Transactions on Computational Science
and Computational Intelligence

Robert Stahlbock · Gary M. Weiss
Mahmoud Abou-Nasr · Cheng-Ying Yang
Hamid R. Arabnia
Leonidas Deligiannidis *Editors*

Advances in Data Science and Information Engineering

Proceedings from ICDATA 2020 and IKE 2020

Springer

Transactions on Computational Science and Computational Intelligence

Series Editor
Hamid Arabnia
Department of Computer Science
The University of Georgia
Athens, Georgia
USA

Computational Science (CS) and Computational Intelligence (CI) both share the same objective: finding solutions to difficult problems. However, the methods to the solutions are different. The main objective of this book series, "Transactions on Computational Science and Computational Intelligence", is to facilitate increased opportunities for cross-fertilization across CS and CI. This book series will publish monographs, professional books, contributed volumes, and textbooks in Computational Science and Computational Intelligence. Book proposals are solicited for consideration in all topics in CS and CI including, but not limited to, Pattern recognition applications; Machine vision; Brain-machine interface; Embodied robotics; Biometrics; Computational biology; Bioinformatics; Image and signal processing; Information mining and forecasting; Sensor networks; Information processing; Internet and multimedia; DNA computing; Machine learning applications; Multiagent systems applications; Telecommunications; Transportation systems; Intrusion detection and fault diagnosis; Game technologies; Material sciences; Space, weather, climate systems, and global changes; Computational ocean and earth sciences; Combustion system simulation; Computational chemistry and biochemistry; Computational physics; Medical applications; Transportation systems and simulations; Structural engineering; Computational electro-magnetic; Computer graphics and multimedia; Face recognition; Semiconductor technology, electronic circuits, and system design; Dynamic systems; Computational finance; Information mining and applications; Astrophysics; Biometric modeling; Geology and geophysics; Nuclear physics; Computational journalism; Geographical Information Systems (GIS) and remote sensing; Military and defense related applications; Ubiquitous computing; Virtual reality; Agent-based modeling; Computational psychometrics; Affective computing; Computational economics; Computational statistics; and Emerging applications. For further information, please contact Mary James, Senior Editor, Springer, mary.james@springer.com.

More information about this series at http://www.springer.com/series/11769

Robert Stahlbock • Gary M. Weiss
Mahmoud Abou-Nasr • Cheng-Ying Yang
Hamid R. Arabnia • Leonidas Deligiannidis
Editors

Advances in Data Science and Information Engineering

Proceedings from ICDATA 2020 and IKE 2020

Springer

Editors
Robert Stahlbock
HBS – Hamburg Business School, Institute of Information Systems
University of Hamburg
Hamburg, Hamburg, Germany

Mahmoud Abou-Nasr
College of Engineering & Computer Science
University of Michigan-Dearborn
Dearborn, MI, USA

Hamid R. Arabnia
Department of Computer Science
University of Georgia
Athens, GA, USA

Gary M. Weiss
Department of Computer & Information Science
Fordham University
New York, NY, USA

Cheng-Ying Yang
Department of Computer Science
University of Taipei
Taipei City, Taiwan

Leonidas Deligiannidis
School of Computing and Data Sciences
Wentworth Institute of Technology
Boston, MA, USA

ISSN 2569-7072 ISSN 2569-7080 (electronic)
Transactions on Computational Science and Computational Intelligence
ISBN 978-3-030-71703-2 ISBN 978-3-030-71704-9 (eBook)
https://doi.org/10.1007/978-3-030-71704-9

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

It gives us great pleasure to introduce this collection of papers that were presented at the following international conferences: Data Science (ICDATA 2020) and Information & Knowledge Engineering (IKE 2020). These two conferences were held simultaneously (same location and dates) at Luxor Hotel (MGM Resorts International), Las Vegas, USA, July 27–30, 2020. This international event was held using a hybrid approach, that is, "in-person" and "virtual/online" presentations and discussions.

This book is composed of nine Parts. Parts I through V (composed of 46 chapters) include chapters that address emerging trends in data science (ICDATA). Parts VI through IX (composed of 25 chapters) include a collection of chapters in the areas of information and knowledge engineering (IKE).

An important mission of the World Congress in Computer Science, Computer Engineering, and Applied Computing, CSCE (a federated congress to which this event is affiliated with), includes *"Providing a unique platform for a diverse community of constituents composed of scholars, researchers, developers, educators, and practitioners. The Congress makes concerted effort to reach out to participants affiliated with diverse entities (such as: universities, institutions, corporations, government agencies, and research centers/labs) from all over the world. The congress also attempts to connect participants from institutions that have* ***teaching*** *as their main mission with those who are affiliated with institutions that have* ***research*** *as their main mission. The congress uses a quota system to achieve its institution and geography diversity objectives."* By any definition of diversity, this congress is among the most diverse scientific meeting in the USA. We are proud to report that this federated congress had authors and participants from 54 different nations representing variety of personal and scientific experiences that arise from differences in culture and values.

