Modeling and Generating Extreme Volumes of Financial Synthetic Time-Series Data with Knowledge Graphs

Laurentiu Vasiliu[1], S. Haleh S. Dizaji[2], Aaron Eberhart[3], Dumitru Roman[4] and Radu Prodan[2]

[1]*Peracton Ltd. DHKN Galway Financial Services Centre, Moneenageisha Rd, Galway, H91 V2R6, Ireland*
[2]*Institute of Information Technology, University of Klagenfurt, Universitätsstraße 65-67, A-9020 Klagenfurt am Wörthersee, Austria*
[3]*metaphacts GmbH, 36 Daimlerstraße, Walldorf, 69190, Germany*
[4]*SINTEF AS, Forskningsveien 1, 0373 Oslo, Norway*

Abstract

This paper outlines the approach and technology employed to model and generate extreme volumes of synthetic financial time-series data. We introduce the Graph-Massivizer project and its financial use case, focusing on green sustainable finance. One project objective is to create synthetic financial data in extreme volumes to facilitate advanced testing and simulations of investment and trading algorithms. Afterward, we provide an overview of the methodology, detailing the utilization of ontologies and knowledge graphs. Furthermore, we elaborate on modeling correlations between different markets' time-series and how we can benefit in combination with graph neural network models to generate financial data. We then present the current implementation status and conclude with a discussion of future work.

Keywords

Knowledge graphs, ontologies, synthetic data, financial time-series, extreme data, machine learning, correlation analysis, pattern recognition

1. Introduction

In the financial investment and trading domains, synthetic data—artificially generated datasets that mimic real-world financial time-series characteristics—has become a robust solution for quantitative analysis and back-testing. The demand for synthetic data has arisen due to increasingly complex financial models and algorithms driven by data-demanding machine learning (ML) models. These models find real historical data time-series to have multiple limitations, such as reduced volumes, high costs, incomplete data, or irrelevance as we go further back in time. The core characteristic of synthetic data is its ability to capture the statistical properties of real-world markets while maintaining a completely artificial nature. This allows for intensive testing before financial models and algorithms are further validated on real-time financial data. The Graph-Massivizer project [1] aims to develop a software platform consisting of independent yet integrated tools. In one of its use cases, this platform will generate synthetic data in extreme volumes, closely matching the quality and characteristics of historical data samples of stocks and commodities futures, with plans to expand to other securities such as ETFs, bonds, and options. At the core of the approach are knowledge graphs (KGs), chosen for their ability to capture, store, and represent historical financial time-series. All technologies used are designed around creating, processing, storing, and generating these KGs. KGs, designed to represent entities and their relationships utilizing ontologies, can be significantly enhanced. Firstly, ontologies provide a shared vocabulary and semantic alignment, particularly useful when integrating data from different sources across various KGs. Secondly, KGs can leverage ontologies to perform inference and reasoning, allowing the discovery of new relationships within the graph. Thirdly, ontologies can enrich

Woodstock'22: Symposium on the irreproducible science, June 07–11, 2022, Woodstock, NY
*Corresponding author.
laurentiu.vasiliu@peracton.com (L. Vasiliu); Seyedehhaleh.Seyeddizaji@aau.at (S. H. S. Dizaji); ae@metaphacts.com (A. Eberhart); Dumitru.Roman@sintef.no (D. Roman); radu.prodan@aau.at (R. Prodan)
0009-0000-9791-2759 (L. Vasiliu); 0000-0002-5886-9636 (S. H. S. Dizaji); 0000-0003-3007-5460 (A. Eberhart); 0000-0001-6397-3705 (D. Roman); 0000-0002-8247-5426 (R. Prodan)

KGs by adding missing information, such as properties or classes that are not explicitly present but are implied based on existing relationships.

2. Graph-Massivizer Project - The Financial Use Case

Green and sustainable finance This use case [2] aims to enhance algorithmic investment and trading capabilities in green-focused products and investment/trading styles by generating and utilizing extreme volumes of synthetic data for testing and training. In this respect, the Graph-Massivizer project seeks to overcome the limitations posed by financial market data providers—such as restricted data volume, reduced accessibility, and high costs, by enabling the rapid, semi-automated creation of realistic and affordable synthetic financial datasets that are unlimited in size and accessibility. It also aims to improve ML-based green investment and trading simulations, eliminating critical biases such as prior knowledge, over-fitting, and indirect contamination due to current data scarcity. The approach first maps samples of historical financial data (stocks and commodities futures) to a massive graph (F-MG) through a time-series to graph transformation. Next, using a generative model, we create a synthetic financial massive graph (SF-MG). Finally, we generate synthetic financial data from the SF-MG by enforcing specific quality rules. To achieve this, the Graph-Massivizer platform is provided with 10 TB of historical data samples, with the primary goal (KPI 1) of generating between 1 and 5 PB of synthetic financial time-series data. Another goal (KPI 2) is to achieve 90% energy consumption accountability for synthetic data creation. We use this data to test and improve financial algorithms, and aim to achieve (KPI 3) a measurable return increase of 2-4% in the enhanced financial algorithms that use synthetic data. Additionally, we aim to achieve (KPI 4) an increase in the financial algorithms' alpha by 1-2% and a Sharpe ratio greater than 1.5.

