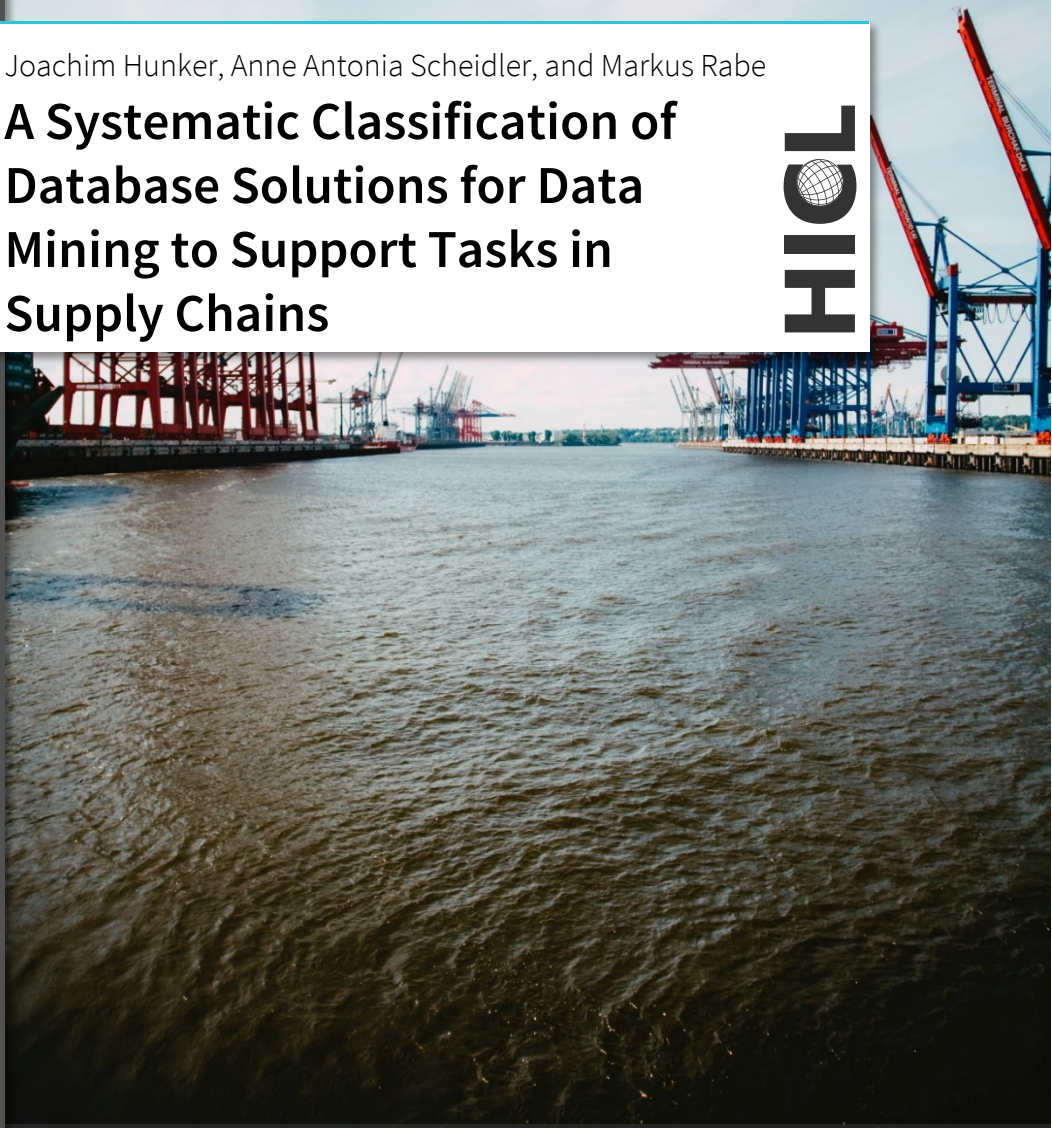
Hunker, Joachim; Scheidler, Anne Antonia; Rabe, Markus

