❒   1140

A Survey on Graph Database Management Techniques for Huge Unstructured Data

Patil N. S.[1], Kiran P[2], Kavya N. P.[3], Naresh Patel K. M.[4]
[1]Information Science & Engineering, BIET, Davangere, Karnataka, India
[2]Computer Science & Engineering, RNSIT, Bengaluru, Karnataka, India
[3]MCA Department, RNSIT, Bengaluru, Karnataka, India
[4]Computer Science & Engineering, BIET, Davangere, Karnataka, India

Article Info	ABSTRACT
	Data analysis, data management, and big data play a major role in both social and business perspective, in the last decade. Nowadays, the graph database is the hottest and trending research topic. A graph database is preferred to deal with the dynamic and complex relationships in connected data and offer better results. Every data element is represented as a node. For example, in social media site, a person is represented as a node, and its properties name, age, likes, and dislikes, etc and the nodes are connected with the relationships via edges. Use of graph database is expected to be beneficial in business, and social networking sites that generate huge unstructured data as that Big Data requires proper and efficient computational techniques to handle with. This paper reviews the existing graph data computational techniques and the research work, to offer the future research line up in graph database management.

Corresponding Author:

Patil N. S.,
Information Science & Engineering,
BIET,
Davangere, Karnataka, India.
Email: nsp4041@gmail.com

1. INTRODUCTION

Today, the user data is increasing rapidly due to many data generating processes like latest social media networks, rapid adaptation of smartphones and handheld devices further enhances the data creation. The computation of this data is becoming more difficult day by day, as the users of the digital data and networks are increasing by manifolds [1]. Traditional databases cannot compute this huge data without complexity for the real-time responses; whereas, in the case of graph databases, a graph is generated for each entity, which speeds up the process. The use case for a graph database scenario is content-based data filtering. Graph database provides better performance and data consistency; hence many researchers are considering the graph models [2].

In order to handle the issues of storing huge data, many of the researchers have presented the concept of graph and graph storage, in which the graphs are implied to model the huge data with complicated design. In every graph, there will be nodes, properties, and edges as the relationship among them. The connected data graph database also offers the significant choice to deal with the structured, semi-structure and unstructured data [3]. The graph database offers the fastest response to a query, many times, in milliseconds. Today, the graph databases are widely used in retail, social network, healthcare, communication and other online solutions. Operations like create, update, read and delete are available in graph database system. The drawback of these systems is that it is more expensive by nature than the traditional methods [4].

This survey paper discusses the concepts of graph databases, review of the existing research regarding existing computational techniques of data management. Section 2 discusses some basic conceptual aspects of graph databases, modeling, computational techniques and comparisons of techniques. Section 3 provides the literature review of recent research work in graph database management, Graph database computational techniques. Section 4 provides the research gap in recent research work of Graph Databases. Section 5 describes the future research lineup, and finally, Section 6 concludes the paper.

2.   GARAPH DATA

In recent years the way Internet and mobile communication have been used for different and varied needs and applications by a common user, academicians, researchers have been started rethinking as for how to store the huge data which is being generated every day, every hour and every minute. This need for the storage and retrieval of data and information brought back the concepts of graph and graph models [4], [5].

Graphs are used to model complicated structures. The graph is a collection of nodes, edges, and the relationships between them. In the graph, nodes are called entities, and there are many ways in which these entities are co-related in a different type of applications. The connection between these entities is called as a relationship. In graphs, data term "Attributes" related to entities and relationships are called labels. In a graph like structure, data is stored into nodes, and these nodes have some properties. In graphs, relationships consist of properties and connect one node to the other node.