Graph-Massivizer toolkit The Graph-Massivizer toolkit is an integrated platform composed of five tools (Graph-Inceptor, Graph-Scrutinizer, Graph-Optimizer, Graph-Greenifier, and Graph-Choreographer) that perform specific and unique functions for massive graph processing:

- Graph-Inceptor: realizes a massive graph for the system to use.
- Graph-Scrutinizer: provides analytic capabilities and probabilistic reasoning for insights.
- Graph-Optimizer: ensures that large graph operations are completed efficiently.
- Graph-Greenifier: evaluates the energy consumption of massive graph operation.
- Graph-Choreographer: allows serverless deployment to use resources on-demand.

Further on, Figure 1 shows the overall Graph-Massivizer architecture and how the five tools are interconnected. Additionally, we can see in this diagram the external components, such as the metaphactory platform, the graph database, and the hardware and infrastructure used by the toolkit.

3. Challenges in Modeling Financial Data

Modeling financial data presents several challenges due to financial markets' dynamic nature and complexities. In addition to numerous variables, volatility clustering, fat tails, and noise, which are all specific to financial data, we focus particularly on five aspects to generate synthetic data.

KG relations' extraction and enrichment In large-scale financial data, extracting relationships among various data types can be complex and non-intuitive, often requiring inference methods to identify them. We aim to enhance the quality of KG relation extraction by utilizing ontologies and reasoning methods to identify and extract non-obvious relationships.

Heterogeneous time-series data Financial data consists of different types of time-series data with different semantics, domains, and dynamics. We can mitigate this diversity by using ontologies underlying their relationships and finding correlations among them.

