Graph database implementation of fine spatial scale urban infrastructure networks

Ji Q[1], Barr S[1], James P[1], Fairbairn D[1]

[1]School of Engineering, Newcastle University, Newcastle upon Tyne

NE1 7RU

Summary

An effective database is an essential component in managing geospatial infrastructure data and the development of infrastructure asset decision support platforms. Traditional approach is the relational spatial database. Such a solution performs well for standard spatial queries, but is often poor at efficiently retrieving data and performing queries for large infrastructure network instances. In this paper, we propose the usage of a graph database (Neo4j) to model such networks, and compare its performance with a traditional solution (PostgreSQL/PostGIS). Performance tests indicate that graph databases offer significant performance improvements when modelling fine scale urban infrastructure networks that have complex topology and dependencies.

KEYWORDS: infrastructure network, graph database, performance

1. Introduction

Urban infrastructure networks such as transport, energy, and water play a key role in the functioning of modern cities (Murray and Grubesic, 2007). The location and state of infrastructure assets is vital for infrastructure providers and utility companies (Almadi-Echendu et al, 2010), and information on infrastructure vulnerability, demand/capacity, and dependencies or interdependencies is equally important to understand infrastructure networks systematically (Rinaldi et al, 2001). In many countries, individual operators in specific infrastructure sectors (Woodhouse, 2014), as well as several large research initiatives (Barr et al, 2016), have realised the importance of developing their data and information management platforms for better infrastructure planning and decision support.

At its core, such platforms require appropriate database systems that can handle the wide range of disparate data and relationships required for infrastructure systems modelling and analysis (Barr et al, 2016). Traditionally a spatial relational approach is used, such as the Oracle Spatial Network Extension (British Telecom, 2012; Fikjez and Řezanina, 2016) or specifically developed schema for representing dependence/interdependence between infrastructure networks (e.g., the NISMOD-DB approach developed by the Infrastructure Transitions Research Consortium (Barr et al, 2013)).

The spatial relational approach is naturally strong in dealing with queries involving the attributes matching (such as finding all the assets with specific attribute values), and spatial calculations (such as finding all assets within a certain distance). However, it is somewhat limited in analysing large complex network topologies, such as intra-city scale electricity distribution networks. Recently, NoSQL graph database have been proposed as a general approach for the more efficient storage and retrieval of network data (Have and Jensen, 2013). In this paper, we propose the use of a graph database to model large-scale urban spatial

infrastructure networks with complex topology. Several performance tests and comparison between traditional relational database approaches are presented.

2. Relational and graph database approaches

Urban infrastructure networks are represented by spatial networks, which comprise geometry, attributes, and topological connectivity. Spatial relational approaches to network representation rely on a schema approach to define relational tables to store all the network information (Barr et al, 2013). Figure 1 shows the general flow of reading and writing network data within a spatial relational approach that has been developed for representing infrastructure networks (Barr et al, 2013).

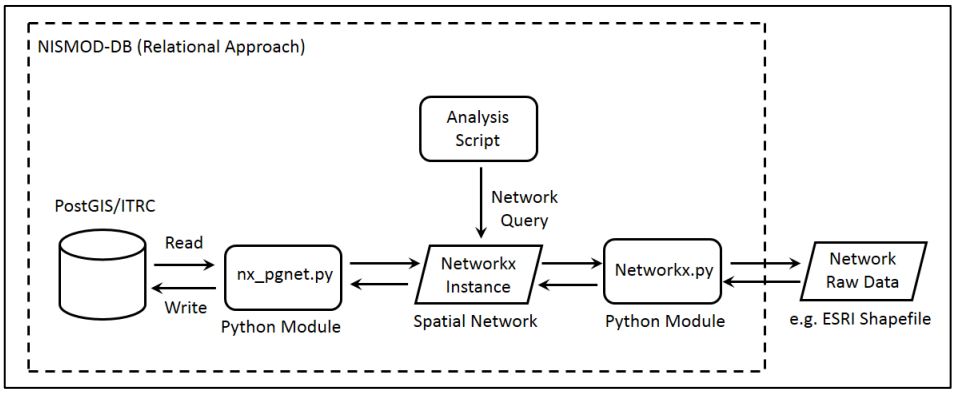

Figure 1. Relational approach for modelling urban infrastructure networks.

