combines the above models, uses performance modelling to predict different configurations and iterates them to find the best solution.
4. **Graph-Greenifier**: Analyzes graph data and the processing metrics gathered by Graph-Optimizer to optimize both performance and sustainability. It has the following elements [7]:
    1. **Workload-driven simulation toolchain** - for modelling the impact of graph processing.
    2. **Sustainability predictor -** is ranking graph processing scenarios based on performance, energy efficiency and sustainability at scale.
    3. **Monitoring -** of the relevant sustainability metrics.
    4. **Sustainability benchmark -** provides run-time energy labels including information on energy sourced derived from data centers.
    5. **Power grid data interface** - automates data gathering based on the electrical energy offer and price, energy sources and greenness.
5. **Graph-Choreographer**: Integrates the graph processing with various infrastructures and workflows. It has the following elements [7]:
    1. **Monitoring** – continuously checks the lifecycle of incoming and outgoing events and logs the observations.
    2. **Graph profiling** – obtains essential information about raw input graphs, sampled graphs, and analyzed data. Then categorizes the graph BGOs based on the UC needs, time and resources required.
    3. **Resource profiling and partitioning** - handles and categorizes the monitored data from nodes, processing cores, memory, storage, network bandwidth, deployed functions also handle resources' diversity and network structures.
    4. **Function scheduling and provisioning -** a heuristic scheduling model defines constraints considering resource utilization and cloud, fog and edge resource limitations and provisions the nodes to the BGOs to optimize objective metrics; then a suitable node is identified to deploy the new BGOs.
    5. **Sustainability analysis –** verifies the sustainability evaluation criteria and instructs the function scheduling and execution engine to make the appropriate decisions.

6. **Execution engine** – deploy and run the BGOs' and their libraries on the computing nodes.

Each tool logs its operations internally, providing the toolkit with general inspection, development, and debugging capabilities. For generating synthetic financial data, the complete process flow to generate it was divided and implemented into three partial ones:

1. Historic data ingestion (using Graph-Inceptor tool)
2. Synthetic graph generation (using Graph-Scrutinizer and Optimizer tools)
3. Synthetic graph-to-time-series generation (using Graph Optimizer and ts2g2 library [6])

This phased approach was chosen for rapid prototyping, concentrating on the essential tools relevant for this use case: Graph-Inceptor, Graph-Scrutinizer, and Graph-Optimizer. While these are partial functionalities, testing them independently first is an essential step towards a full integrated synthetic data generation process, The next phase will include the integration of all tools, including Graph-Greenifier to measure energy consumption during synthetic data generation and Graph-Choreographer to ensure synchronized operation of all components.

5. Future work

The Graph-Massivizer EU project is currently at the midpoint of its development, progressing steadily towards completion. We have reached a stage where a first proof-of-concept for generating synthetic time series data in extreme volumes is currently being demonstrated. While the core concepts and approaches have been identified and are in the prototyping phase, the focus of our future work will shift to the implementation, integration and debugging of the G-M tools. This will include a continuous effort to enhance the quality of the synthetic data until it becomes indistinguishable from historical data samples. The synthetic data generated will be utilized to test green investment algorithms within Peracton's back testing engine, where their performance will be compared to their behavior when using historical financial data. The results from these tests will provide valuable feedback to the G-M platform, guiding further improvements and consolidation of the tools and methodologies.

Acknowledgements

This project has received funding from the European Union's Horizon Research and Innovation Actions under Grant Agreement Nº 101093202 [1].

References

[1] E. U. H. R. Graph-Massivizer EU Project, G. A. N. . Innovation Actions, Graph-Massivizer, 2023. URL: https://graph-massivizer.eu/.

[2] E. U. H. R. Graph-Massivizer EU Project, G. A. N. . Innovation Actions, Use case 1 green-finance, 2023. URL: https://graph-massivizer.eu/project/green-and-sustainable-finance/.

[3] N. Kertkeidkachorn, R. Nararatwong, Z. Xu, R. Ichise, Finkg: A core financial knowledge graph for financial analysis, in: 2023 IEEE 17th International Conference on Semantic Computing (ICSC), IEEE, 2023, pp. 90–93.

[4] M. Dogariu, L.-D. Ștefan, B. A. Boteanu, C. Lamba, B. Kim, B. Ionescu, Generation of realistic synthetic financial time-series, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 18 (2022) 1–27.

[5] Peracton Ltd. Website, 2024, URL: https://peracton.com.

[6] Graph-Massivizer Github repositories, 2024, URL: https://github.com/orgs/graph-massivizer/repositories.

[7] E. U. H. R. Graph-Massivizer EU Project, G. A. N. . Innovation Actions, D2.1 'Graph-Massivizer Requirements, Elicitation and First Architecture Design', July, 2023.