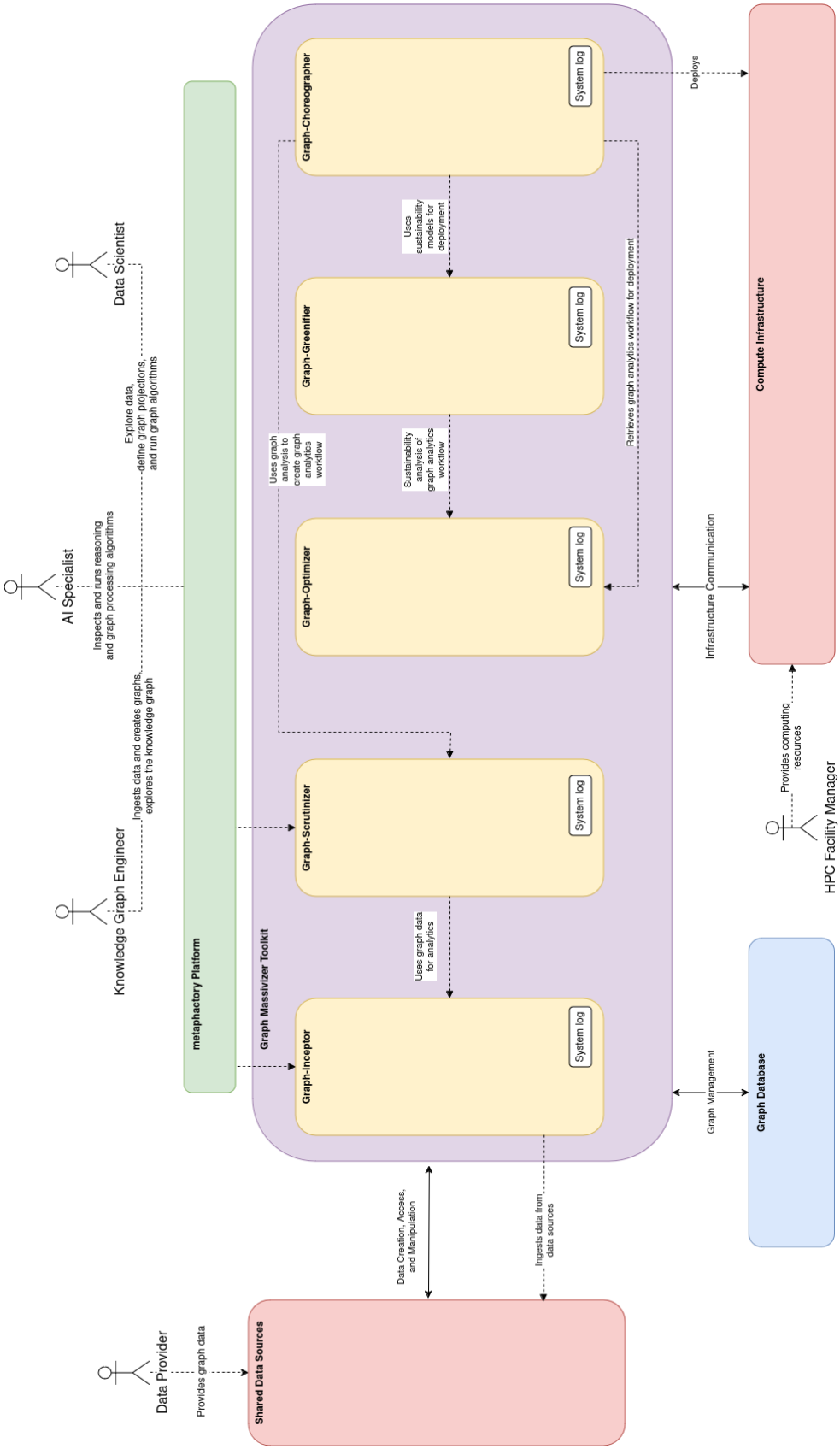

Figure 1: Graph-Massivizer architecture.

Changing statistical properties Financial time-series often display changing statistical properties over time, such as means, variances, and covariances. These properties are used for measuring an asset's

performance and risk. These statistical patterns must be identified and replicated in the synthetic data to model the original data accurately.

Quality assessment of generated data The evaluation of synthetic data is ongoing research [3] and [4] and [5] review several evaluation methods of financial and other synthetic time-series. The method of [3] applies various qualitative, quantitative, and predictive methods. These methods consist of statistical models and distances, such as various distribution divergence metrics, the Kolmogrov-Smirnov test, real and synthetic data correlation analysis, and the non-parametric model of MMD. Other methods evaluate the ML models on real data trained on synthetic data. Additionally, we can evaluate specific quantities such as Value at Risk (VAR). These methods differ depending on the use case, and we aim to select appropriate metrics.

4. Using Ontologies and Knowledge Graphs

KGs and ontologies allow scientists and domain experts to model complex relations between data in a logically structured and machine-readable format. This capability allows ontologies to connect diverse sources of information, such as the use case presented here and similar related data.

In the Graph-Massivizer project, ontologies represent and integrate data from diverse use cases. The metaphactory platform was chosen to manage ontologies and integrate data with a front-end interface. Metaphactory has many applications for developing and managing ontologies, KGs, and other related semantic artifacts [6]. With metaphactory, users can interact with and create ontologies and integrate and use data that aligns with the ontology.

By decoupling the data and the schema for the data, the ontology allows developers to model and prepare for handling massive amounts of data in an abstract way. For instance, a user or developer can write queries to inspect only the relevant data of interest inside the large data set. While queries like this are not themselves a direct algorithmic optimization, they do play a critical role in ensuring scalability is possible by identifying critical semantic information and metadata that can reduce a huge chunk of data into something more tractable.

In this section, we will describe the data represented by ontology and then show the ontology that schematizes it to integrate it with a KG.

4.1. Ontology data

This use case initially focuses on two types of financial products: stocks and commodities futures. However, this paper will concentrate on one financial product: stocks. The financial ontology for stocks consists of four main types of financial data:

Fundamental data Fundamental data [7] indicators represent accounting data related to a company and its particular industry. These indicators have a low update frequency (quarterly on average or yearly) [8] and provide long-term insights into a company's valuation and price evolution.

Technical data In contrast with fundamental data, technical data [9] has a very high update frequency (tick/second/minute, etc.), offering short-term insights into stock price movements. This data includes fine-grained historical stock price information in the form of Open, High, Low, Close, and Volume (OHLCV). Numerous additional statistics can be derived from this financial modeling and prediction data, particularly for intra-day trading.

ESG data Environmental, social, governance (ESG) [10] data measures companies based on various responsibility metrics, including environmental, social, and governance criteria. By considering these criteria in their investments, investors encourage responsible corporate behavior and avoid investing in companies with risky or unethical practices.

