GRAPH DATA BASE: AN ENABLING TECHNOLOGY FOR DRUG PRESCRIPTION PATTERNS ANALYSIS

Ilaria Giordani[1], Francesco Archetti
Department of Computer Science, Systems and Communication, University of Milano Bicocca, Milano, Italy

Antonio Candelieri
Department of Economics, Management and Statistics, University of Milano Bicocca, Milano, Italy

Gaia Arosio
Consorzio Milano Ricerche, Milano, Italy

Roberto Mattina
Department of Biomedical, Surgical and Dental Sciences, University of Milano, Milano, Italy

Abstract. *This paper has two main objectives: first to show that new data base technologies (DB) like graph data bases can enable the efficient design and implementation of network -based models, second that this type of models enables new insights on biomedical data and in particular prescription patterns allowing to link data about patients, prescriptions and prescriber. Albeit the application domain is potentially the whole field of health care data, the focus of this paper is on prescription patterns and specifically of antibiotics whose prescription pattern is difficult to analyze due to the antibiotics resistance. This problem can take advantage of the approach proposed: a network-based model, specifically suitable for community-based medicine, which is a suitable framework for antibiotics prescription and resistance analysis.*

Keywords*: Graph databases, network analytics, antibiotics resistance, cluster analysis, exploratory analysis.*

1. INTRODUCTION

Prescription pattern monitoring studies (PPMS) exploit medical information to improve the prescribing practices and thus the standards of medical treatments at all levels of healthcare (Prescott et al., 2018).

Data analytics already gives a major contribution to PPMS, tackling health and socioeconomic challenges: by means of insights originated from analytical results health authorities can plan informed actions, promote appropriate use and

[1] Ilaria Giordani, email: ilaria.giordani@unimib.it

prescription, compare drug consumption patterns within specified time ranges, focus on specific products and defined geographical areas and reduce the abuse/ misuse of monitored drugs.

One of the main contributions of this paper is to use network analytics to uncover novel insights from PPMS analytical results. This has been made possible by methodological advances and by a new generation of graph-based databases. This offer not only big data capabilities, but a huge number of native algorithms for network analysis and visualization.

2. THE ANALYTICS FRAMEWORK

The analytics framework proposed in this paper is composed of two main modules: exploratory analysis and network analytics.

Exploratory analysis is specifically focussed on the doctor-patient relationship: person-centred care concerning diagnoses and prescriptions, analysing their changes according to habits, physiological aspects, time and geographical area of both interested parts. After collecting the first batch of results and checking them with domain experts, information with unusual patterns is outlined, and further examination is made on a subset of features.

Network analytics provides a very powerful model to analyse connected data because its algorithms focus on the relationships between nodes to infer the organization and dynamics of complex systems. Also, simple network characterization like degree and centrality indexes are used to identify communities of similar nodes and their dynamics. Community identification is common in all types of networks and its characterization can uncover structures like hubs and hierarchies, find nested relationships and infer similar behaviours.

Defined timeframes in limited temporal windows are selected to make a detailed trajectory analysis related to prescriptive appropriateness compared to antibiotic resistance, without going into details of pathologies.

3. DATA DESCRIPTION

The database contains the medical history of patients using healthcare services in one Italy region: it is composed of pseudonymised data by encryption before acquisition and is related to eighteen years data about outpatient visits at GPs' ambulatories located in several Health Districts. The database comprehends data on medical histories of patients between January 2000 and October 2018.

Global inferences have been made using the entire dataset, while the need of detailed recent reports leads to the decision of using a limited range of years for analysing prescription changes and patient journey. Following an overview of the principal characteristics of the complete database: 1356 general practitioners; around 1 million patients, around 15,000,000 diagnoses; around 119,000,000 prescriptions (information about therapies and prescribed medicines).

3.1 THE DATABASE MODEL

In Fig. 1 is reported the database model to provide an immediate comprehension of how entities are related and visualize information to identify keys (primary keys -PK- and foreign keys -FK-) and unique fields, essential for data joining.