The example shown in Figure 1, demonstrates the relationship between the two animals. In the above figure, two things are identified that Thing-1 and Thing-2, exhibit properties like animal type, name (cat & dog) and relationship. The representation says that the Thing-1 and Thing-2 are dog and cat respectively and are named as cute and handsome respectively. Finally, both dog and cat relationship is mentioned due to its animal category.
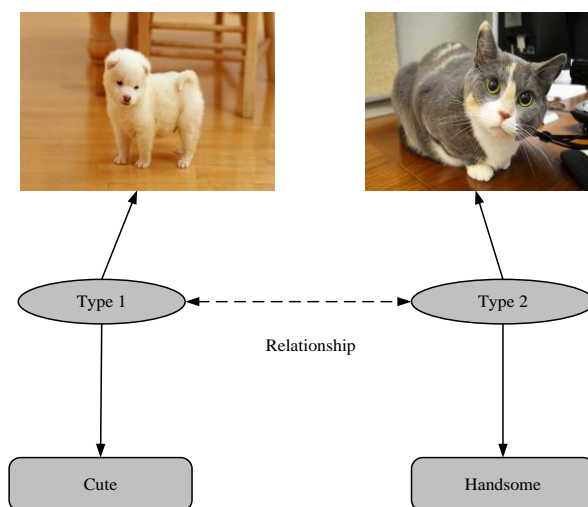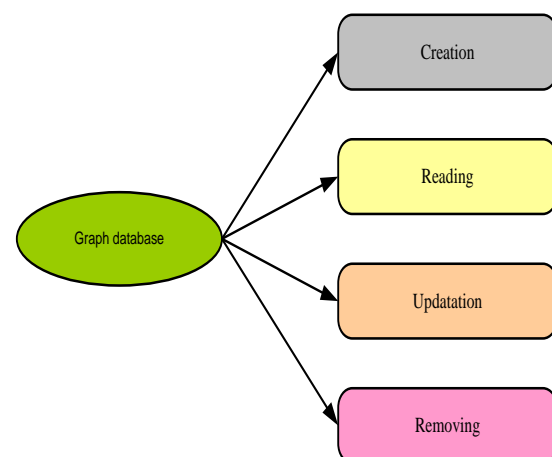

Figure 1. Example of Graph data   Figure 2. Units of graph database system

2.1. Graph Databases

The Graph database system has four different units such as creation, reading, updatation and removing, which can be used in designing of graph data model. The Index free square matrix of finite graph representation is more necessary to get the high-performance graph traversal. Graph database utilizes the square matrix or adjacency then each node manages the direct relationship with the adjacent nodes. The graph database exhibit a single data structure known as a graph, and it has no combined operation, and hence each edge will be connected to another edge. The graph will store the data in nodes having a relationship. The data in the graph database will follow the model graph property. Graph-oriented database is specialized No SQL, where relationships among the nodes are stored and managed generically. Built-in support for relations makes the traversal much faster for multidimensional, interconnected datasets; hence suitable for online transaction processing (OLTP). For the same reason, organic and ground-up products like neo4j offer multifold performance benefits in comparison to multilayer abstractions over the traditional technologies like relational databases (RDB), and object-oriented databases (OODB). It also simplifies the complexity of design and

implementation; popular notation is "if you can see whiteboard, you can graph." Being a high-level abstraction to the network model database, it reduced the coding effort to one-tenth; it's a key technology used in rapid application development (RAD) [5], [6].

Graph databases are quickly making inroads into real life from research laboratories; many social networking enterprises like Twitter, Facebook, and Google have already adopted years ago. Recently the technology- not only the scientific data but also the web and many different kinds of data can be modeled as a graph. This helps to overcome the limitations of RDBMS, like predefined schema and to process complex queries in milliseconds. Especially, lack of schema allows developers to gain high productivity, besides providing the capability to process complex multilevel queries in real-time. E-commerce sites and users benefit from the easy processing of the recommended product. Machine learning algorithms are utilized the most, for the applications such as these where big data analytics is used by global top 100 companies. Bug Localization is another application area worth mentioning for the use of graph databases. Overall there is a variety of domains where graph data modeling can be applied to revolutionize the user experience [7].

2.2. Existing types of Graph Database Models

In recent past, many tools are developed using the graph database concept, for example, Neo4J and Sparksee [8]. The tools like Oracle spatial and graph, OQGraph, and ArangoDB are designed as an abstraction with the underlying architecture of relational databases MySql, Oracle [9]. Until now, there is no industry standard [10], and moreover many of them are designed to be suited for a particular domain [11]. In the case of In-memory model, scalability is limited as the memory holds the content [12]. Another reason for inefficiency into the model is due to horizontal scaling and layering mechanism. The requirement of the new paradigm is for handling an extensive data; very few models are designed to adopt parallelism as well OLTP. The early models lack standard query language, Application programming interface, and protocols as found in conventional models such as SQL, JDBC, and REST. Lately, Gremlin and SPARQL are gaining consensus, but the adaptation is too slow.