Conference Paper

A systematic classification of database solutions for data mining to support tasks in supply chains

Joachim Hunker, Anne Antonia Scheidler, and Markus Rabe

A Systematic Classification of Database Solutions for Data Mining to Support Tasks in Supply Chains



Published in: Data science and innovation in supply chain management
Wolfgang Kersten, Thorsten Blecker and Christian M. Ringle (Eds.)
ISBN: 978-3-753123-46-2 , September 2020, epubli

A Systematic Classification of Database Solutions for Data Mining to Support Tasks in Supply Chains

Joachim Hunker [1], Anne Antonia Scheidler [1], and Markus Rabe [1]

1 – TU Dortmund Fachgebiet IT in Produktion und Logistik

Purpose: Our research shows that considering well suited NoSQL databases is beneficial for logistics tasks. For answering tasks we rely on the widespread methods of Data Mining. We stress that using relational databases as basis for Data Mining tools cannot cope with the growing amount of data and that using NoSQL databases can be an important step to address these issues.
Methodology: This paper discusses Data Mining in the context of Supply Chain Management tasks in logistics and its requirements on databases. The paper demonstrates that using NoSQL databases as basis for Data Mining process models in logistics is a very promising approach. The research is based on a case study, whose core element is the analysis of different well established studies.
Findings: The paper presents results which show that Data Mining tools widely support NoSQL databases through available interfaces. Findings are presented in a comparison table which considers dimensions such as Data Mining tools and supported NoSQL databases. To show practical feasibility, a Data Mining tool is used on data of a Supply Chain stored in a NoSQL database.
Originality: The novelty of this paper emerges from addressing issues that have so far been insufficiently analyzed in the scientific discussion. The modular structure of the addressed research method ensures scientific traceability. Breaking down tasks and their requirements on databases in the field of Data Mining is a first step towards meeting trends like Big Data and their challenges.

First received: 12. Mar 2020 Revised: 21. Jun 2020 Accepted: 12. Aug 2020

1 Introduction

Logistics is one of the most important economic disciplines in Germany. Companies work together and, based on the requirements like globalization and just-in-time processes, form global networks, called Supply Chains. Trends that are summarized by buzz words like "Logistics 4.0" and "Big Data" have a major impact on Supply Chains (Borgi et al. 2017). The addition "4.0" stresses the importance of digital change in the sense of a fourth industrial revolution (Bousonville 2017). One of the consequences of these trends is the exponential growth of highly connected data, e.g., where forecasts predict a rise in worldwide data volume from 25 up to 163 zettabyte in the year 2025 (Reinsel et al. 2018). Therefore, executing tasks in Supply Chains is getting more complex. As an example for the complexity, a Supply Chain Management is confronted with multiple logistics tasks, e.g., real-time monitoring of deliveries throughout whole Supply Chains. Based on the fact that Supply Chains have emergent and coherent effects, the need to assist in answering logistics tasks as decision support for Supply Chain Management is necessary (Teniwut and Hasyim 2020). One of the frequently used methods in logistics is Knowledge Discovery in Databases, with Data Mining as its core process (Rahman et al. 2011). The prerequisite for running a successful Data Mining is a valid and preprocessed data basis. Since the 1970s, relational databases are dominant in the worldwide market (Garcia-Molina et al. 2009). Based on the addressed current requirements and trends, relational databases have difficulties in adapting to and processing of highly connected high volume data (Hecht and Jablonski 2011, Li and Manoharan 2013). These developments are resulting in the rise of different concepts such as non-relational (NoSQL) databases. Surveys

and forecasts show that logistics companies focus on hardware and Business Intelligence Analytics, but pay little to no attention towards NoSQL databases (Kelly 2015).