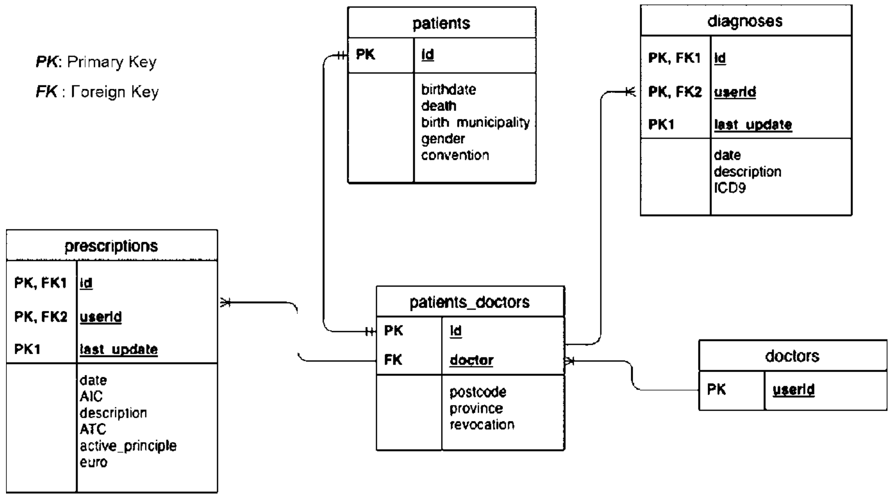

Fig. 1: Entity - Relation model of the database

The data model is built on 4 tables, which are briefly described in the following. Each diagnosis and prescription is uniquely distinguished by the triplet {patient, doctor, date}.

The table *patients* includes information about patients, to ensure privacy dealing with sensitive data, there are no full names. The table *patients_doctors* contains information similar to patients, with additional fields focussing on their relationship with the general practitioners, which are essential to link and analyse data. The table *diagnoses* comprehends the diagnoses associated to patients and relative GPs. Each diagnosis is defined by its ICD-9 code, an international identifier for diseases. The table *prescriptions* contains the prescribed medicines for each patient. There is no linkage between diagnoses and prescriptions in the database, therefore additional work is required to detect correlations. Each prescription is defined by an ATC code (Anatomical Therapeutical Chemical) and active principle code (AIC).

4. EXPLORATORY ANALYSIS

4.1 VARIATIONS THROUGH TIME

To give a general idea on variations of available information and magnitude orders, a snapshot of 2010 - 2017 has been used to show patterns and differences: as shown in Table 1, patients getting diagnoses tend to remain stable, yet prescriptions have a difference of more than 450,000 records, confirming the hypothesis of an increase in registering information while raising concerns of a general overprescribing.

Tab. 1: Number of patients with diagnosis (D.) and prescriptions (P.)

	2010	2011	2012	2013	2014	2015	2016	2017
Age mean	47.43	47.89	48.44	48.68	48.98	49.78	50.12	50.54
D.	329,715	320,253	316,431	320,948	324,920	323,441	330,641	326,987
P.	7,326,923	7,108,288	7,268 855	7,541,539	7,777,753	7,847,313	7,856,246	7,785,325

4.2 MOST FREQUENT DISEASES

Selection of a subset of diseases is possible after preliminary analysis, of which the most immediate is concerned with. There are 9895 distinct ICD-9 in the diagnoses table: of those, the 10 most popular ones in terms of counts have been selected for further analysis. The most common ICD-9 (other unknown or unspecified cause of morbidity and mortality) corresponds to a generic unspecified disease, therefore, cannot be subject of detailed analysis.

4.2.1 MOST FREQUENT DISEASES BASED ON AGE

Since some diseases are largely age dependent, patients are grouped into ranges and the most common ICD-9 are counted based on the difference between birthdate and prescription date. Analytics regarding gender must be performed choosing a different batch of illnesses, taking the ones which are proven to affect one gender rather than the other.

4.3 DRUG PRESCRIPTION ANALYSIS: THE ANTIBIOTICS CASE

Antibiotics, also known as antibacterial, are medications produced by microorganisms that destroy or slow down the growth of bacteria. They include a range of powerful drugs and are used to treat diseases caused by bacteria.

A general practitioner can prescribe a broad-spectrum antibiotic to treat a wide range of infections. A narrow-spectrum antibiotic is only effective against a few types of bacteria. In the database we have analysed there are 794,267 patients with at least one antibiotic prescription in the whole-time range (2000 - 2018). The following analysis is performed on a set composed by 8,386,057 prescriptions with an AIC corresponding to an antibiotic. The average number of prescriptions (670,634) has been calculated only considering the number of patients with at least one antibiotic prescription from 2008 to 2017.

5. NETWORK ANALYSIS

This section is devoted to an analysis of antibiotic prescription patterns through a graph database, applying graph algorithms to identify similarities between prescriptions and popularity among prescriptions (Cavallo et al., 2013; Nagarajan and Talbert, 2019).