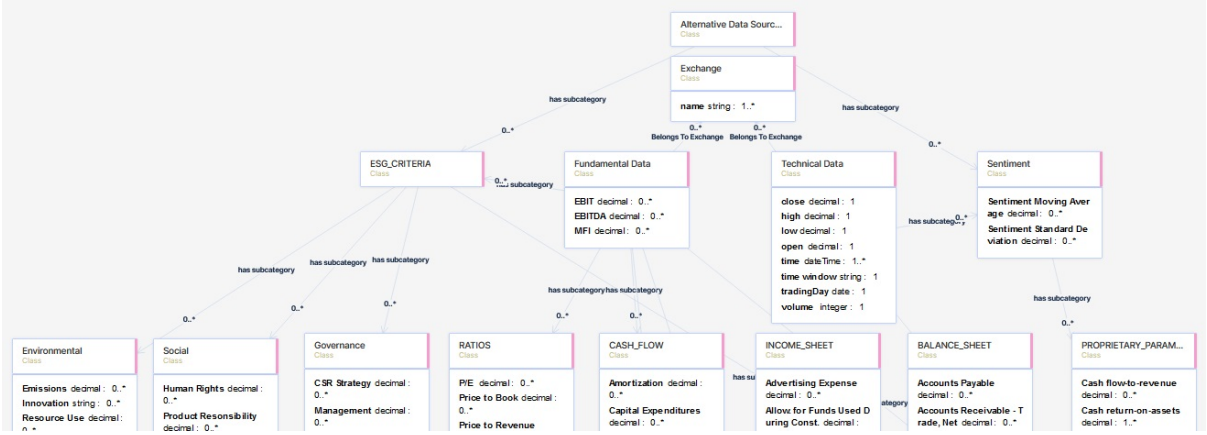

Figure 2: Financial ontology diagram.

Sentiment data Market sentiment data [11] reflects investors' attitudes toward a company, sector, or financial market. Various indicators derived from statistical technical analysis, social media, or alternative data sources can be used to measure market sentiment.

4.2. Ontology diagram

The financial ontology overview in Figure 2 shows the objects and data constituting the use case, namely historical and synthetic data and financial algorithms. They run inside the `PeractonSecuritiesPlatform` class that ingests the `SyntheticStocksData` and `SyntheticCommoditiesData` generated by the `Graph-MassivizerPlatform` class.

4.3. Synthetic ontology diagram

The synthetic financial data ontology in Figure 3 shows the created categories and mirrors the original historical financial data set structure. The `SyntheticSymbol` belongs to `SyntheticCommoditiesData` and `SyntheticStocks` classes, with the `TechnicalData` and `FundamentalData` classes as features. It also shows the `SyntheticFinancialData` class with `SyntheticCommoditiesMultiverse` and `SyntheticStocksMultiverse` subclasses, with further `SyntheticCommoditiesData` and `SyntheticStocksData` subclasses.

5. Related Work

We briefly review the literature on applying KG in financial data analysis, then elaborate on other financial data modeling and generation methods.

5.1. Ontology in financial data analysis

Several methods leverage ontology and KG in financial analysis, such as KG extraction and enrichment, querying and reasoning over KGs, extracting correlations, and modeling financial time-series.

[8] studies the effect of considering fundamental and technical data in stock price prediction ML models, showing that models benefiting from both indicators outperform models considering them alone. The method of [12] proposes KG extraction, enrichment, and querying methods. [13] drives a high-quality financial KG given the ontology by a semi-automated method and utilizes this KG in reasoning, stock prediction, and generation with two neural network models, multi-layer perceptron and long short term memory (LSTM). [14] provides an ontology-based correlation extraction between different companies. It drives the network of companies by assessing time-series and uses the node2vec and k-nearest neighbor (kNN) methods to embed and cluster the extracted nodes. [15] proposes a joint