Our research closes the addressed gap and shows that NoSQL databases are well supported by existing Data Mining tools. We will highlight that focusing on well suited databases will be a big benefit for logistics tasks. For solving the logistics tasks, we rely on the widespread methods of Data Mining. For example, we will emphasize that graph databases are a native way to store data, e.g. for routing in Supply Chains, and can be an important step towards real-time decision support in Supply Chain Management.

The paper is structured as follows: Section 2 introduces the theoretical background necessary for this paper. In Section 3 we discuss our research, highlighting interfaces of Data Mining tools in regards to NoSQL databases against the background of logistical tasks and present results while Section 4 discusses our findings. The paper closes with a brief summary and an outlook in Section 5.

2 Theoretical Background

In the following sections the theoretical background necessary for this paper is discussed. First, we introduce the Supply Chain as the problem domain and highlight exemplary tasks. In the light of our problem domain, we discuss different types of data storages, relational databases, and NoSQL databases. Since we rely on the well-established method of Knowledge Discovery in Databases, the process of Data Mining is briefly discussed. At the end of this section, common database interfaces are presented.

2.1 Tasks in Supply Chains

Trends such as globalization and digitalization have a major impact on Supply Chains. Actually, Supply Chains are not chains as the term indicates, but networks of different linked organizations that work together, with different processes and activities that produce a value for customers (Lambert 2014, Christopher 2016). Due to these trends, Supply Chains are nowadays very complex global networks (Serdarasan 2013). Mastering the complexity in a Supply Chain is a problem for Supply Chain Management. It is responsible for the cross-company design of the planning, control, and monitoring of the processes within a Supply Chain. The attempt to master the complexity results, for example, in various logistical tasks with which the Supply Chain Management is confronted and the answers to which are a central task within the framework of a suitable decision support. In this context, Supply Chain Management is confronted with a multitude of different logistics tasks. Typical tasks can be differentiated along the flows of a Supply Chain, e.g., material flow or information flow, e.g., the choice of the most

appropriate means of transport or the real-time monitoring of on-time delivery. A typical way to categorize tasks is to use the five top level categories (Plan, Source, Make, Deliver, Return) of the Supply Chain Operations Reference Model (SCOR), an established model for the standardization of processes within a Supply Chain, which has been used in previous work of the authors (Gürez 2015, Scheidler 2017). For example, an exemplary task for the category "Plan" is determining the future customer requirements, or for the category "Deliver" a typical task is finding the right and most efficient means of transport for a delivery. One of the typical characteristics of such tasks is the challenge of finding correlations, which are especially relevant for Supply Chain Management (Harland 1996).

The key factor to support the decision making process for tasks in Supply Chain Management is knowledge. Following the definition given by North and Maier (2018), knowledge is based on information that is combined with more information in a certain context to answer questions like "how", whereas information is based on data that are interpreted and answer questions like "who", "where", and "when". The reader is kindly referred to Rowley (2007) for a deeper analysis of the terms data, information, and knowledge.

Generating knowledge from a data set is a challenge. One of the consequences of today's trends for Supply Chains is the emergence of large, strongly interrelated data volumes, which have to be stored in a persistent and suitable way.

2.2 From Relational to NoSQL Databases

To support tasks in Supply Chain Management adequately and since methods to discover knowledge work on data, an appropriate way of persistent