Goal of the graph analysis is completion of antibiotic patterns changes and patient journey, providing a different point of view on those two important aspects altogether. The main focuses are: understanding whether specified couples of drugs are often pre-scribed together (co-prescription); highlight particularly important entities in the graph (centrality measures of nodes); identify similar kinds of doctors according to their prescription history (community clustering).

5.1 GRAPH DATABASES

Graph databases are a particular type of NOSQL data management systems allowing persistent representation of entity and relationship in a graph structure, implementing the Property Graph Model effciently down to the storage level (Park et al., 2014). Activity around graph databases flourished in the first half of the nineties and then the topic almost disappeared (Angles, 2012).

Recently, the need to manage information with graph-like nature has re-established the relevance of this databases in different domains in which the importance of the information relies on the relations at least as much as on the entities for example chemistry, biology, semantic Web). Moreover, the explosion of massive and complex graph-like knowledge and information structures make a graph database a crucial requirement. This new relevance has brought on the market several graph databases systems.

In graph database, both data and schema are modelled as graphs, or as data structures generalizing the notion of graph (hypergraphs or hypernodes) and data manipulation is expressed by graph-oriented operations whose main primitives are on graph features and graph statistics (diameter, centrality, etc.).

The benefits of using a graph data model are given by: the introduction of a level of abstraction which allows a more natural modelling of graph data; query languages and operators for querying directly the graph structure and ad-hoc structures and algorithms for storing and querying graphs.

A graph G = <E; V> is an abstract data type showing connections (edges E) between pairs of vertices (V). Nodes identify entities and their properties, while relationships are joining attributes between tables with eventual additional characteristics. Unlike other databases, relationships take first priority. A graph database is purpose-built to handle highly connected data, providing great performance, flexibility and frictionless development. Queries allow to match pattern of nodes and relationships in a graph, providing ACID transaction compliance without specifying details on how to implement operations. The data model for a graph database is significantly simpler and more expressive than those of relational or other NoSQL databases (e. column DB). In this model data are stored in the form of an edge, node, or attribute. Each node and edge can have attributes, and both nodes and edges can be labelled.

Although graph database technology can be considered as a new innovation, there are several graph database implementations and a wide range of studies comparing their performance with respect to different aspects. In the last time there have been an increasing number of different implementations of graph databases, for example AllegroGraph, DEX, HypergraphDB, Tigergraph, Neo4J and Amazon Neptune (Rawat and Navneet, 2017).

Neo4j is a world-leading open-source graph database working on a graph data model created in 2007 and actually ranked as the first graph database by db-engines.com (Db-engines.com, 2020). Applications built using Neo4j tackle connected queries written using Cypher (Neo4j, 2020), a declarative graph query language that allows for expressive and effcient querying and updating of the graph. The database offers built-in visualization and implementations of all covered graph algorithms. In this way Neo4j brings ETL, analytics and improved performance in a unique tool. Beyond declarative, Neo4j brings unique numerical tools to characterize the graph, which have made it the elective instrument for data challenges including artificial intelligence, fraud detection, real-time recommendations and master data (Nagarajan and Talbert, 2019).

5.2 THE GRAPH DATABASE STRUCTURE

Starting from the database E-R structure depicted in Fig. 1 and following the technique proposed in (Singh and Kaur, 2015), the resulting structure is an unweighted directed graph, with a data composition of: 670,634 patients; 1377 doctors; 8,328,272 antibiotics prescription; 2465 antibiotics; 7,587,009 other prescriptions and 21248 medicines. As stated before, these numbers represent an extraction of the complete DB due to the need of detailed recent reports representing the problems of co-prescription patterns. Following, in Fig. 2, is depicted the representation of the structure of the graph database obtained.

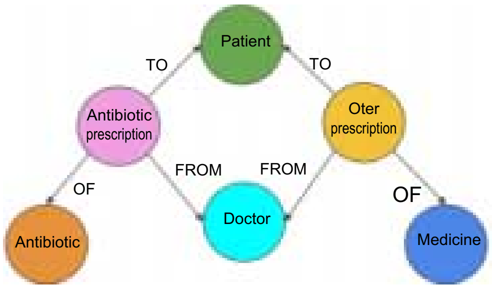

Fig. 2: The graph database structure obtained with Neo4j library APOC

Fig. 3 depicts an example of graph subset obtained extracting the prescriptions history in 10 years of one elderly patient and his associated drugs and doctors.