The program committees (refer to subsequent pages for the list of the members of committees) would like to thank all those who submitted papers for consideration. About 50% of the submissions were from outside the USA. Each submitted paper was peer reviewed by two experts in the field for originality, significance, clarity, impact, and soundness. In cases of contradictory recommendations, a member of the conference program committee was charged to make the final decision, often this involved seeking help from additional referees. In addition, papers whose authors included a member of the conference program committee were evaluated using the double-blind review process. One exception to the above evaluation process was for papers that were submitted directly to chairs/organizers of pre-approved sessions/workshops; in these cases, the chairs/organizers were responsible for the evaluation of such submissions. The overall paper acceptance rate for regular papers was 20%; 18% of the remaining papers were accepted as short and/or poster papers.

We are grateful to the many colleagues who offered their services in preparing this book. In particular, we would like to thank the members of the program committees of individual research tracks as well as the members of the steering committees of ICDATA 2020 and IKE 2020; their names appear in the subsequent pages. We would also like to extend our appreciation to over 500 referees.

As sponsors-at-large, partners, and/or organizers, each of the following (separated by semicolons) provided help for at least one research track: Computer Science Research, Education, and Applications (CSREA); US Chapter of World Academy of Science; American Council on Science and Education & Federated Research Council; and Colorado Engineering Inc. In addition, a number of university faculty members and their staff, several publishers of computer science and computer engineering books and journals, chapters and/or task forces of computer science associations/organizations from three regions, and developers of high-performance machines and systems provided significant help in organizing the event as well as providing some resources. We are grateful to them all.

We express our gratitude to all authors of the articles published in this book and the speakers who delivered their research results at the congress. We would also like to thank the following: UCMSS (Universal Conference Management Systems & Support, California, USA) for managing all aspects of the conference; Dr. Tim Field of APC for coordinating and managing the printing of the programs; the staff at Luxor Hotel (MGM Convention) for the professional service they provided; and Ashu M. G. Solo for his help in publicizing the congress. Last but not least, we would like to thank Ms. Mary James (Springer Senior Editor in New York) and

Arun Pandian KJ (Springer Production Editor) for the excellent professional service they provided for this book project.

Hamburg, Germany — Robert Stahlbock

New York, NY, USA — Gary M. Weiss

Dearborn, MI, USA — Mahmoud Abou-Nasr

Taipei City, Taiwan — Cheng-Ying Yang

Athens, GA, USA — Hamid R. Arabnia

Boston, MA, USA — Leonidas Deligiannidis

Book Co-editors and Chapter Co-editors:
Advances in Data Science and Information Engineering + ICDATA 2020 & IKE 2020

Preface

It gives us great pleasure to introduce this collection of papers that were submitted and accepted for the 16th International Conference on Data Science 2020, ICDATA'20 (https://icdata.org), July 27–30, 2020, at Luxor Hotel, Las Vegas, USA. Obviously, the year 2020 is very different from others due to the Covid-19 pandemic that had severe impact on all our lives. That was not at the horizon when planning the conference. The conference was held, but almost all authors were not allowed to travel during the summer, and even if it would have been allowed, it would have been wise to stay at home instead of travelling if possible. As a consequence, the typical communication, face to face, during sessions, in front of the conference rooms and during social events, was replaced by the opportunity to give talks via the web, either as pre-recorded talk or "live." All organizers and presenters did their best in that situation. Thank you very much for all your effort!

Some words about ICDATA and data mining: data mining or machine learning is critically important if we want to effectively learn from the tremendous amounts of data that are routinely being generated in science, engineering, medicine (take Covid-19 and the search for better understanding of the disease as well as for medicine and better treatment as an example), business, sports and e-sports, and other areas. The aim is gaining insight into processes and transactions, extract knowledge, make better decisions, and deliver value to users or organizations. This is even more important and challenging in an era in which scientists and practitioners are faced with numerous challenges caused by exponential expansion of digital data, its diversity, and complexity. The scale and growth of data considerably outpace technological capacities of organizations to process and manage it. During the last decade, we all observed new, more glorious, and promising concepts or labels emerging and slowly but steadily displacing "data mining" from the agenda of CTOs. It was and still is the time, more than ever before, of data science, big data, advanced-/business-/customer-/data-/predictive-/prescriptive-/ . . . /risk-analytics, to name only a few terms that dominate websites, trade journals, and the general press – although there is even a rebirth of terms such as artificial intelligence (AI) and (machine) learning (e.g., deep learning) in academia, companies, and even on the agenda of political decision makers.