data storage in Supply Chains is necessary. Nowadays, databases are typically used in, e.g., decision support systems. Simply put, a database is a collection of related data and, together with a database management system (a collection of software programs), forms a database system (Elmasri and Navathe 2011, Connolly and Begg 2015). Since the work by Edgar F. Codd in the early 1970s (Codd 1970), databases based on the relational datamodel have become the worldwide de facto standard and a default in systems for decision support today. Prominent examples are Oracle, MySQL, MariaDB, Microsoft SQL Server and IBM DB2 (Solid IT 2020). However, focusing only on relational databases leads to serious drawbacks regarding the modeling of data, e.g., the transformation of graphs into tables or the handling of datasets that fit the Big Data paradigm (Hecht and Jablonski 2011). Such datasets are characterized by at least 3V, volume, variety, and velocity and have been supplemented by experts with two additional Vs, value and veracity, to emphasize the financial value and the varying quality of data (Meier and Kaufmann 2019). To tackle the mentioned problems, a different type of databases has gained attention in recent research (Moniruzzaman and Hossain 2013, Jose and Abraham 2017). These databases are summarized under the term NoSQL, which stands for "not only SQL", to highlight that these databases do not rely on the Structured Query Language (SQL), the dominating, standardized database language to query and manipulate data stored in relational databases (Batra 2018). The term NoSQL was coined by Carlo Strozzi in 1998 while introducing a relational database without the need for SQL (Strozzi 2017). It should be noted that the term NoSQL is in fact misleading as it describes a non-property of such, even relational, databases and should be specified by non-relational. The authors

of this paper decided to use the term NoSQL since it has been established in both theory and practice.

Organizing data in a non-relational way is not a new idea per se and has existed even before the relational database has been invented, e.g., in the form of hierarchical databases. A precise, uniform definition for NoSQL databases cannot be identified in the scientific discourse. In this paper, the authors understand NoSQL databases as characterized by Meier and Kaufmann (2019). The authors state that data in NoSQL databases are not stored in relational tables and the database language is not SQL. Besides multiple different database types like object-oriented or XML-databases and a multitude of special-use databases, four core types of NoSQL databases can be distinguished (Edlich et al. 2011):

Key-Value Stores (e.g., Couchbase) store data by using an identifier (the key) and associate a value of any kind and complexity (hashes, strings, lists, sets, XML, etc.) to a key. Searches, for example, can be conducted by querying the keys, but not against values.

Column-Family (or Wide Column) Stores (e.g., Cassandra, HBase) store data in tables, but handle the data in columns (more precise: column-families) rather than in rows. Keys are applied here to any number of Key-Value-Pairs, which can itself be extended by a Key-Value-Pair and form a Column-Family.

Document databases (e.g., MongoDB) handle data similar to Key-Value Stores, but store the data in documents that follow a standard exchange format like the Javascript Option Notation, which enforces the data which are stored in the documents to be at least semi-structured.

Graph Databases (e.g., Neo4j) are databases that store the data in the form of a tree or a graph using nodes and edges, e.g., a labeled property graph.

Within a property graph, nodes and edges can be labeled with properties. Manipulation of the data is done via graph transformation or using the properties of a graph, e.g., traversing. One of the typical database languages is Cypher, which is used for labeled property graphs.
The advantage of this type of databases is that they can handle large volumes of unstructured and highly connected data (Big Data). Graph databases, for example, can handle highly networked data very well due to their graph structure. Both, relational and NoSQL databases, have their strengths. The concept to use different databases in parallel for different situations to make use of their advantages is called Polyglot Persistence (Sadalage and Fowler 2013). The authors stress that the nature of the data that are stored in a database and how to work with the data must be understood first and that using only relational databases per default as the single type of database will lead to disadvantages, e.g., in performance.
The data stored within a database serve as an input for knowledge discovery techniques.

2.3 Knowledge Discovery in Databases

Extracting knowledge from databases and making it available to logistical processes is a value-adding task. Through the targeted analysis of data sets, valuable knowledge can be gained for different business areas. Such knowledge can secure a competitive advantage in the long run. One method of extracting knowledge from large data sets is Knowledge Discovery in Databases (KDD). KDD is a non-trivial (Fayyad et al. 1996), iterative, and interactive process (Wrobel et al. 1996).