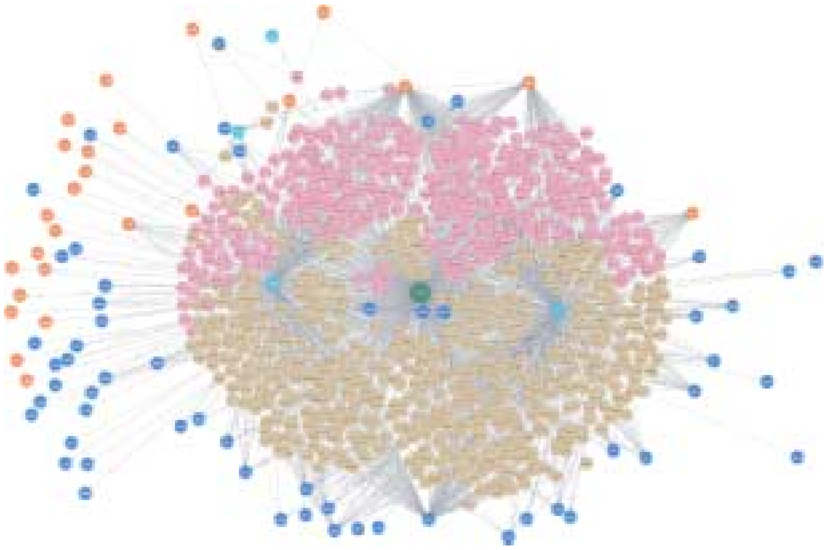

Fig. 3: Prescription graph in 10 years of an elderly male patient

In 10 years, the patient had 360 antibiotic prescriptions and 524 other prescriptions, he changed 4 doctors, assumed 25 antibiotics and 59 other medicines. From this first patient-centred visualisation of the graph is possible to identify different behaviours of general practitioners: each one of them is linked to specific antibiotics and medicines. Fig. 4 displays the antibiotic prescription graph of a general practitioner in a particular year.

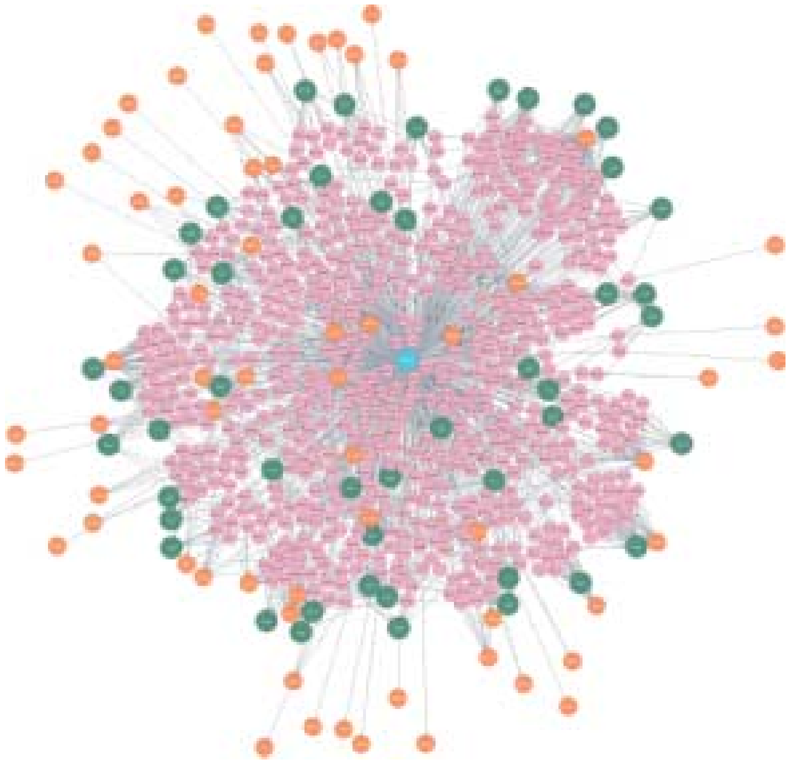

Fig. 4: antibiotic prescription graph in 2017 of a general practitioner.

5.3 GRAPH STATISTICS

A first set of global and local statistics (Table 2) are used to get the first insight on the graph and its components, using 10 years of data.