2.3. Neo4j (Neo Technology)

Neo4j is a disk-based transactional graph database and named as "World leading graph database." Its first release date was in 2007. Neo4j also supports another language like Python except for Java for graph operations. Neo4j is an open source project [7] available in a GPLv3 Community edition, with Advanced and Enterprise editions available under both the AGPLv3 as well as a commercial license. Neo4j is best graph database for enterprise deployment. It scales to billions of nodes and relationships in a network. Neo4j manages all the operations that modify data in a transaction. In Neo4j both nodes and relationship can contain properties. Neo4j is a graph database that manages graphs and is optimized for graph structure instead of tables. It is the more expressive type of graph database is similar to other graph databases. Neo4j is most popular graph databases today [8].

2.4. Hyper Graph DB

It is an open-source database supports hype graphs. Hyper graph [8] is different from the normal graph because in this edge is points to the other edges. In various fields, it is used in the modeling of the graph data. It supports online querying with an API written in Java. It is based on the Hyper Graph DB model. It is a universal data model highly complex and large-scale knowledge application. It has graph-oriented storage and customizable indexing. In this graph database, a hyper edge is easy to convert into a tuple. It is a distributed and graph-oriented database [8-14].

2.5. DEX

DEX [15] is said to be very efficient and bitmaps-based graph database and is written in C++ language. It was first released in 2008. It makes graph querying possible in different networks like social network analysis and pattern recognition. It is also known as high-performance graph database in the case of large graphs and useful for most of the NoSQL applications. The latest version of DEX supports both Java and.NET programming. It's portable and requires only a single JAR file for execution. DEX is called the fourth most popular graph database today [3], [6].

2.6. Trinity

Trinity is a distributed graph system [9] over a memory cloud. Memory Cloud is globally addressable in memory key-value store over a cluster of the machine. It provides fast data access power when we have large datasets. It is a large graph processing machine. It provides fast graph exploration and parallel computing for larger datasets. It also provides high throughput on large graphs which have a billion nodes.

2.7. Infinite Graph (Objectivity)

Infinite Graph is produced by an organization called Objectivity. It is a type of company that works to develops database technologies supporting large-scale, object persistence and relationship analytics. An infinite graph database is a distributed graph database in Java, and it is based on a graph like structure. We can call infinite graph as a cloud-enabled graph database. It is designed for to handle the very high throughput. It is a single graph database distributed across multiple machines. There is a lock server which handles lock requests from database applications. It is capable of dealing with complex relationship requiring multiple hops. It provides graph-wise indexes on multiple key fields and also provides high performance regarding query [7], [10].

2.8. Titan

Titan [9] was adopted in 2012. It is written in Java and an open source project. The main benefit of using Titan is its scaling feature. It also provides support to very large graphs and scales with the number of machines in a cluster. It is also highly scalable graph database regarding concurrent users and size of the graph. It provides a batch graph processing with Hadoop framework and also gives answers to complex queries in milliseconds. It consists of three main components:

    a. Native Blueprints Implementation
    b. Gremlin Query language
    c. Rexster Server

It follows property graph model and supports Gremlin: a graph traversal query language. It also offers an optimized disk representation for efficient use of storage and speed of accessing data. Applications can interact with Titan in mainly two ways:

    a. First Method is that calls Java-language API's related to Titan which includes its native API implementation.
    b. TinkerPop stack utilities such as Gremlin query language built atop Blueprints.

Recent research survey
The research in the domain of graph data is classified into ten different categories by considering IEEE Xplore journals. The categorization is given as below.

2.9. Data Store Efficiency

In order to bring the better efficiency of the data storage, some of the issues may exist among these data compression is necessary to store more data. Also, the data standardization may play a greater role to map the data and translate for cloud storage. In the large, the super graph search is required to choose the data graphs features. The recent work done in this category Table 1 below.