All the concepts of data science aim at leveraging data for a better understanding of complex real-world phenomena. They all pursue this objective using some formal, often algorithmic, procedures, at least to some extent. This is what data miners have been doing for decades. The very idea of all those similar or identical concepts with different labels; the idea to think of massive, omnipresent amounts of data as strategic assets; and the aim to capitalize on these assets by means of analytic procedures is, indeed, more relevant and topical than ever before. Although there are very helpful advances in hardware and software, there are still many challenges to be tackled in order to leverage the promises of data analytics. Obviously, technological change is never ending and appears to be accelerating. The world is especially focused on machine learning and data mining (not contradictory but similar or even equivalent to data science), as these disciplines are making an ever-increasing impact on our society. Large multinational corporations are expanding their efforts in these areas, small startups are founded, and students are flocking to computer science and related disciplines in order to learn about these disciplines and take advantage of the many lucrative job opportunities. Many industries, even conservative ones like, for example, the port industry, are working towards "Version 4.0" (e.g., "Port 4.0"), with digitization, digitalization, and even digital transformation of traditional processes resulting in improved workflows, new concepts, and new business plans. Their goal usually includes data analytics, automation, autonomization, robotics, and AI. The industry is interested in feasibility studies and results of scientific research. Data science is popular like never before. Data scientists are rare on the job market and, therefore, very well compensated.

The growth in all these areas has been dramatic enough to require changes in nomenclature. Most of these "hot" technologies and methods are increasingly considered part of the broad field of data science, and there are benefits to viewing this field as a unified whole, rather than a collection of disparate sub-disciplines. ICDATA, the former data mining conference DMIN merged with the big data conference ABDA, is much broader than just data mining and big data. It includes all of the following main topics: all aspects of data mining and machine learning (tasks, algorithms, tools, applications, etc.), all aspects of big data (algorithms, tools, infrastructure, and applications), data privacy issues, and data management. The conference is designed to be of equal interest to researchers and practitioners, academics and members of industry, computer scientists, physical and social scientists, and business analysts.

ICDATA'20 attracted submissions of theoretical research papers as well as industrial reports, application case studies, and, in a second phase, late breaking papers, position papers, and abstract/poster papers. The program committee would like to thank all those who submitted papers for consideration. We strived to establish a review process of high quality. To ensure a fair, objective, and transparent review process, all review criteria are published on the website. Papers were evaluated regarding their relevance to ICDATA, originality, significance, information content, clarity, and soundness on an international level. Each aspect was objectively evaluated, with alternative aspects finding consideration for application

papers. Each paper was refereed by at least two researchers in the topical area, taking the reviewers' expertise and confidence into consideration, with most of the papers receiving three reviews. The review process was competitive. The overall acceptance rate for submissions was 47%.

We are very grateful to the colleagues who helped in organizing the conference. In particular, we would like to thank the members of the program committee of ICDATA'20 and the members of the congress steering committee. The continuing support of the ICDATA program committee has been essential to further improve the quality of accepted submissions and the resulting success of the conference. The ICDATA'20 program committee members are (in alphabetical order): Mahmoud Abou-Nasr, Ruhul Amin, Jérôme Azé, Kai Brüssau, Paulo Cortez, Zahid Halim, Tzung-Pei Hong, Wei-Chiang Hong, Andrew Johnston, Madjid Khalilian, Robert Stahlbock, Chamont Wang, Gary M. Weiss, Yijun Zhao, and Zijiang Yang. They all did a fantastic job in evaluating a lot of submissions in very short time. We are aware that their workload was particularly high due to the Covid-19 situation, so we are grateful for their support of ICDATA'20. The conference's quality depends on reliable and good reviewers. We would also like to thank Mahmoud Abou-Nasr for organizing the special session on "Real-World Data Mining & Data Science Applications, Challenges, and Perspectives" for more than a decade. We would like to thank our publicity co-chair Ashu M. G. Solo (Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.) for circulating information on the conference, as well as www.KDnuggets.com, a platform for analytics, data mining, and data science resources, for listing ICDATA'20. We are also grateful for support by the Institute of Information Systems at Hamburg University, Germany and would like to thank all supporters and sponsors of CSCE. Last but not least, we wish to express again our sincere gratitude and utmost respect towards our colleague and friend Prof. Hamid R. Arabnia (Professor, Department of Computer Science, University of Georgia, USA; Editor-in-Chief, *Journal of Supercomputing* [Springer]), General Chair and Coordinator of the federated congress, and also Associate Editor of ICDATA'20, for his excellent, tireless, and continuous support, organization, and coordination of all affiliated events, particularly in these hard and difficult times of Covid-19. His exemplary and professional effort in 2020 and all the earlier years in the steering committee of the congress make these events possible. We are grateful to continue our data science conference as ICDATA'20 under the umbrella of the CSCE congress.