KDD consists of different phases, ranging from the pre-processing of data to the actual application of procedures and data preparation for monitoring purposes. The actual application of methods is the central step and is known as Data Mining. This central step is so important that nowadays the terms KDD and Data Mining are often used synonymously and many authors do not make a distinction in content (Adriaans and Zantinge 1996). This also becomes clear in the task definition of Data Mining, which Runkler (2010) states with "Extracting knowledge from data". In his process model, Fayyad et al. (1996) envisage not only Data Mining but also a selection as the selection and export of analysis data, preprocessing as the cleansing and correction of missing or incorrect data, transformation as the transformation of data into a suitable target format for analysis purposes and evaluation respectively interpretation as the evaluation of the results. Figure 1 shows the process model presented by Fayyad et al. (1996).

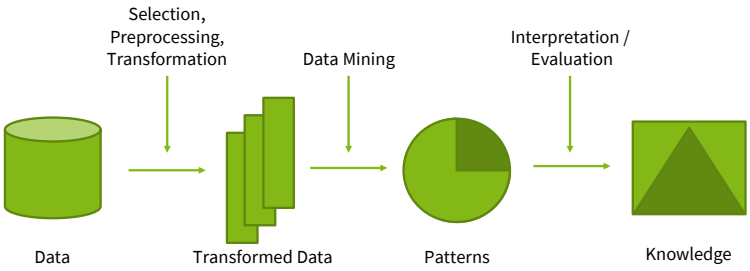


Figure 1: Knowledge Discovery in Databases (according to Fayyad et al. 1996)

Fields of application of Data Mining are complex. In logistics, there are a multitude of questions that can be supported and answered by Data Min-

ing. These include, for example, questions about future customer requirements, distribution centers, or delivery optimization. Depending on the questions, different Data Mining methods are used. As there are usually several Data Mining methods that can be used to solve tasks and often it is not clear which method is best suited, many tools support a wide range of Data Mining methods. This includes in particular various possibilities for preprocessing of data and preparation of Data Mining results. Only by means of a corresponding pre-implemented multitude of methods the company has the possibility to apply different methods and validate results in an acceptable time. In logistics, Data Mining procedures are often embedded in business applications. These include logistic assistance systems or decision support systems. Marakas (2003) and Turban and Volonio (2011) define them as systems that are under the control of one or more decision makers and support the decision-making process by using defined tools. The tools used pursue the goal of structuring decision-making situations and ultimately improving the effectiveness of logistical decision-making processes (Turban and Volonino 2011).

There are many tools available for the application of Data Mining methods, which have different functionalities. For example, the tools have different interfaces to support data import and, in many cases, enable direct application to different database systems.

2.4 Database Interfaces

Data Mining tools heavily rely on data, and the integration of a suitable database is, therefore, a key element. Since in many cases data stored in databases cannot be accessed from external tools directly (unless support is

directly integrated in the tool), a middleware can be used to bridge the gap between the server, where the database is running, and the client, where the software is executed (Elmasri and Navathe 2011). This middleware will also support independence from the specific database tool. The common approach of most interfaces is to offer a programming application interface that can be used to convert requests from an external application software such as Data Mining tools into standardized SQL commands, e.g., to extract data (Elmasri and Navathe 2011). This approach has the big benefit that there is no need to know the specifics and specialties of a used database. Both the database and the software must support the interface.

There is a plethora of existing programming interfaces that can be used to establish such a link. Prominent examples of programming interfaces are Open Database Connectivity (ODBC), Java Database Connectivity (JDBC), Object Linking and Embedding (OLE-DB), and ActiveX Data Objects (ADO). ODBC and JDBC are predominant and will, therefore, be discussed briefly in the following:

ODBC is a common, standardized interface when working with relational databases and was developed by Microsoft and the SQL Access Group. It uses standardized SQL to communicate with such databases (Garcia-Molina et al. 2009). ODBC offers a wide range of functions through a library for external applications to connect to an ODBC-capable database and execute SQL statements, e.g., to retrieve data (Li 2009b). It is independent of the programming language and used as a basis for multiple adaptations, e.g., SQL/CLI. Although intended for relational databases, some NoSQL databases also support ODBC (Li 2009b).