Table 1. Work for data store efficiency

Author	Issue considered	Method adopted	Result
Sutrisna et al. [4]	Data compression	Graph clustering algorithm with data set of collaboration data among journal writers	Lossless compressed graph data is achieved
Bansal et al. [5]	Data standardization and classification	online compression algorithm	Reduction in number of nodes and achieved graph database compression for less storage space
Lyu et al. [6]	Super graph search	indexing and query processing algorithms , CCD dataset and NCI dataset	Indexing and processing time
Chen and Chen [7]	Supporting reach ability queries	Decomposition of graphs	High efficiency, effectiveness and querying time

2.10. Database Indexing Method

In order to facilitate isomorphism and similarity queries and building efficient graph database systems and accelerate graph similarity search, much significant works is performed. The following Table 2 is the some of the chosen work in database indexing method.

Table 2. Work for database indexing method

Author	Issue considered	Method adopted	Result
Williams et al. [8]	Graph storage, similarity and indexing	Sub-graph isomorphism and similarity queries mechanism by using Protein motifs and synthetic datasets	Achieved improved query times for sub-graph isomorphism queries.
Luo et al. [9]	Large scale graph database indexing and search approach	T-mixture model (Combination of optimized vector quantizer and probabilistic approximate ate based indexing scheme)	Robust in handling outliers.
Yuan et al. [10]	Graph feature mining	Query grouping mechanism	Achieved better, faster and light weight filtering,
Bei et al. [11]	Graph search	Distributed graph searching mechanism	Achieved distributed graph database
Goldberg et al. [12]	Problem of fragment identification	Heuristic mechanism	Achieves optimized running time

2.11. Graph Indexing Method

The improvisation of graph data modeling was done for different data. The following Table 3 briefs the work performed for graph data modeling and graph based management system.

Table 3. Work for graph indexing method

Author	Issue considered	Method adopted	Result
Dongoran et al. [13]	Data modeling	Index construction, database filtering, sub-graph matching	Achieved more path length, more indexing time
Kang et al. [14]	Dynamic graphs storage and manage	Graph based database management system	Robust in handling outliers.

2.12. Sub-graph matching method

This part briefs some research ideas presented by many researchers in data querying, sub-graph matching, etc. The recent works for better sub-graph matching are presented. Also, the works shown in the following Table 4 gives the ideas about various graph data techniques.

Table 4. Work for sub-graph matching method

Author	Issue considered	Method adopted	Result
Giugno Shasha [15]	Graph querying	Regular expression graph query language that combines Xpath and Smart; hash-based finger-printing	performs well for small query graphs on large graph databases (in the thousands
Bröcheler et al. [16]	Sub-graph Matching	probabilistic method to estimate probabilities; Partition algorithm for creating index	Works efficiently, answering 778M edge real-world SN in under one second.
Bröcheler et al. [17]	Approximate sub-graph Matching	PMATCH algorithm	Efficient and scales to over a billion edges.
Bröcheler et al. [18]	Sub-graph Matching; long-tailed degree distributions	delicious social book-marking service	Faster than static cost models for warm caches.
Hong et al. [19]	Set similarity	Set similarity pruning and structure-based pruning; dominating-set-based sub-graph matching; inverted pattern lattice and structural signature buckets are designed	outperforms state-of-the-art methods by an order of magnitude
Hoksza and Jelínek [20]	protein-protein interface (PPI) identification	knowledge-based approach Using Neo4j for mining protein graphs	in comparison to Microsoft SQL Server, Neo4j is a viable option for small, sub-graph query types

2.13. Semantic

The recent works that are addressed the semantic approach, Query semantic data processing; data analysis in a graph database is given Table 5.