To write a network into the spatial relational database framework requires converting raw network data to a Networkx instance (Networkx, 2014), and then to write it to the database via another module specific for this schema. When undertaking specific network queries (such as shortest path calculation), data from the database must be read back to a Networkx instance, which is then queried via Networkx functions. This processing flowline can lead to performance issues when the network to be read is very large (hundreds of thousands of nodes/edges).

To address this issue, Neo4j, the most popular graph database software (Neo4j, 2017), is proposed to model urban spatial infrastructure networks. Its data model is a "property graph", which consists of nodes and relationships, both with their own properties. Neo4j uses its own query language, Cypher, which is capable of both graph querying (based on topological connections) and value querying (based on properties). Figure 2 shows the graph-based approach to network write, read and analysis.

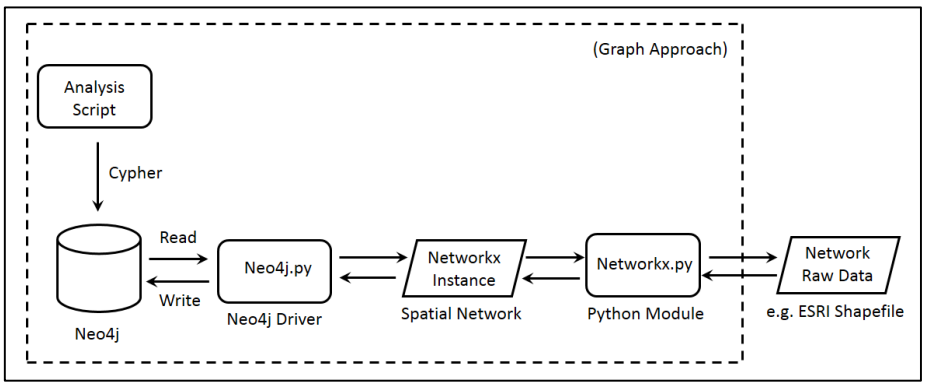

Figure 2. Graph approach (Neo4j) for modelling infrastructure networks.

3. Performance Test

The spatial relational interdependent infrastructure network schema developed by ITRC (Barr et al, 2013) and Neo4j were compared with regards to their ability to ingest, read and analyse infrastructure networks. The spatial relational approach used PostgreSQL 10.3/PostGIS 9.4 along with Networkx1.11, nx_pgnet 0.9 as its base RDMS and software dependencies. The graph-based approach employed Neo4j 3.1.3 and Neo4j Python driver 1.5.1. Three scenarios were developed to evaluate the databases performance to process infrastructure networks consisting of write, read and network search operations. The performance test was run on a Windows 8.1 operation desktop machine, with dual core processor (Intel® Core ™ i7-4720HQ CPU @ 2.60GHZ) and 8GB memory. The three scenarios are demonstrated below, each using electricity feeder/distribution spatial infrastructure networks. The performance (processing time) of relational approach is regarded as the benchmark (100%) for comparison.

Scenario 1 – single small network test

The network datasets used were differently sized electricity feeder/distribution networks in a suburban area within the city of Newcastle upon Tyne. Normally, a feeder/distribution network contains a substation node, building nodes and distribution nodes. Figure 3 shows an example electricity feeder/distribution network containing 814 edges, 815 nodes, and serving 409 buildings. Overall five distribution networks were used in this scenario, with approximately 100, 200, 400, 800, and 1600 nodes respectively. The specific network search task was to find the shortest path from the substation node to every building node. Test results are shown in Figure 4 for the Neo4j graph-based approach compared to the processing time for the spatial relational approach.