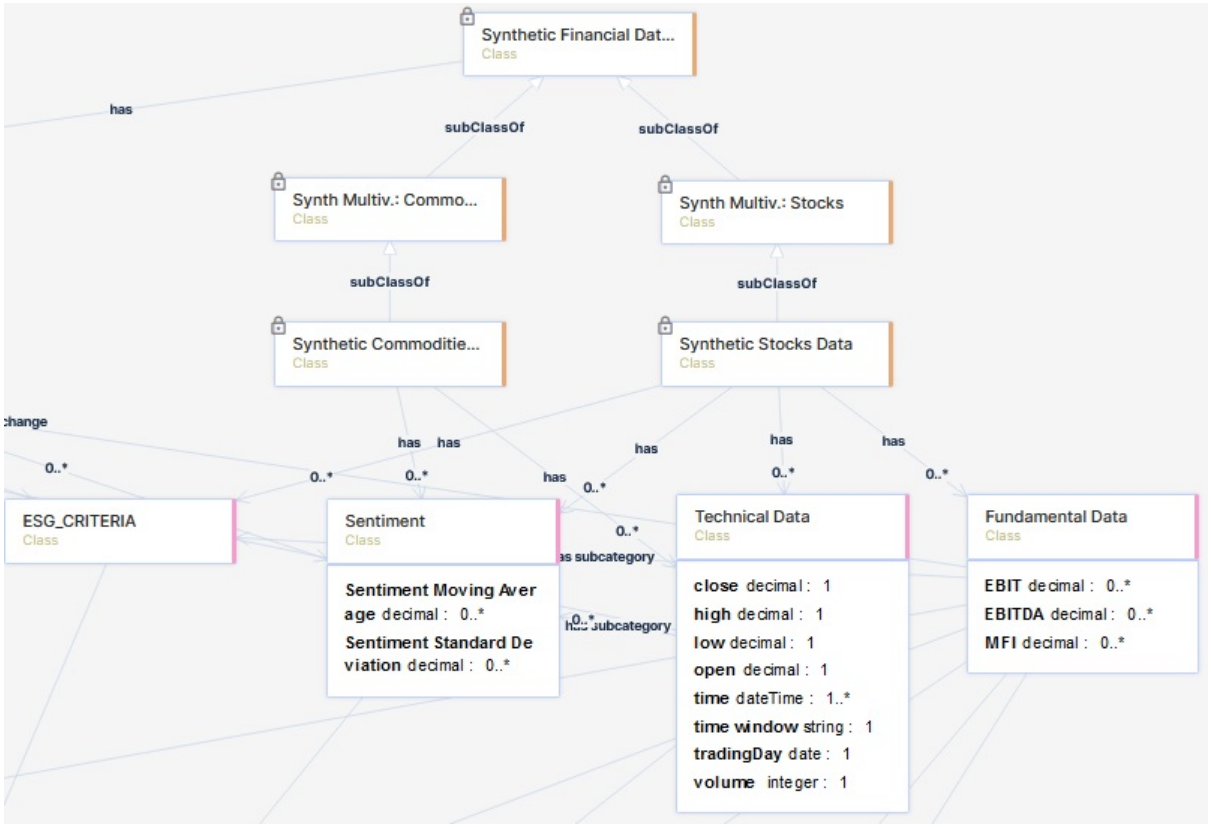

Figure 3: Synthetic financial ontology diagram.

graph learning and prediction model on time-series data. It uses KG and graph neural networks (GNNs) to derive the correlation among different time-series. The method of [16] benefits from KGs in finding first and second-order relationships among companies. It applies an LSTM for time-series embedding of each node and a temporal graph convolutional network (GCN) to incorporate the varying neighborhood effect. The method in [17] formulates stock prediction as a stochastic optimization and introduces genetic programming with a generalized crowding method using financial KG to predict prices.

5.2. Financial time-series modeling

Statistical models Various statistical linear models such as autoregressive models exist for stock prices, however, they can't capture the complex non-linear structure of these data [18].

Event based models These methods formulate time-series data as event data and define it as remarkable changes in time-series in continuous time. The methods of [19] and [20] construct correlation (influence) graphs of time-series utilizing the Hawkes process. [20] defines events as long-lasting volatility values. It uses an attention layer to weigh and capture neighborhoods and an LSTM to predict the next price. The method in [21] uses an event graph and tackles dimensionality. It formulates the intensity function utilizing GNNs and the attention layer to dynamically embed node features and recurrent neural networks (RNNs) to embed event sequences. Graphical event models are another form of marked point processes event type utilizing graphical information for event graph construction [21].

Pattern recognition These methods perform pattern-matching techniques to predict trends in time-series, including perceptually important points, template matching, and dynamic tree wrapping algorithm. The survey [22] provides a detailed review of these methods.

ML models The review in [22] categorizes these models into supervised and unsupervised models. Supervised learning methods use various ML models such as support vector machines, random forest, Adaboost, kNN, and eXtreme gradient boosting methods for stock prediction. The unsupervised learning methods include clustering methods to help in finding correlations among markets [22].

Recently, several studies used deep learning models for modeling financial time-series data consisting of convolutional neural networks, RNNs, attention mechanisms, and generative adversarial networks (GANs) capable of capturing non-linear and complex data features. The survey [23] provides a comprehensive review of these models.

GANs are powerful data generation models appropriate for time-series and adjusted for event data generation. The method in [24] proposes a marked event data generation method using separate generators and discriminators for each event type and preserves type correlations using a central discriminator. This method provides synthetic data for downstream tasks. The GAN model of [18] constructs the correlation graph among stocks using different correlation analysis methods and uses GCNs to encode interdependent time-series data. The method in [3] captures correlation among stocks and applies three generative models, including GAN. The survey [4] presents more models of this category.

The other category of methods uses natural language processing (NLP) to benefit from financial text data such as financial news and SEC filings. They consist of N-grams and word2vec embeddings as inputs for prediction models. The survey [22], and paper [16] introduce methods of this category.

6. Correlation Analysis among Financial Data

Financial data can show various correlations across financial products, companies, and markets. These correlations may be based on deeper links between companies and industries or simply random, lacking any underlying economic or financial rationale. We aim to identify the relevant and meaningful correlations within historical financial time-series and use these insights to generate synthetic data.