Table 5. Work for graph database semantic

Kivikangas and Ishizuka [21]	Semantic Queries	Utilizing Universal Words (UWs). Concept Description Language (CDL) for semantic data instead of RDF, and Neo4j	Improved Semantic Queries
Kalmegh and Navathe [22]	high-performance graph databases	Survey	Define, Characteristics, Future directions
Irawan and Prihatmanto [23]	Implementation of Graph Database for OpenCog	GraphBacking Store API extends Backing Store C++ API	Atom Space represents knowledge in a hyper graph structure, persisting in a graph database is more intuitive and more portable
Graves et al. [24]	Design of data store for genome	Review all available options and compare	Graph database
Cesare et al. [25]	Automated taxonomy extraction from semantic process models	linguistic approach based on semantic similarity	generalize elements of business processes that are automatically discovered from semi-structured data projects
Morari et al. [26]	Scaling Semantic Graph Databases	SPARQLTO-C++ COMPILER, A Library Of Distributed Data system, And A Custom Multithreaded Runtime	Better scaling
Wardani and Kiing [27]	Semantic Mapping Relational to Graph Model	create property relationship in the result of the mapping and converting process	map and convert the relational data model to graph model without semantic loss
Souza et al. [28]	Graph Databases with Semantic Network Models for Software-Defined Networking (SDN) Applications	Import the Network Markup Language (NML) model	modeling tasks are considerably more natural compared to RDBM; to reproduce SDN application primitives
Hayakawa and Nishiyama [29]	Query Processing of Semantic Data	Summarized graph in advance	improve the query performance by 6.62 times
Bednar et al. [30]	theoretical analysis and performance testing	Benchmark data modeling and population	Comparative benefits& weakness
Hartley et al. [31]	storing, Accessing and analyzing Trillion vertices and edges.	semantic graphs execution of parallel out-of-core graph algorithms	outperforms widely used open-source
Cavoto et al. [32]	Network-Driven Data Analysis	FishBase global information system	discovery of new information and validating the existing data
Caldarola et al. [33]	Big Graph-based Data Visualization	The WordNet	efficiency, effectiveness and clearness; two different representation of Word-Net
Lamhaddab and Elbaamrani [34]	Graph Modeling For Mobiles	implementation of an extractor module (in java language)	Reverse engineering from iOS platform to Android platform
Leida and. Chu [35]	Distributed SPARQL query answering over RDF data streams	Business Process Monitoring domain for Query workload balancing	Approach for efficient and scalable query processing over RDF graphs distributed over a local data grid.
Mordinyi et al. [36]	efficient data store that is capable of versioning and querying local and common concepts	NoSQL graph database	outperforms ontology stores and match solutions relying on relational databases
Balboni et al. [37]	Evolution Analysis	natural language processing engines to build temporal graph database	Got large amount of open source documents
Wu and Chen [38]	Frequent Sub-graph Mining	By normalizing the incidence matrix	Achieved higher speed and efficiency
John et al. [39]	Learning process enhancement against population	Natural Language Processing	enhanced learner centered online learning experience
Xu and Luo [40]	Expression-Driven Sketch Graph Matching for Face Recognition	multi-layer grammatical face model	recognition rates were improved, especially for the smiling and screaming faces whose line-edge maps are greatly distorted
Figueira and Libkin [41]	Querying Graphs	Parikh automata	*real-life querying*

2.14. Social Networking

The recent ideas towards the graph data generated by social networks are presented in the following Table 6.

Table 6. Work for social networking graph database

Author	Issue considered	Method adopted	Result
Dayarathna and Suzumura [42]	Graph Database for Hybrid Clouds	Distributed Graph Database	Achieves faster performance
Soussi [43]	Social Network Extraction	Graph Database	Achieves better extraction
Yar and Tun [44]	Searching Personnel Relationship	Graph Database	relationship among persons can simply and accurately be inferred
Mir and Wright [45]	Differentially Private Estimator	Kronecker graph model	Generate synthetic graphs that are "similar" to the original target graphs in a privacy preserving manner.
Shrivastava and Pal [46]	Graph Mining	Graph data processing, extraction and visualization	Achieved quality in processing

2   RESEARCH GAP IN THE EXISTING WORK

The research work that performed in recent years is lacking with effectiveness in a huge unstructured graph database. The work of the various authors lacks the research efficiency in their methods. The applicability of these methods in real time applications shows poor performance. There is a need for proper research in a huge an unstructured graph database management. The following are the listed research gaps in the existing work:

   a. The existing researches are doesn't support the huge unstructured graph data induces complexity of technique and some of the performance metrics are need to be explored in graph data.
   b. The existing researches doesn't fulfill the challenges of the hosting graph database like dynamic nature of graph database volume, tough to maintain the graph data and consumes higher computational time for evaluation of graph queries.
   c. The graph has partitioning issue, information loss and unnecessary computation are still unaddressed and are need to address for unstructured graph database.
   d. The researches which are mentioned in past are not designed for data mining of graph database.
   e. The existing privacy preservation techniques using data anonymity approaches are not efficient as this approach doesn't provide theoretical evidence that the provided solution is effective against the security issues.