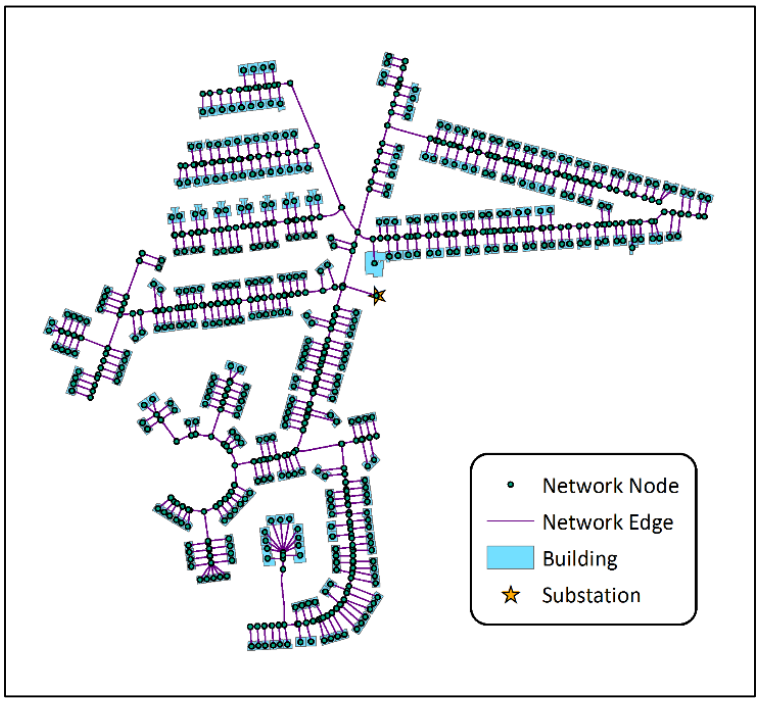

Figure 3. An example of electricity distribution networks of 815 nodes.

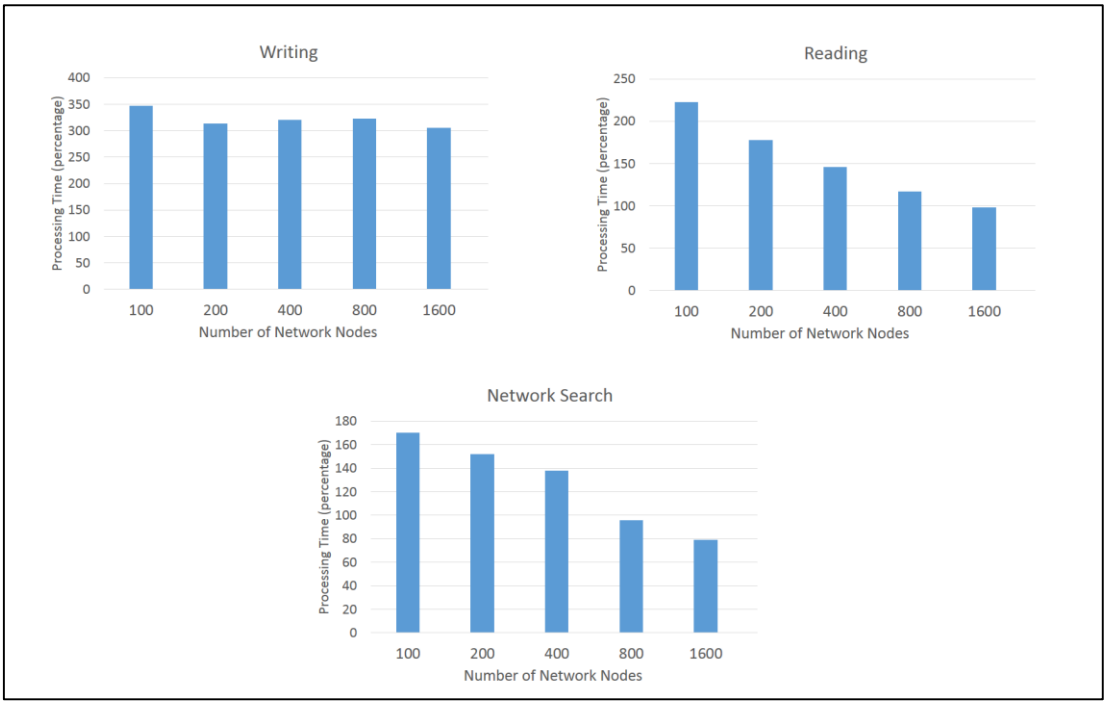

Figure 4. Performance tests for Scenario 1.

Figure 4 shows that the spatial relational approach has an obvious advantage in writing the raw network data to the database. The graph approach required more than 3 times the processing time due to encoding geometry into WKT string within Neo4j. Many spatially complex edges resulted in long strings when converted into WKT, causing the poor performance overhead. In reading and network searching, at very small network sizes the relational approach performed significantly better. However, as the size of network increases, such difference became much smaller.