6.1. Point process approach

Point process models are stochastic processes that successfully model event sequences [25]. They vary depending on the definition of the conditional intensity function, representing the expected number of events in a small time interval given the event history. We consider a multi-dimensional Hawkes (self-exciting) process [26] to model dependencies between various time-series of markets and obtain the influence graph. We convert financial time-series data to event sequences by defining events as relatively significant changes in time-series.

Self-exciting process This process represents the triggering effect of event history in the intensity and occurrence of future events, usually as an exponential exciting kernel (Eq. 1) [26]:

$$\lambda_k(t) = \mu_k + \sum_{i;t_i<t} a_{k_i,k} \cdot e^{-\beta\cdot(t-t_i)}, \tag{1}$$

where $\lambda_k(t)$ is the intensity of event type k at time t, μ_k is the base intensity for even type k, β is the excitation decay rate and $a_{k_i,k}$ is the influence parameter between event types of k_i and k.

We infer model parameters (μ and influence matrix $A = (a_{i,j})$) by maximizing the log-likelihood of events given in Eq. 2 using the expectation-maximization or stochastic gradient descent methods:

$$\mathcal{L} = \sum_{i:t_i<T} \log \lambda_{k_i}(t_i) - \int_0^T \lambda(t)dt, \tag{2}$$

where [0,T] is the time interval of events we consider for correlation analysis, and $\lambda(t)$ is the sum of intensities of all event types.

Stocks' correlation analysis within a given exchange (market) We consider different point processes for the time-series of stocks and obtain the correlation graph among them by analyzing event data and inferring the influence matrix of the multi-dimensional process [27]. This matrix reveals the weighted dependency (influence) graph among different market' stocks. Due to the ever-changing dependencies among companies, we can update this graph over time and drive a dynamic dependency. As a remedy for high dimensional market data, which decreases the accuracy of these models, we can drive an initial dependency graph by processing various textual data using NLP techniques.

Correlation analysis of internal stock data Despite the expressiveness of point process models, they are inaccurate in high-dimensional spaces. In correlation analysis of internal stock data (fundamental and technical), we can mitigate this problem by leveraging KGs and considering only relations appearing in the KG.

Formulating intensity function using neural networks In addition to high dimensionality, the assumption of solid intensity functions such as the formulation in Eq. 1 with predefined triggering effect (positive and additive effect of history), might not apply to every real scenario with intricate dependencies. Therefore, several methods benefit from neural networks such as LSTMs in [28] to formulate the intensity function of the point process models that capture variable and complex dependencies and adjust the model for the specific use case. Additionally, the attention mechanism [29], which can explicitly model the influence of event types, guides in finding the correlation graph.

6.2. Spurious correlations

Every correlation does not represent a causality relationship known as spurious correlation [30]. This correlation can be a random effect or caused by other hidden variables and, therefore, be spurious [30]. In particular, deep learning models usually result in spurious correlations without enough diverse data [31]. To mitigate this misinterpretation, we will apply methods to test the correlations.

- Considering other factors in the stock market in correlation analysis, in addition to stocks, as much as possible.
- Considering long-term correlations among time-series or comparing these correlations in the long term to test if they are not random.
- Applying null hypothesis for finding significant p-values, such as the method of [32], which constructs a spurious relationship graph among time-series of stocks using Granger causal relation test to evaluate causality between time-series and T-test to estimate the p-value.

We aim to prune the initial dependency graph and reduce the dimensionality of point process models by detecting spurious correlations and driving a more meaningful model based on causality dependencies.

7. Modeling Synthetic Financial time-series

An important input in modeling synthetic time-series is using a correlation graph among stocks, as presented in Figure 4. The temporal and dependency features of stocks are embedded using this correlation graph. These features serve as inputs for ML models to generate time-series data. In the following sections, we propose various types of these models:

Improving event-based models using GNNs and LSTM The correlation graph can be used directly for generating time-series. However, to enhance the initial point process models for modeling time-series (that consist of more details than event sequences) and to improve graph weights, we will add GNN with attention layers [33]. This model can simultaneously encode time-series and correlation graph structure and obtains graph weights [20] (Figure 4). Then, we will model the synthetic time-series data using an LSTM given the encoded data [20].