3   LINE OF RESEARCH IN FUTURE

The better graph database scalability and management of huge unstructured graph database can be attained as below steps.

   a. Outlining the existing issues in current work.
   b. A prototype is needs to be designed to generate large and unstructured data for real-time applications and also implement the respective graph theory to model the graph database.
   c. A novel data mining algorithm is needed to be developed for the graph database having low computational complexity.
   d. A cost effective mechanism is needed to be build, and it should offer privacy for huge unstructured data.
   e. The effectiveness of the mechanism is needed to tally with the existing work.

4   CONCLUSION

This survey paper gives the better idea of required attributes to manage the huge unstructured graph database. This paper gives the different types graph database like Neo4j, DEX, and Titan, etc. The survey of recent work towards graph data is collected studied and represented in various sections like Data store efficiency, Database Indexing Method, Graph Indexing Method, Sub-graph matching method, Semantic and social networking graph database. From the above existing research survey, a research gap is defined, and flowingly the research ideas for improving the graph database are presented.

REFERENCES

[1] Nirwansyah, Ferdy, and Suharjito Suharjito, "Hybrid Disk Drive Configuration on Database Server Virtualization," *Indonesian Journal of Electrical Engineering and Computer Science* 2.3 (2016): 720-728.
[2] Ismaeel, Salam, Ayman Al-Khazraji, and Karama Al-delimi, "Fuzzy Information Modeling in a Database System," *International Journal of Artificial Intelligence (IJ-AI)* 6.1 (2017): 1-7.

[3] Laday, Ravie Kurnia, Heru Sukoco, and Yani Nurhadryani, "Distributed System and Multimaster Replication Model on Reliability Optimation Database," *Indonesian Journal of Electrical Engineering and Computer Science* 13.3 (2015): 529-536.

[4] B. A. Sutrisna, W. K. RahmatSaleh and A. A. Gozali, *"Implementation of GRAC algorithm (Graph Algorithm Clustering) in graph database compression,"* Information and Communication Technology (ICoICT), 2015 3rd International Conference on, Nusa Dua, 2015, pp. 391-395.

[5] Bansel, H. González-Vélez and A. E. Chis, *"Cloud-Based NoSQL Data Migration,"* 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Heraklion, 2016, pp. 224-231.

[6] B. Lyu, L. Qin, X. Lin, L. Chang and J. X. Yu, *"Scalable supergraph search in large graph databases,"* 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, 2016, pp. 157-168.

[7] Y. Chen and Y. Chen, *"Decomposing DAGs into spanning trees: A new way to compress transitive closures,"* 2011 IEEE 27th International Conference on Data Engineering, Hannover, 2011, pp. 1007-1018.

[8] D. W. Williams, J. Huan and W. Wang, *"Graph Database Indexing Using Structured Graph Decomposition,"* 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, 2007, pp. 976-985.

[9] Luo, A. Zheng, J. Tang and H. Zhao, *"Large-Scale Graph Database Indexing Based on T-mixture Model and ICA,"* Image and Graphics, 2007. ICIG 2007. Fourth International Conference on, Sichuan, 2007, pp. 815-820.

[10] Yuan, P. Mitra, H. Yu and C. L. Giles, *"Iterative Graph Feature Mining for Graph Indexing,"* 2012 IEEE 28th International Conference on Data Engineering, Washington, DC, 2012, pp. 198-209.

[11] Y. Bei, Z. Lin, C. Zhao and X. Zhu, *"HBase System-Based Distributed Framework for Searching Large Graph Databases,"* Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference on, Honolulu, HI, 2013, pp. 151-156.

[12] M. Goldberg, J. Greenman, B. Gutting, M. Magdon-Ismail, J. Schwartz and W. Wallace, *"Graph search beyond text: Relational searches in semantic hyperlinked data,"* Intelligence and Security Informatics (ISI), 2012 IEEE International Conference on, Arlington, VA, 2012, pp. 153-155.