Thank you all for your contribution to ICDATA'20! We hope to see you at ICDATA'21. Stay safe and healthy!

We present the proceedings of ICDATA'20.

ICDATA'20 General Conference Chair Robert Stahlbock

Steering Committee ICDATA'20
https://icdata.org

Data Science: ICDATA 2020 – Organizing Committee (Leadership)

- *Dr. Robert Stahlbock (ICDATA 2020 Chair); University of Hamburg, Germany*
- *Dr. Gary M. Weiss; Fordham University, New York, USA*
- *Dr. Sven F. Crone; Lancaster University, UK*
- *Dr. Mahmoud Abou-Nasr, USA*
- *Dr. Hamid R. Arabnia, USA*

For the complete list of program committee refer to: https://icdatascience.org/

Information & Knowledge Engineering: IKE 2020 – Program Committee

- *Prof. Emeritus Nizar Al-Holou (Congress Steering Committee); Vice Chair, IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Emeritus Hamid R. Arabnia (Congress Steering Committee); The University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Dr. Travis Atkison; Director, Digital Forensics and Control Systems Security Lab, Department of Computer Science, College of Engineering, The University of Alabama, Tuscaloosa, Alabama, USA*
- *Dr. Arianna D'Ulizia; Institute of Research on Population and Social Policies, National Research Council of Italy (IRPPS), Rome, Italy*
- *Prof. Emeritus Kevin Daimi (Congress Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Zhangisina Gulnur Davletzhanovna; Vice-rector of the Science, Central-Asian University, Kazakhstan, Almaty, Republic of Kazakhstan; Vice President of International Academy of Informatization, Kazskhstan, Almaty, Republic of Kazakhstan*
- *Prof. Leonidas Deligiannidis (Congress Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*
- *Prof. Mary Mehrnoosh Eshaghian-Wilner (Congress Steering Committee); Professor of Engineering Practice, University of Southern California, California, USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA*
- *Prof. Ray Hashemi (Session Chair, IKE & Steering Committee member); Professor of Computer Science and Information Technology, Armstrong Atlantic State University, Savannah, Georgia, USA*

- *Prof. Dr. Abdeldjalil Khelassi; Computer Science Department, Abou beker Belkaid University of Tlemcen, Algeria; Editor-in-Chief, Medical Technologies Journal; Associate Editor, Electronic Physician Journal (EPJ) - Pub Med Central*
- *Prof. Louie Lolong Lacatan; Chairperson, Computer Engineerig Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of Computer Science and Information Technology (IACSIT), Singapore; Member, International Association of Online Engineering (IAOE), Austria*
- *Dr. Andrew Marsh (Congress Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations*
- *Dr. Somya D. Mohanty; Department of CS, University of North Carolina - Greensboro, North Carolina, USA*
- *Dr. Ali Mostafaeipour; Industrial Engineering Department, Yazd University, Yazd, Iran*
- *Dr. Houssem Eddine Nouri; Informatics Applied in Management, Institut Superieur de Gestion de Tunis, University of Tunis, Tunisia*
- *Prof. Dr., Eng. Robert Ehimen Okonigene (Congress Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Technology, Ambrose Alli University, Nigeria*
- *Prof. James J. (Jong Hyuk) Park (Congress Steering Committee); Department of Computer Science and Engineering (DCSE), SeoulTech, Korea; President, FTRA, EiC, HCIS Springer, JoC, IJITCC; Head of DCSE, SeoulTech, Korea*
- *Dr. Prantosh K. Paul; Department of CIS, Raiganj University, Raiganj, West Bengal, India*
- *Dr. Xuewei Qi; Research Faculty & PI, Center for Environmental Research and Technology, University of California, Riverside, California, USA*
- *Dr. Akash Singh (Congress Steering Committee); IBM Corporation, Sacramento, California, USA; Chartered Scientist, Science Council, UK; Fellow, British Computer Society; Member, Senior IEEE, AACR, AAAS, and AAAI; IBM Corporation, USA*
- *Chiranjibi Sitaula; Head, Department of Computer Science and IT, Ambition College, Kathmandu, Nepal*
- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*
- *Prof. Fernando G. Tinetti (Congress Steering Committee); School of CS, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Varun Vohra; Certified Information Security Manager (CISM); Certified Information Systems Auditor (CISA); Associate Director (IT Audit), Merck, New Jersey, USA*
- *Dr. Haoxiang Harry Wang (CSCE); Cornell University, Ithaca, New York, USA; Founder and Director, GoPerception Laboratory, New York, USA*