5.4 GRAPH VISUALIZATION OF ANALYSIS

Fig. 5 depicts the co-prescription graph: light nodes are antibiotics, dark other medicines, the number of co-prescriptions is represented thru the weight of the link. The two antibiotics at the center are the two most prescribed. Along with the relation

Tab. 2: Global graph statistics

Number of nodes	16,611,005
Number of relationships	47,745,841
Average prescriptions per doctor	11557.93
Standard deviation	15457.6
Maximum per doctor	96841
Minimum per doctor	1
Average prescriptions per patient	23.73
Standard deviation	40.46
Maximum per patient	1567
Minimum per patient	1

one can visualize the features of each node and link. All the antibiotics are among the most prescribed ones, which justifies their presence in the co-prescriptions as well. The isolated component represents an antibiotic (antimalarial) given to treat arthritis coupled with a corticosteroid for rheumatism (co-prescribed about 5000 times).

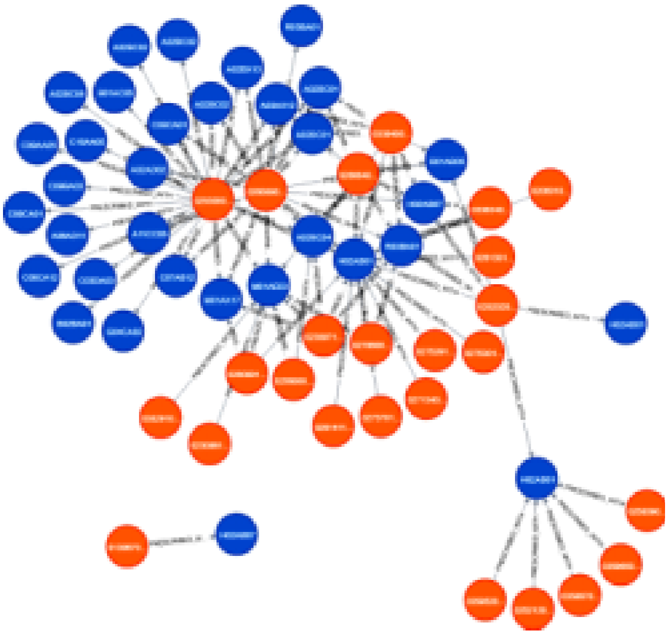

Fig. 5: Co-prescription graph

5.5 NETWORK MODELS

Graph algorithms are the powerhouse behind analytics for connected systems (Newman, 2018; Needham and Hodler, 2019). These algorithms use the connections between data to evaluate and infer the organization and dynamics of real-world systems.

5.5.1 CENTRALITY ALGORITHMS

The Betweenness Centrality algorithm calculates the shortest (weighted) path between every pair of nodes in a connected graph, using the breadth-first search algorithm. Each node receives a score, based on the number of these shortest paths that pass through the node. Nodes that most frequently lie on these shortest paths will have a higher betweenness centrality score. As expected, the results highlight the most prescribed antibiotics have the highest betweenness scores: since they often get prescribed, the number of relationships involving these nodes is high, increasing the probability of a shortest path crossing them.

Degree Centrality is the simplest of all the centrality algorithms. It measures the number of incoming and outgoing relationships from a node, analysing its influence. Obtained results are similar to Betweenness Centrality: most prescribed products have the largest number of paths.

Another approach of Degree Centrality analysis involves general practitioners, to under-stand their influence as prescribers. Calculating degree among doctors is useful to determine whether the top antibiotics prescribers are also the top prescribers, using the top 5 doctors. Analysing the obtained results, the overprescribing of some general practitioners is clear: most of them writes a number of antibiotic prescriptions equal to others having double the amount of patients.

5.5.2 SIMILARITY ANALYSIS

Due to the heavy amount of resources and computational time required for computation on a graph with nodes in the magnitude order of millions, we decide to extract small time range and only the relevant features in order to reduce the computational cost and produce recent reports. A new subset is extracted, selecting all data from the year 2017 of patients having at least 10 antibiotic prescriptions during that year. A new subset is extracted an it’s composed by: 8,228 patients; 115,443 antibiotic prescriptions; 775 doctors; 946 antibiotics; 181,016 other prescriptions; 4654 other medicines. Having a time range of one year and only considering patients getting antibiotic prescriptions rather often does not offer additional information on antibiotic resistance, yet it allows to obtain relevant patterns of prescribing habits. The output of graph processing will be used to apply computationally‘expensive graph algorithms.

Jaccard similarity is computed among doctors based on antibiotic prescriptions in 2017, considering the Cartesian product of nodes (775,775) with a threshold of 0.4. The algorithm computes for each node (doctor), similar other node, according to the defined threshold. Since a single doctor can be the similar to several ones, nodes tend to group in clusters: connected components have a high probability of having common prescription patterns. A large cluster groups most of general practitioners, allowing to quantify the reasonable assumption that popular antibiotics are prescribed by the majority of doctors.