[13] S. S. Dongoran, W. K. Rahmat Saleh and A. A. Gozali, *"Analysis and implementation of graph indexing for graph database using GraphGrep algorithm,"* Information and Communication Technology (ICoICT), 2015 3rd International Conference on, Nusa Dua, 2015, pp. 59-64.

[14] J. M. Kang, H. Bannazadeh and A. Leon-Garcia, *"SDIGraph: Graph-based management for converged heterogeneous resources in SDI,"* 2016 IEEE NetSoft Conference and Workshops (NetSoft), Seoul, 2016, pp. 88-92.

[15] R. Giugno and D. Shasha, *"GraphGrep: A fast and universal method for querying graphs,"* Pattern Recognition, 2002. Proceedings. 16th International Conference on, 2002, pp. 112-115 vol.2.

[16] M. Bröcheler, A. Pugliese and V. S. Subrahmanian, *"COSI: Cloud Oriented Subgraph Identification in Massive Social Networks,"* Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on, Odense, 2010, pp. 248-255.

[17] M. Brocheler, A. Pugliese and V. S. Subrahmanian, *"Probabilistic Subgraph Matching on Huge Social Networks,"* Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on, Kaohsiung, 2011, pp. 271-278.

[18] M. Bröcheler, A. Pugliese and V. S. Subrahmanian, *"A budget-based algorithm for efficient subgraph matching on Huge Networks,"* Data Engineering Workshops (ICDEW), 2011 IEEE 27th International Conference on, Hannover, 2011, pp. 94-99.

[19] L. Hong, L. Zou, X. Lian and P. S. Yu, "Subgraph Matching with Set Similarity in a Large Graph Database," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2507-2521, Sept. 1 2015.

[20] Hoksza and J. Jelínek, *"Using Neo4j for Mining Protein Graphs: A Case Study,"* 2015 26th International Workshop on Database and Expert Systems Applications (DEXA), Valencia, 2015, pp. 230-234.

[21] P. Kivikangas and M. Ishizuka, *"Improving Semantic Queries by Utilizing UNL Ontology and a Graph Database,"* Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on, Palermo, 2012, pp. 83-86.

[22] P. Kalmegh and S. B. Navathe, "Graph Database Design Challenges Using HPC Platforms," High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion: Salt Lake City, UT, 2012, pp. 1306-1309.

[23] H. Irawan and A. S. Prihatmanto, *"Implementation of graph database for OpenCog artificial general intelligence framework using Neo4j,"* 2015 4th International Conference on Interactive Digital Media (ICIDM), Bandung, 2015, pp. 1-6.

[24] M. Graves, E. R. Bergeman and C. B. Lawrence, "Graph database systems," in *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 6, pp. 737-745, Nov/Dec 1995.

[25] S. d. Cesare, D. Juric and M. Lycett, *"Automated Taxonomy Extraction from Semantic Business Process Models,"* 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, 2016, pp. 4394-4403.

[26] Morari et al., "Scaling Semantic Graph Databases in Size and Performance," in *IEEE Micro*, vol. 34, no. 4, pp. 16-26, July-Aug. 2014.

[27] D. W. Wardani and J. Kiing, *"Semantic mapping relational to graph model,"* Computer, Control, Informatics and Its Applications (IC3INA), 2014 International Conference on, Bandung, 2014, pp. 160-165.

[28] T. D. P. C. D. Souza, C. E. Rothenberg, M. A. S. Santos and L. B. D. Paula, *"Towards Semantic Network Models via Graph Databases for SDN Applications,"* 2015 Fourth European Workshop on Software Defined Networks, Bilbao, 2015, pp. 49-54.

[29] Hayakawa and H. Nishiyama, *"Efficient Query Processing of Semantic Data Using Graph Contraction on RDBMS,"* Signal-Image Technology & Internet-Based Systems (SITIS), 2013 International Conference on, Kyoto, 2013, pp. 958-965.

[30] P. Bednar, M. Sarnovsky and V. Demko, *"RDF vs. NoSQL databases for the semantic web applications,"* Applied Machine Intelligence and Informatics (SAMI), 2014 IEEE 12th International Symposium on, Herl'any, 2014, pp. 361-364.

[31] T. D. R. Hartley, U. Catalyurek, F. Ozguner, A. Yoo, S. Kohn and K. Henderson, *"MSSG: A Framework for Massive-Scale Semantic Graphs,"* 2006 IEEE International Conference on Cluster Computing, Barcelona, 2006, pp. 1-10.