- *Prof. Shiuh-Jeng Wang (Congress Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.*
- *Prof. Layne T. Watson (Congress Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*
- *Prof. Jane You (Congress Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Contents

Part I
Graph Algorithms, Clustering, and Applications

Phoenix: A Scalable Streaming Hypergraph Analysis Framework

Kuldeep Kurte, Neena Imam, S. M. Shamimul Hasan, and Ramakrishnan Kannan

1 Introduction

Over the last few years, we have witnessed the explosive growth of data due to the technological advancements in the fields of social networking, e-commerce, smart mobile devices, etc. This necessitates the development of novel data mining/analysis approaches to address the various analytical challenges posed by the massive growth in data. Some examples of data analytics include live tracking in the transportation sector, fraud management in insurance, product recommendations in the retail industry, and predictive analysis in health care. These analyses study the relations, dynamics, and behavior at an individual level (entity level) as well as at the group level. The graph representation, $G = (V, E)$, in which entities are represented by vertices ($V = \{v_1, v_2, .., v_n\}$) and relations among entities are represented by edges ($E = \{e_1, e_2, \ldots, e_m\}$), is a natural way to model such relational information. For instance, in an e-commerce system, customers and products are modeled as vertices, and customer-product relations are represented by edges.

This manuscript has been authored in part by UT-Battelle, LLC, under contract no. DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

K. Kurte (✉) · N. Imam · S. M. S. Hasan · R. Kannan
Computing and Computational Sciences Directorate, Oak Ridge National Laboratory, Oak Ridge, TN, USA
e-mail: kurtekr@ornl.gov; imamn@ornl.gov; hasans@ornl.gov; kannanr@ornl.gov

R. Stahlbock et al. (eds.), *Advances in Data Science and Information Engineering*, Transactions on Computational Science and Computational Intelligence,
https://doi.org/10.1007/978-3-030-71704-9_1

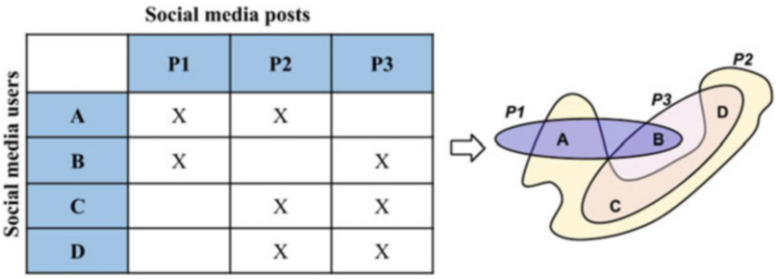

Fig. 1 Example hypergraph showing social media users (rows) and three social media posts (columns). Each post P_i is represented as an hyperedge, and those users who interacted with that post are the hypergraph vertices incident on that hyperedge

The graph representation of the information is able to capture the dyadic relations, i.e., relations between two entities, but fails to model the group-level interactions. Due to the fact that the individual's behavior is mainly influenced by the group-level interactions, modeling group-level dynamics is important. Hypergraphs—the generalization of graphs—provide an excellent way to model the group-level interactions [6, 9, 28]. A hypergraph $HG = (V, H)$ is an ordered pair of "n" vertices, i.e., $V = \{v_1, v_2, v_3, \ldots, v_n\}$, and H is a set of "m" hyperedges, i.e., $H = \{H_1, H_2, H_3, \ldots, H_m\}$. Each hyperedge H_i is a vector of incident vertices such that $V \equiv h_1 \cup h_2 \cup h_3 \cup \ldots \cup h_m$. Figure 1 shows an example hypergraph which includes four social network users, A, B, C, D, and three social media posts, P_1, P_2, P_3. Each post P_i represents a hyperedge, and its incident vertices are the users who interacted with the content, say, shared, liked, or commented on the post (represented by "X"). From this example, it is evident that such hypergraph-based representation is useful to understand the information propagation among entities and the categorization of groups according to specific interests over the social network.