5.5.3 COMMUNITY DETECTION

The concept of community refers to the structure (topological and relational aspects) of dense sub-components of a graph (Candelieri et al., 2017). Node attributes are considered as well, to obtain additional inferences on their membership. Data is linked with discrete attributes, and applications are based on graphs properties.

Identifying communities allows to understand the global pattern and relationships between individuals with the similar features. Different groups interact between one another, and can be related, giving information on the global schema. The used algorithm is the Louvain algorithm (Lu et al., 2014) based on similarity detection results. Communities are identified between doctors according to their similarity, based on antibiotic prescriptions and resulting in 526 communities, 495 of which are composed by a singlet. The presence of clusters implies that groups of doctors have the same prescriptive habits. These results highlight four most representative communities: Large prescribers of antibiotic A; Large prescribers of antibiotic B; Rare prescribers; Large prescribers. Clustering chronological snapshots of data highlights general practitioners transitioning from antibiotic B to antibiotic A, and rare prescribers to large.

6. CONCLUSION

Graph databases offer a completely different perspective compared to relational ones, allowing to understand linkage between nodes and their behaviour. Despite not having a relationship between each pair of nodes, the whole graph can be effciently crossed through paths, displaying different views focussed on nodes or their type while linking them to the whole structure. Building targeted datasets and extracting features is immediate, adding relationships with eventual weight and filtering according to properties.

Those datasets or query results are fed to graph algorithms, which‘can add resulting properties to enable further considerations without having to export data in a format readable by programming languages. This approach has also scaled up the region of computational feasibility: networks up to millions of nodes are routinely handled and stored in graph DB, while computationally intensive analysis can be performed in the order of hundred/thousands of nodes, while millions of nodes require high performance environment.

REFERENCES

Angles R., (2012). “A Comparison of Current Graph Database Models”, in *IEEE 28th International Conference on Data Engineering Workshops*, Arlington, VA:171-177.

Candelieri, A., Giordani, I., Archetti, F. (2017). “Automatic Configuration of Kernel-Based Clustering: An Optimization Approach”, in *Learning and Intelligent Optimization. Prooceeding of LION 2017 conference* eds. R. Battiti, D. Kvasov, Y. Sergeyev Lecture Notes in Computer Science, Springer, Cham: 34-49.

Cavallo, P., Pagano, S., Boccia, G., De Caro, F., De Santis, M. and Capunzo, M. (2013). “Network analysis of drug prescriptions”, *Pharmacoepidemiol Drug Safety*, 22(2):130-7.

Db-engines.com, *DB-Engines Ranking of Graph DBMS*. https://db-engines.com/en/ranking/graph+dbms. Last access: 20/03/2020.

Lu, H., Mahatesh, H. and Kalyanaraman, A., (2014). “Parallel Heuristics for Scalable Community Detection”, *Parallel Computing*, SI: scientific graph analysis.

Nagarajan, R. and Talbert, J. (2019) “Network Abstractions of Prescription Patterns in a Medicaid Population”, *AMIA Jt Summits Transl Sci Proc*: 524–532.

Needham, M., Hodler, A. E. (2019). Graph Algorithms: Practical Examples in Apache Spark and Neo4j. O'Reilly Media.

Neo4j, Inc. *The Neo4j Cypher Manual v4*.0 https://neo4j.com/docs/cypher-manual/current . Last access: 05/02/2020.

Newman, M. (2018). *Networks*. Oxford University press.

Park, Y., Shankar, M., Park, B. and Ghosh, J. (2014). “Graph databases for large-scale healthcare systems: A framework for efficient data management and data services,” *IEEE 30th International Conference on Data Engineering Workshops*, Chicago, IL: 12-19.

Prescott, G.M., Patzke, CL., Brody, P.M. and Prescott, W.A.Jr. (2018). “Comparison of prescribing patterns between United States and Dominican Republic prescribers on short-term medical mission trips.” *Int Health*. 10(1):27-32.

Rawat, D.S., and Navneet K. K. (2017). “Graph Database: A Complete GDBMS Survey.” *International Journal for Innovative Research in Science & Technology*. 3(12): 217-226.

Singh, M. and Kaur, K. (2015). “SQL2Neo: Moving health-care data from relational to graph databases,” *IEEE International Advance Computing Conference (IACC)*, Banglore: 721-725.