[32] P. Cavoto, V. Cardoso, R. V. Lebbe and A. Santanchè, *"FishGraph: A Network-Driven Data Analysis,"* e-Science (e-Science), 2015 IEEE 11th International Conference on, Munich, 2015, pp. 177-186.

[33] G. Caldarola, A. Picariello and A. M. Rinaldi, *"Big graph-based Data visualization experiences: The WordNet case study,"* 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), Lisbon, Portugal, 2015, pp. 104-115.

[34] K. Lamhaddab and K. Elbaamrani, *"Model driven reverse engineering: Graph modeling for mobiles platforms,"* 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA), Marrakech, 2015, pp. 392-397.

[35] M. Leida and A. Chu, *"Distributed SPARQL Query Answering over RDF Data Streams,"* 2013 IEEE International Congress on Big Data, Santa Clara, CA, 2013, pp. 369-378.

[36] R. Mordinyi, P. Schindler and S. Biffl, *"Evaluation of NoSQL graph databases for querying and versioning of engineering data in multi-disciplinary engineering environments,"* 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, 2015, pp. 1-8.

[37] A. Balboni, M. Marchetti, M. Colajanni and A. Melegari, *"Supporting sense-making and decision-making through time evolution analysis of open sources,"* Cyber Conflict: Architectures in Cyberspace (CyCon), 2015 7th International Conference on, Tallinn, 2015, pp. 185-202.

[38] J. Wu and L. Chen, *"A Fast Frequent Subgraph Mining Algorithm,"* Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for, Hunan, 2008, pp. 82-87.

[39] B. John, V. Thavavel, J. Jayakumar, A. Muthukumar and K. J. Poornaselvan, *"Enhanced learner centered pedagogical strategy: Promoting STEM with underrepresented populations,"* Integrated STEM Education Conference (ISEC), 2014 IEEE, Princeton, NJ, 2014, pp. 1-4.

[40] Zijian Xu and Jiebo Luo, *"Face Recognition by Expression-Driven Sketch Graph Matching,"* 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, 2006, pp. 1119-1122.

[41] D. Figueira and L. Libkin, *"Path Logics for Querying Graphs: Combining Expressiveness and Efficiency,"* Logic in Computer Science (LICS), 2015 30th Annual ACM/IEEE Symposium on, Kyoto, 2015, pp. 329-340.

[42]   M. Dayarathna and T. Suzumura, *"Towards Scalable Distributed Graph Database Engine for Hybrid Clouds,"* Data-Intensive Computing in the Clouds (DataCloud), 2014 5th International Workshop on, New Orleans, LA, 2014, pp. 1-8.

[43] R. Soussi, M. A. Aufaure and H. Baazaoui, *"Towards Social Network Extraction Using a Graph Database,"* Advances in Databases Knowledge and Data Applications (DBKDA), 2010 Second International Conference on, Menuires, 2010, pp. 28-34.

[44] Kay Thi Yar and Khin Mar Lar Tun, *"Searching Personnel Relationship from Myanmar census data using Graph database and Deductive Reasoning prolog rules,"* 2016 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2016, pp. 1-7.

[45] D. J. Mir and R. N. Wright, *"A Differentially Private Graph Estimator,"* 2009 IEEE International Conference on Data Mining Workshops, Miami, FL, 2009, pp. 122-129.

[46] S. Shrivastava and S. N. Pal, *"Graph Mining Framework for Finding and Visualizing Substructures Using Graph Database,"* Social Network Analysis and Mining, 2009. ASONAM '09. International Conference on Advances in, Athens, 2009, pp. 379-380.

BIOGRAPIES OF AUTHORS

Patil N S, working as Assistant Professor Department of Information Science, BIET, Davangere. My teaching experience is 7yrs area of interest Database management system, Datamining, Graph Datamining, Big Data.

Dr Kiran P, working as Associate Professor Department Computer science, R. N. S Institute of Technology, Bangalore. His teaching experience 12yrs Area of interest Cryptography, Randomization methods & Anonymization methods in generalization, Indexing techniques and Design patterns. He has published 12 papers.