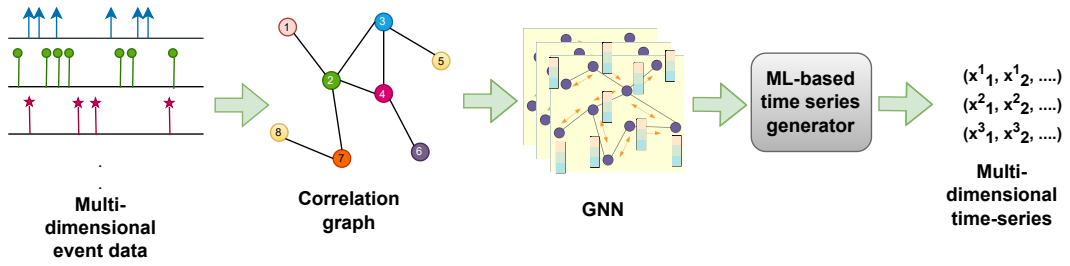

Figure 4: Multi-dimensional synthetic time-series model.

Generating synthetic time-series using correlation graphs The other method combines the correlation graph and GANs to generate interrelated time-series data. We will apply this graph to embed time-series data, capturing their relationships and providing inputs for the GAN model.

8. Implementation Considerations

Generating extreme volumes of synthetic financial time-series data has unique challenges and requirements from both software and hardware perspectives. Here, we outline the most relevant ones.

Scalable data generation The five tools of the Graph-Massivizer platform are being implemented scalable to generate synthetic time-series data across a distributed system. One strategy employed is task-level parallelism, which divides the data generation process into small chunks processed simultaneously across different nodes in the HPC cluster. As a computing framework, Apache Spark provides libraries and functionalities for parallel processing and data management on large clusters.

Data storage and streaming Managing and storing the produced synthetic data at the petabyte level requires various solutions, such as compression, third-party storage, and transferring data only when necessary. Ideally, the synthetic data should remain within the HPC cluster, with financial simulations and testing conducted in the same environment to minimize data movement.

Parallel processing of data generation and large memory capacity We avail CINECA's Leonardo Pre-exascale supercomputer [34] that will allow the parallel processing of synthetic data generation. A low-latency, high-bandwidth network is in place for communication between compute nodes within the HPC cluster to facilitate data exchange

Energy consumption monitoring The Graph-Massivizer platform has a dedicated tool called 'Graph-Greenifier' to monitor and analyze the energy consumption used for generating the synthetic data time-series.

Data security Even though the underlying historical financial time-series data does not contain personally identifiable information, its synthetic data can still be commercially sensitive. Therefore, security measures are necessary to ensure data confidentiality, data integrity, and secure coding practices, such as strict access control, synthetic data encryption at rest and in transit, and digital signatures of the synthetic data to ensure its authenticity and prevent tampering and data logging monitoring.

Cost optimization Finally, cost optimization in generating synthetic data is critical to the entire process. It involves balancing hardware costs, licensing fees, and ongoing maintenance expenses with the desired performance and scalability. The goal is to offer a more cost-effective solution than real historical financial data while maintaining competitiveness in pricing.

9. Conclusions

The work presented in this paper, as part of the Graph-Massivizer EU project, is currently at the halfway point and demonstrates the initial proof-of-concept developments for generating synthetic data in extreme volumes. The mechanisms and approaches identified here will be further implemented as a robust workflow within the five tools of the Graph-Massivizer platform. The focus will be on software implementation, integrating the five tools, and identifying relevant correlations in historical time-series to enhance the quality of the generated synthetic data. After generating synthetic data samples, they will be tested using Peracton's back-testing engine with various investment and trading financial algorithms. The behavior of these algorithms will be analyzed and compared with their performance on real historical data to assess differences and similarities. Feedback will be provided to the Graph-Massivizer platform to fine-tune the quality of the synthetic data further.

Acknowledgement The Graph-Massivizer project has received funding from the European Union's Horizon Research and Innovation Actions under Grant Agreement N° 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] M. Dogariu, L.-D. Ştefan, 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.

[4] S. A. Assefa, D. Dervovic, M. Mahfouz, R. E. Tillman, P. Reddy, M. Veloso, Generating synthetic data in finance: opportunities, challenges and pitfalls, in: Proceedings of the First ACM International Conference on AI in Finance, 2020, pp. 1–8.

[5] M. Stenger, R. Leppich, I. Foster, S. Kounev, A. Bauer, Evaluation is key: a survey on evaluation measures for synthetic time series, Journal of Big Data 11 (2024) 66.

[6] A. Eberhart, P. Haase, W. Schell, metaphactory for massive graphs, in: M. Vieira, V. Cardellini, A. D. Marco, P. Tuma (Eds.), Companion of the 2023 ACM/SPEC International Conference on Performance Engineering, ICPE 2023, Coimbra, Portugal, April 15-19, 2023, ACM, 2023, pp. 215–220. URL: https://doi.org/10.1145/3578245.3585330. doi:10.1145/3578245.3585330.

[7] Investopedia, Fundamental data, 2024. URL: https://www.investopedia.com/terms/f/fundamentalanalysis.asp.