Scenario 2 – mixed networks test

This scenario used seven datasets comprising of multiple electricity feeder/distribution networks with 2500, 5000, 10000, 20000, 40000, and 80000 nodes and the entire feeder/distribution network for Newcastle upon Tyne comprising of over 600 sub-feeder networks with a total of 191577 nodes (Ji et al, 2017) (Figure 5). The actual network searching involved for each substation node finding the shortest path to all its building nodes. Test results are shown in Figure 6.

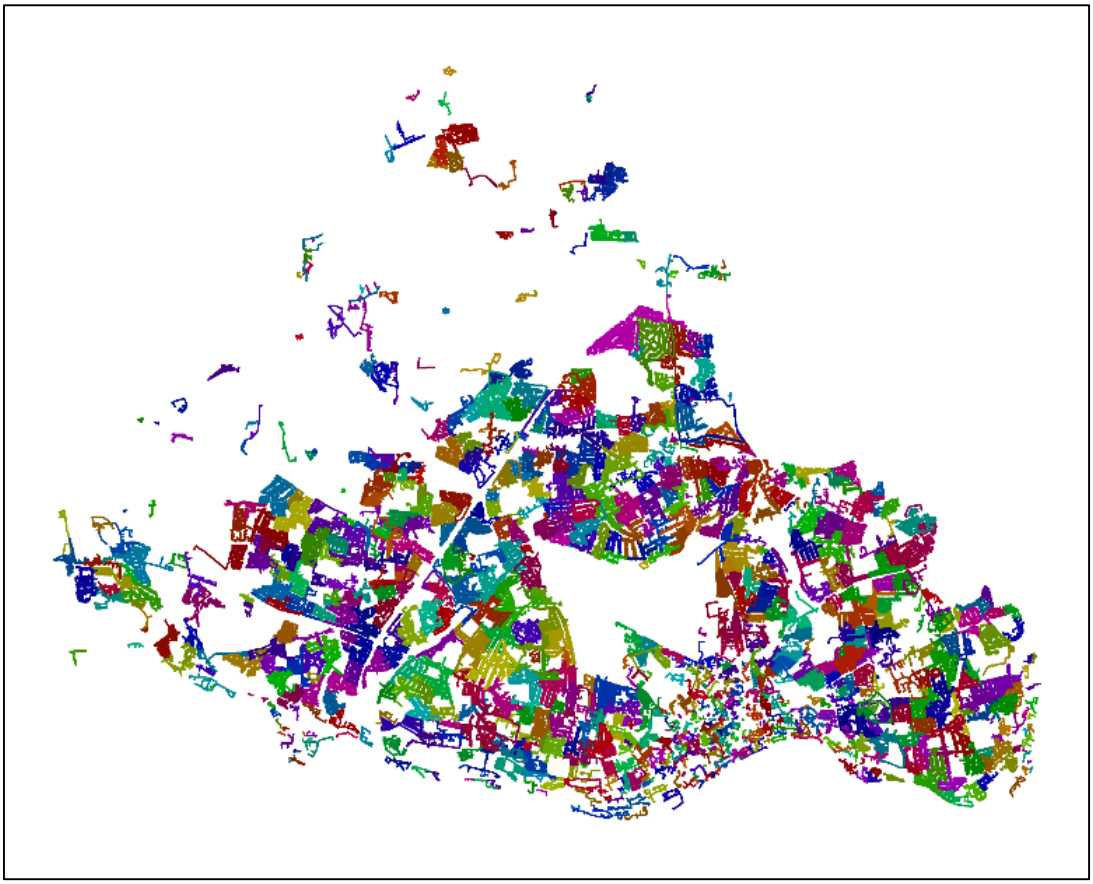

Figure 5. Entire-city level electricity feeder/distribution networks for Newcastle upon Tyne.