JDBC is part of the Standard Application Programming Interface of the JAVA programming language and enables applications written in JAVA to

access databases via build in packages (Elmasri and Navathe 2011). It follows the same approach and style as ODBC, but makes heavy use of the object orientation of JAVA. Although oriented towards relational databases (Li 2009a), some NoSQL databases offer JDBC programming interfaces.
Even more flexibility is provided by the use of so-called bridges, which translate from ODBC to JDBC or vice versa.

3 Database Solutions for Data Mining in Supply Chains

In the following section, we present our research that is based on a preliminary study conducted at our department IT in Production and Logistics (Rellensmann 2019). First, we conducted a structured analysis based on several well-established studies to select Data Mining tools. Second, we examined the selected tools regarding database and interface support. Third, we identified the databases supported by the tools and created a matrix based on the findings of the research carried out. The results of the matrix are in the final step exemplary matched to a corresponding task in Supply Chain Management.

3.1 Selection and Analysis of Data Mining Tools

As described in Section 2.3, Data Mining is the core phase of KDD. To perform Data Mining on a given data set, a specialized tool is used. For this research, we conducted a structured analysis based on three well established studies.

First, we used Gartners "Magic Quadrant for Data Science and Machine-Learning Platforms" from 2020, which covers the results from the study carried out in 2019 (Krensky et al. 2020). The analysis of the study resulted in 16 vendors respectively tools. The 16 identified tools are: Altair Knowledge Studio**,** Alteryx, Anaconda, Databricks, Dataiku, DataRobot, Domino, Google, H20.ai, IBM SPSS Modeler, KNIME, MathWorks MATLAB, Microsoft Analysis Service, RapidMiner, SAS, and TIBCO Software Statistica. In addition to the study carried out by Gartner, we analyzed the results of "Einsatz und Nutzenpotentiale von Data Mining in Produktionsunternehmen" from

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA (Weskamp et al. 2014). Supplementary to the already mentioned 16 tools by Gartner, the analysis of the study resulted in additional tools, which we included in our research. The tools are Oracle Data Mining, Statsoft Statistica, and SAP BI. As a third study we analyzed "Data Mining Software 2009" carried out by Dill (2009), which reduced the market of tools to a cross-section consisting of twelve programs and subjected it to a functional and benchmark comparison. The examination of this cross-section resulted in the following tools in addition to some programs already mentioned: Weka of University of Waikato, KXEN Analytic Framework, Viscovery SOMine, prudsys Discoverer, prudsys Basket Analyzer, and Bissantz Delta Master.

This results in a total of 25 tools, which were examined in the first step whether they still exists or whether it has been, e.g., bought by a competitor in the meantime. This resulted in four tools which were cut out:

KXEN Analytic Framework was bought by SAP and the features of the KXEN tool have been integrated into the applications offered by SAP.

prudsys Discoverer and prudsys Basket Analyzer have been discontinued and removed from the vendors portfolio.

Statsoft Statistica is distributed by Tibco Software under the same name since 2017 after been acquired by Dell.

In addition, the selected tools where examined with regards to external database support and if not, were cut out. This resulted in leaving out two Data Mining tools:

Google is offering a cloud-only based storage concept. On a side note, most of the tools are still in beta mode.

SAP BI has been integrated into a Data Warehouse Portfolio called SAP BW / 4HANA. It uses an SAP-integrated relational database (SAP HANA).
This reduces the number of Data Mining tools to 19, which will be further examined for database and interface support. It should be noted at this point that some tools offer specialized interfaces or import functions, e.g., for certain rare file formats. This will not be discussed further, since in the context of this paper the connection to databases is in the foreground.
The different tools are examined for database support on the basis of the accompanying documentation, website appearances and descriptions in the literature. Although possible through community support, we cut out self-programmed solutions and programming snippets and look for out-of-the-box-support via programming interfaces or directly implemented tool support as discussed in Section 2.4. In addition, we focus on relational and NoSQL databases as per our definitions given in Section 2.2 and will cut out a multitude of hybrid database systems and data warehouses, that inherit features both of the relational and NoSQL-world (e.g., NewSQL-databases).
Some of the Data Mining tools did not mention a supported database directly. However, if they supported one of the interfaces mentioned in Section 2.4, they were included in the results, accordingly.
In the following, we present the analysis results of three selected tools that are representative for the investigation of all Data Mining tools.