Although the efficacy of hypergraphs for modeling group dynamics is well documented [1], efficient hypergraph analytics must overcome challenges associated with accurate hypergraph representation and scalable computation models that can deal with very high data ingestion rates without creating bottlenecks. While several large-scale graph processing software are available such as [5, 7, 18, 26], only a limited number of options are available for *hypergraph* analysis frameworks [28]. Very-large-scale hypergraph analysis requires scalable and distributed computing systems which present novel challenges as well as opportunities. The situation becomes more challenging when streaming data need to be incorporated in the framework. Some challenges posed by the streaming scenario include variability in the streaming rates from various external hypergraph sources, heterogeneity in representing the hypergraph, and efficient hypergraph representation at a system level to sustain the streaming scenario.

Little research has been done for methodical performance evaluation of large-scale hypergraph analysis frameworks in a streaming scenario. The leadership

class high-performance computing facilities, such as hosted at Oak Ridge National Laboratory, provide petascale to exascale computing powers, large amounts of per node memory, efficient storage, and high-speed interconnects. Such leadership class computing facilities can meet the computational requirements of large-scale streaming hypergraph analysis. As such, researchers at Oak Ridge National Laboratory developed *Phoenix*, a high-performance, hybrid system enabling concurrent utilization of online and offline analysis worlds. Phoenix architecture is distributed for scalability of problem size and performance. In addition, Phoenix is designed for fast and scalable ingest of streaming data sources. Phoenix also incorporates fast online (CRUD) operations and has dynamic (and fixed) schema. Using Phoenix, researchers are able to perform fast decoupled offline global analytics with in-memory snapshots and commit logs. Phoenix was deployed on Oak Ridge National Lab's Titan (ranked number one on top500[1] list in 2012) and showed good performance. Originally designed for simple graph analytics, we recently enhanced Phoenix to handle *hypergraph*s. The performance of Phoenix for streaming datasets is the subject of this paper.

In the following sections, we present our approach to scalable streaming hypergraph analysis as implemented in Phoenix. Section 2 presents an overview of the various hypergraph analysis tools. Section 3 presents the Phoenix framework for streaming hypergraph analysis and describes various technical aspects of Phoenix. Section 4 presents results of the numerical experiments we performed to evaluate metrics such as streaming performance, ingestion performance, and hypergraph clustering efficiency. Section 5 summarizes our observations and discusses few future extensions of this work.

2 Related Work

Many hypergraph analysis tools are available. However, none of these tools presents the scalability and flexibility associated with Phoenix. In addition, Phoenix incorporates scalable hypergraph generators. Most other hypergraph analytics software tools do not have this attribute. In the following paragraphs, we present an overview of the various hypergraph analysis tools and the advantages and disadvantages of each.

HyperNetX is a Python library that supports hypergraph creation, hypergraph-connected component computation, sub-hypergraph construction, hypergraph statistics computation (e.g., node degree distribution, edge size distribution, toplex size computation for hypergraphs), and hypergraph visualization (e.g., draw hypergraphs, color nodes, and edges). *HyperNetX* was released in 2018 under the Battelle Memorial Institute license [21]. *HyperNetX* library does not support high-

[1] https://www.top500.org/system/177975.

performance computing (HPC)-based parallel processing. Also, *HyperNetX* library documentation does not provide any scalability information.

Chapel HyperGraph Library (CHGL) was developed in the Chapel programming language by the Pacific Northwest National Laboratory. In the CHGL, users can use both shared and distributed memory systems for the storage of hypergraphs. The CHGL is not well documented and requires knowledge of the Chapel programming language, which is Partitioned Global Address Space (PGAS) language. PGAS languages are not as widely used as the C or C++ programming language. However, CHGL does offer valuable functionality within the context of parallel computations [2, 4].

HyperX offers a scalable framework for hypergraph processing and learning algorithms, which is developed on top of Apache Spark. It replicates the design model that is utilized within GraphX. *HyperX* directly processes the hypergraph rather than converting the hypergraph to a bipartite graph and employs GraphX to do the processing [2, 15]. Apache Spark programming paradigm cannot match the scalability offered by a leadership class computing platform.

HyperGraphLib package was developed in the C++ programming language, which supports k-uniform, k-regular, simple, linear, path search, and isomorphism algorithms. *HyperGraphLib* employs both OpenMP and Boost libraries. *HyperGraphLib* cannot represent a hypergraph as a bipartite graph or a two-section graph. Moreover, *HyperGraphLib* is not integrated with any graph libraries for advanced analytics [2, 14].