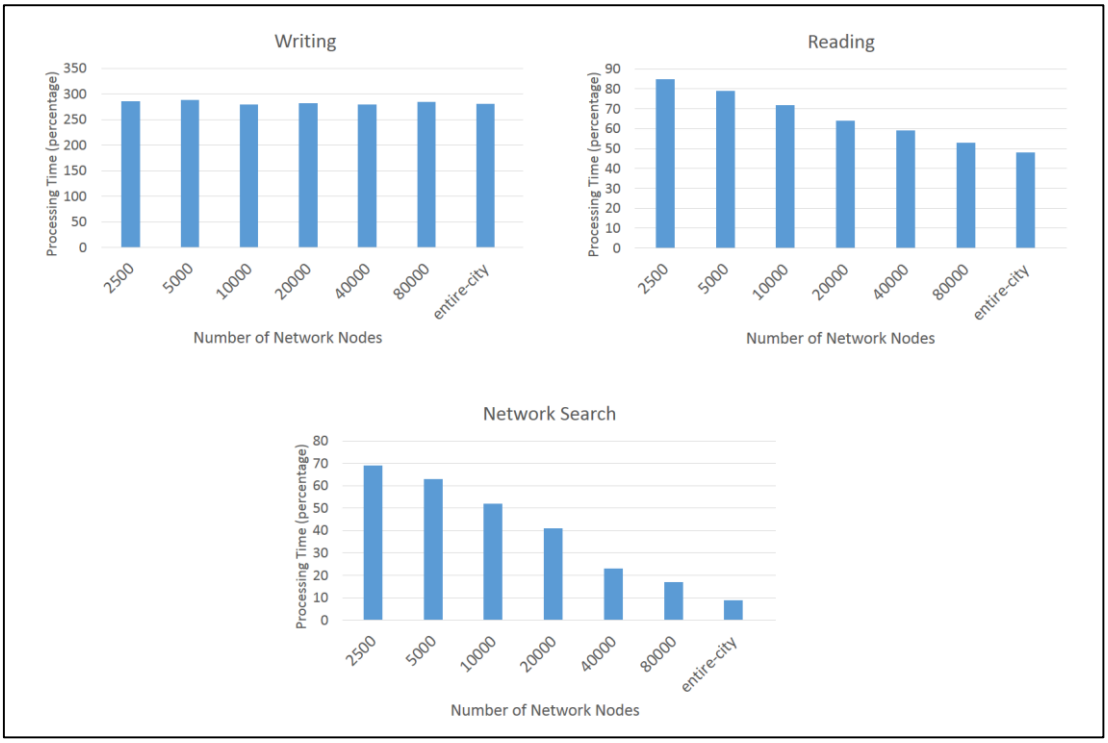

Figure 6. Performance test for scenario 2.

In Figure 6, the spatial relational approach still performed better in writing. However, for network reading, the graph approach performed better as the size of networks increased and it

only required half the time for the city-scale network. Moreover, the Neo4j graph approach has an obvious advantage in network searching in larger networks with it being ten times faster in the case of the entire network for Newcastle upon Tyne. The main reason is that in order to perform the network search, the relational approach needs to first read the network into a Networkx instance. The graph approach, however, allows the use of Cypher to query the database directly.

Scenario 3 – single large network test

The final scenario investigated used a large single spatial infrastructure network comprising of the England and Wales national electricity transmission-distribution network (Figure 7), containing 170,667 nodes and 173,039 edges. The network search, in this scenario, involved selecting 10 nodes at random and then for each searching for the closest node with a topological path greater than 20 nodes. Test result is shown in Figure 8.

Figure 7. The UK national electricity transmission-distribution network.

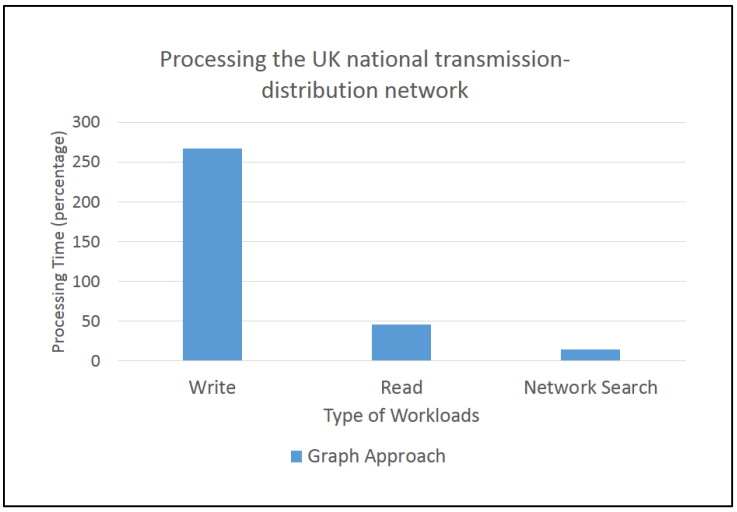

Figure 8. Performance test for scenario 3.