[8] E. Beyaz, F. Tekiner, X.-j. Zeng, J. Keane, Comparing technical and fundamental indicators in stock price forecasting, in: 2018 IEEE 20th international conference on high performance computing and communications; IEEE 16th international conference on smart city; IEEE 4th international conference on data science and systems (HPCC/SmartCity/DSS), IEEE, 2018, pp. 1607–1613.

[9] Investopedia, Technical data, 2024. URL: https://www.investopedia.com/terms/t/technicalanalysis.asp.

[10] Investopedia, Esg data, 2024. URL: https://www.investopedia.com/terms/e/environmental-social-and-governance-esg-criteria.asp.

[11] E. L. Jeffriess, J, Sentiment data, 2024. URL: https://www.eightcap.com/labs/exploring-the-most-common-sentiment-indicators-on-tradingview/.

[12] S. Zehra, S. F. M. Mohsin, S. Wasi, S. I. Jami, M. S. Siddiqui, M. K.-U.-R. R. Syed, Financial knowledge graph based financial report query system, IEEE Access 9 (2021) 69766–69782.

[13] 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.

[1] More information available at: https://graph-massivizer.eu/

[14] C. Erten, D. Kazakov, Ontology graph embeddings and ilp for financial forecasting, in: International Conference on Inductive Logic Programming, Springer, 2021, pp. 111–124.
[15] S. Ibrahim, W. Chen, Y. Zhu, P.-Y. Chen, Y. Zhang, R. Mazumder, Knowledge graph guided simultaneous forecasting and network learning for multivariate financial time series, in: Proceedings of the Third ACM International Conference on AI in Finance, 2022, pp. 480–488.
[16] D. Matsunaga, T. Suzumura, T. Takahashi, Exploring graph neural networks for stock market predictions with rolling window analysis, arXiv preprint arXiv:1909.10660 (2019).
[17] X. Fu, X. Ren, O. J. Mengshoel, X. Wu, Stochastic optimization for market return prediction using financial knowledge graph, in: 2018 IEEE International Conference on Big Knowledge (ICBK), IEEE, 2018, pp. 25–32.
[18] D. Ma, D. Yuan, M. Huang, L. Dong, Vgc-gan: A multi-graph convolution adversarial network for stock price prediction, Expert Systems with Applications 236 (2024) 121204.
[19] J. Etesami, N. Kiyavash, K. Zhang, K. Singhal, Learning network of multivariate hawkes processes: A time series approach, arXiv preprint arXiv:1603.04319 (2016).
[20] T. Yin, C. Liu, F. Ding, Z. Feng, B. Yuan, N. Zhang, Graph-based stock correlation and prediction for high-frequency trading systems, Pattern Recognition 122 (2022) 108209.
[21] K. Yoon, Y. Im, J. Choi, T. Jeong, J. Park, Learning multivariate hawkes process via graph recurrent neural network, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5451–5462.
[22] D. Shah, H. Isah, F. Zulkernine, Stock market analysis: A review and taxonomy of prediction techniques, International Journal of Financial Studies 7 (2019) 26.
[23] W. Jiang, Applications of deep learning in stock market prediction: recent progress, Expert Systems with Applications 184 (2021) 115537.
[24] A. Seyfi, J.-F. Rajotte, R. Ng, Generating multivariate time series with common source coordinated gan (cosci-gan), Advances in neural information processing systems 35 (2022) 32777–32788.
[25] D. J. Daley, D. Vere-Jones, et al., An introduction to the theory of point processes: volume I: elementary theory and methods, Springer, 2003.
[26] A. G. Hawkes, Spectra of some self-exciting and mutually exciting point processes, Biometrika 58 (1971) 83–90.
[27] E. Lewis, G. Mohler, A nonparametric em algorithm for multiscale hawkes processes, Journal of nonparametric statistics 1 (2011) 1–20.
[28] H. Mei, J. M. Eisner, The neural hawkes process: A neurally self-modulating multivariate point process, Advances in neural information processing systems 30 (2017).
[29] Y. Gu, Attentive neural point processes for event forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 7592–7600.
[30] H. A. Simon, Spurious correlation: A causal interpretation, Journal of the American statistical Association 49 (1954) 467–479.
[31] S. Wu, M. Yuksekgonul, L. Zhang, J. Zou, Discover and cure: Concept-aware mitigation of spurious correlation, in: International Conference on Machine Learning, PMLR, 2023, pp. 37765–37786.
[32] G. Li, J. J. Jung, Dynamic relationship identification for abnormality detection on financial time series, Pattern Recognition Letters 145 (2021) 194–199.
[33] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al., Graph attention networks, stat 1050 (2017) 10–48550.
[34] CINECA, High performance computing, leonardo pre-exascale supercomputer, 2024. URL: https://leonardo-supercomputer.cineca.eu/.