Halp is a Python library that provides both directed and undirected hypergraph implementations as well as a range of algorithms. These include a variety of hypergraph algorithms–for instance, k-shortest hyperpaths as well as random walk and directed paths [2, 13]. However, *Halp* does not provide parallel implementation of the algorithms.

SAGE hypergraph generator was developed in the Python language and supports the creation of complete random, uniform, and binomial random uniform hypergraphs. Nevertheless, large-scale hypergraph generation is not possible in *SAGE*. Besides, *SAGE* does not support parallel hypergraph generation.

Karlsruhe Hypergraph Partitioning (KaHyPar) was developed in C++ and is a multilevel hypergraph partitioning framework. It supports hypergraph partitioning with variable block weights and fixed vertices. Although *KaHyPar* is a useful tool, it does not support the hypergraph generation facility [16, 24].

The Julia programming language was used to develop the *SimpleHypergraphs.jl* hypergraph analysis framework. It is an efficient hypergraph analysis tool that supports distributed computing. However, *SimpleHypergraphs.jl* is heavily dependent on the *HyperNetX* library, specifically for hypergraph visualization. Moreover, *SimpleHypergraphs.jl* tool provides limited hypergraph analysis functionalities and is not highly scalable [2].

networkR was developed in the R programming language, which supports hypergraphs' projection into graphs. networkR also supports degree distribution, diameter, centrality, and network density computation. One of the limitations of the *networkR* is that it needs to project hypergraph into graph structure for analysis.

Moreover, vertices and hyperedge-related meta-information is not available in networkR [2, 20].

Gspbox provides hypergraph modeling capability. Although in Gspbox, one can manipulate the hypergraph by transforming a model into a regular graph, it does not provide specific solutions or optimizations for hypergraphs [2, 8].

BalancedGo software was developed in the Go programming language. BalancedGo supports generalized hypertree decompositions via balanced separators. *BalancedGo* supports a limited number of algorithms mainly focused on hypertree decompositions. Moreover, *BalancedGo* supports only HyperBench format or PACE Challenge 2019 format [3] as input.

Pygraph was released under the MIT license and is a Python library that can be used to process graphs. It includes hypergraph support along with standard graph functionalities. However, Pygraph does not offer any hypergraph optimization feature [2, 22].

Yadati et al. developed *HyperGCN*, a new graph convolutional network (GCN) training approach for semi-supervised learning (SSL) on hypergraphs [30]. The Python implementation of the tool is available in [12]. The quality of the hypergraph approximation heavily depends on weight initialization, which is a limitation of *HyperGCN* [30].

Multihypergraph is a Python package that provides support for multi-edges, hyperedges, and looped edges. The main focus of the *Multihypergraph* package is the mathematical understanding of graph than algorithmic efficiency. Moreover, the *Multihypergraph* package is limited with graph model memory definition and isomorphism functionalities and does not provide any other functionalities for hypergraphs [2, 19].

d3-hypergraph is a hypergraph visualization tool developed on top of the D3 JavaScript library. Another example of the hypergraph visualization tool is *visualsc,* which is similar to the open-source graph visualization tool Graphviz. *d3-hypergraph* and *visualsc* tools are solely used for hypergraph visualization.

3 Framework for Analyzing Streaming Hypergraphs

This section describes the overall Phoenix framework and its various components which enable the analysis of the streaming hypergraph. Figure 2 shows Phoenix's end-to-end framework which is composed of various essential modules for analyzing the streaming hypergraphs in a distributed and scalable fashion.

3.1 Hypergraph Sources and Generation

Phoenix is capable of utilizing a diverse set of graph generators as inputs to the framework. One of the candidates is a distributed hypergraph generator called

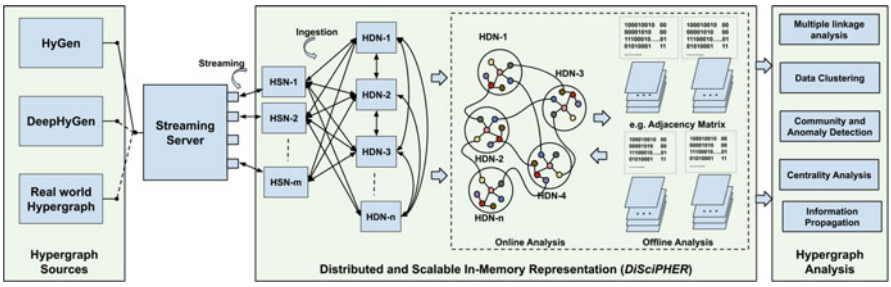

Fig. 2 Phoenix's end-to-end framework for scalable and distributed hypergraph analysis. Streaming server acts as a gateway where various hypergraph generators/external sources can connect. Next the streaming server streams the hypergraph in the form of hyperedge or incidences to the graph service nodes (GSNs). GSNs handle the communication with the streaming server and consume the hypergraph and send it to the graph data nodes (GDNs) where GDNs store the ingested hypergraph as its in-memory representation