The comparison with regards to writing and reading follows almost the same pattern as the scenario 2, where spatial relational approach is more efficient at writing being two times quicker. Importantly, the graph-based approach is significantly better than the spatial relational approach in terms of analytical searching/analysing the network topology; again being six times faster than the spatial relational approach.

To conclude, in each of these scenarios, the spatial relational approach always performed better in network writing, regardless of the network size. Considering the fact that writing the network into a database is not a very frequent action, the underperformance of the graph approach is to some extent acceptable. On the other hand, the graph approach showed its strength in network reading and network search operations, especially for large networks. This is considered more important, since reading and network search tasks are generally more frequent in infrastructure network analysis.

4. Conclusion

The geospatial database is an essential part of developing platforms for modelling urban infrastructure networks, where topology, geometry and attributes from the networks must be stored, retrieved and queried in an efficient manner. Traditional approach is a spatial relational database, while in this paper the use of a graph database is proposed for the representation and analysis of large-scale infrastructure networks. The performance tests showed that the graph approach performed better than the traditional relational approach in network reading and network search. However, the current standard Neo4j graph database does not have a good support to spatial data storage. Future work will be directed towards a more appropriate way to store spatial data in the graph database (other than the WKT string), such as using Neo4j spatial extension (https://neo4j-contrib.github.io/spatial/), or the development of a federated database system that stores spatial data and network topology separately in different databases to optimize the performance of the entire system.

4. Acknowledgement

We thank the ITRC project team for proving database schema and scripts. We also thank the Ordnance Survey (OS) for providing the MasterMap data used in this paper.

6. Bibliography

Qingyuan Ji is a PhD student in the School of Engineering at Newcastle University. His PhD is on the development of a spatial-analytical platform for the analysis, simulation and visualization of infrastructure networks within cities. Stuart Barr is Professor of Geospatial Systems Engineering in the School of Engineering at Newcastle University. Philip James and David Fairbairn are Senior Lecturers in Geographic Information Science in the School of Engineering at Newcastle University.

Reference

Amadi-Echendu, J., Willett, R., Brown, K., Hope, T., Lee, J., Mathew, J., Vyas, N., and Yang, B. (2010). What is Asset Management? In Amadi-Echendu, J., Brown, K., Willett, R., Mathew, J., (eds), *Definitions, Concepts and Scope of Engineering Asset Management*, Springer, p.3-16.

Barr, S.L., Alderson, D., Robson, C., Otto, A., Hall, J., Thacker, S. and Pant, R. (2013). 'A national

scale infrastructure database and modelling environment for the UK', *International Symposium for Next Generation Infrastructure*. Wollongong, New South Wales, Australia.

Barr, S., Alderson, D., Ives, M.C. and Robson, C. (2016). Database, simulation modelling and visualisation for national infrastructure assessment. *The Future of National Infrastructure: A System-of-Systems Approach*, Cambridge University Press, p.268.

British Telecom. (2012). Use of Oracle Spatial Network Data Model at British Telecom. Available at: http://download.oracle.com/otndocs/products/spatial/pdf/british_telecom_ug.pdf

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

Fikejz, J. and Řezanina, E. (2016). The Design of Railway Network Infrastructure Model for Localization of Rolling Stock with Utilization Technology Oracle Spatial and Dynamic Database Views. *Advanced Computer and Communication Engineering Technology*, Springer, Cham, p.935-935.

Have, C.T. and Jensen, L.J. (2013). Are graph databases ready for bioinformatics? *Bioinformatics*, 29(24), p.3107.

Ji, Q., Barr, S., James, P., and Fairbairn, D (2017). A heuristic spatial algorithm for generating fine-scale infrastructure distribution networks. In: 25th GISRUK 2017, Manchester, UK.

Murray, A.T. and Grubesic, T.H. (2007). Overview of reliability and vulnerability in critical infrastructure. *Critical Infrastructure: reliability and vulnerability*, Springer, Berlin, Germany.

Neo4j. (2017). https://neo4j.com/.

NetworkX. (2014). https://networkx.github.io/documentation/stable/

Rinaldi, S.M., Peerenboom, J.P., & Kelly, T.K. (2001). Identifying, understanding and analysing critical infrastructure interdependencies. *IEEE Control Systems*, 21(6), p.11-25.

Woodhouse, J. (2014). Standards in asset management: PAS55 to ISO55000. *Infrastructure Asset Management*, 1(3), p.57-59.