HyGen, which is capable of generating synthetic hypergraphs. HyGen is another high-performance graph analytics project at Oak Ridge National Laboratory and was incorporated in the Phoenix architecture. HyGen takes input parameters such as number of clusters, number of vertices, and number of hyperedges to generate the corresponding hypergraph. For instance, if we have a rough understanding about the number of the clusters in the real-world hypergraph (e.g., communities), HyGen will enable the rapid production of the different sizes of hypergraphs which can be further consumed (by HSNs) and stored in-memory (by HDNs) in a distributed fashion. Refer to Fig. 2 and Sect. 3.3 for more information on HSNs and HDNs. Further, various online and offline analyses can be performed on this generated hypergraph. Similarly, the external hypergraph sources can also connect to the streaming server. More detailed discussions on graph generators can be found in references [17, 27, 32].

3.2 Hypergraph Streaming and Consumption

A streaming server is developed to facilitate the streaming of hypergraphs generated by hypergraph generators and from external sources to the internal core component called *DiSciPHER* (refer Sect. 3.3) which is responsible for hypergraph consumption and in-memory storage. The three advantages of having this layer of streaming server are as follows:

1. **Decoupling**: Streaming server acts as a gateway and prevents hypergraph generators and external sources from directly accessing the *DiSciPHER* which is a core internal module of Phoenix. This provides the flexibility to make changes in the *DiSciPHER* module without impacting the accessibility of the hypergraph

sources. Moreover, syntactic changes made by hypergraph sources do not have any impact on the *DiSciPHER*'s representation.

2. **Standardization**: Streaming servers can acquire data either as a bipartite representation or as a hyperedge representation. It is unlikely that all external sources comply with a unified syntax even though the data follow the semantics of bipartite or hyperedge representation. The streaming server can implement various methods for data translation to address this syntactic heterogeneity problem.
3. **Intermediate caching**: The rate of streaming from different external sources of hypergraphs can be different. At the system level, the heterogeneity in the streaming rates could cause data loss in case of extremely high data streaming and longer wait time for HSN processes in case of slow data streaming. We believe that the intermediate layer of the streaming server can stabilize the rate of streaming hypergraph from various external sources to HSN. The streaming server can provide a temporary storage capability to store the acquired hypergraph data before sending it to the HSNs of *DiSciPHER* module. This way streaming servers can stabilize the streaming rate.

The streaming server can acquire hypergraphs in one of two ways: (1) bipartite representation, a list of incidences, and (2) hyperedge representation, a list of hyperedges. Each incidence in a bipartite representation is a two-dimensional vector $\langle i, j \rangle$, such that $v_i \in H_j$, i.e., vertex v_i incident upon hyperedge H_j. On the other hand, the hyperedge representation constitutes a set of hyperedges (H) in which each hyperedge is a vector of incident vertices, i.e., $H_k = \langle v_{k1}, v_{k2}, v_{k3}, \ldots, v_{kp} \rangle$ and "p" is the total number of incident vertices on hyperedge k.

The streaming server opens multiple communication ports where several hypergraph service node (HSN) processes of *DiSciPHER* module, which is responsible for the consumption of the hypergraph, can connect and consume the hypergraph. In the case of bipartite representation, the streaming server performs streaming of incidences in a batched fashion. The batch size represents the maximum number of hypergraph incidences that can be packed in a batch. The batch size in case of hyperedge representation is the maximum number of hypergraphs per batch. Due to the variable size of hyperedges in a batch, the batch creation is not as straightforward as in the bipartite representation. Here, each hyperedge is reformatted as $\langle h_{id}, p, v_1, v_2, v_3, \ldots, v_p, -1 \rangle$ by appending hyperedge identifier h_{id}, its length in the beginning p, followed by a list of incident vertices, i.e., v_i and "-1" at the end to indicate the termination of the hyperedge. In this way, the hypergraphs are packed to form a batch such that each element in the batch represents either hypergraph identifier, length of hypergraph, vertex identifier, or "-1."

As mentioned in the paragraph above, the hypergraph service node processes (HSNs) connect to the communication ports of the streaming server and consume a hypergraph either as a batched incidences or as hyperedges. We implemented a handshaking and communication protocol to enable the streaming and consumption of the hypergraphs. Figure 3 shows a sequence of commands and data exchanges