3.1.1 Alteryx

Alteryx offers a wide range of supported data sources in an available documentation (Alteryx 2020). In most cases, ODBC is used to establish a connection to a relational database. In some cases, e.g., Microsoft Access, Al-

teryx is able to read the database files directly. Furthermore, the tool enables a proprietary solution by integrating external interfaces directly (Alteryx Tool). The summary of the analysis is presented in Table 2. Supported interfaces are documented in brackets for every database.

Table 1: Relational and NoSQL Data Sources Supported by Alteryx

Database Type	Database
Relational	Amazon Aurora (ODBC), Amazon Redshift (ODBC), Amazon S3 (Alteryx Tool), Exasol (ODBC), HP Vertica (ODBC), IBM DB2 (ODBC, OLE-DB), Microsoft Access (database files directly), Microsoft Azure Data Lake Store (Alteryx Tool), Microsoft Azure SQL Database (ODBC, OLE-DB), MySQL (ODBC), Oracle (ODBC, OLE-DB), Pivotal Greenplum (ODBC), PostgreSQL (ODBC), SAP HANA (ODBC)
NoSQL	Apache Cassandra (ODBC), MongoDB (Alteryx Tool)

3.1.2 RapidMiner

The Data Mining tool RapidMiner enables the integration of databases via the JDBC interface (see Section 2.4) as stated in the official documentation (RapidMiner 2020). In addition, it offers built-in interfaces that can be used directly by using RapidMiner. The vendor's documentation lists directly supported databases, but emphasizes that all databases which support JDBC are supported and that ODBC is supported via a built-in bridge from JDBC to ODBC. NoSQL databases are supported by implemented connectors that can be used to establish a connection. The results of the analysis are presented in Table 3, which also documents the supported interfaces in brackets.

Table 2: Relational and NoSQL Data Sources Supported by RapidMiner

Database Type	Database
Relational	MySQL (JDBC), PostgreSQL (JDBC), HSQLDB (JDBC), Ingres, Microsoft Access, Microsoft SQL Server, Oracle
NoSQL	Apache Cassandra (RapidMiner Connector), MongoDB (RapidMiner Connector)

3.1.3 MathWorks MATLAB

MATLAB is a tool for data analytics and offers different so-called toolboxes to adapt the tool to certain applications. It offers a software feature called Database Toolbox, which supports the connection to different relational and NoSQL databases via ODBC and JDBC interfaces (see Section 2.4). Listed are the tools officially supported by MATLAB as described in the documentation (MATLAB 2020), although the vendor states that more databases can be possibly connected using the supported interfaces. The results of the analysis are presented in Table 4.

Table 3: Relational and NoSQL Data Sources Supported by MATLAB

Database Type	Database
Relational	Microsoft Access (ODBC), Microsoft SQL Server (ODBC, JDBC), Oracle (ODBC, JDBC), MySQL (ODBC, JDBC), PostgreSQL (ODBC, JDBC), SQLite (ODBC)
NoSQL	Apache Cassandra (MATLAB Toolbox), MongoDB (MATLAB Toolbox), Neo4j (MATLAB Toolbox)

3.2 Data Mining Tools and Database Support

The studies in Section 3.1 yielded 44 supported databases. These have already been divided into two broad categories, relational and NoSQL databases (see Section 3.1). In order to further specify the results, the NoSQL databases were divided into the four core categories as discussed in Section 2.2. In order to reduce the amount of results and to make the presentation in the table clear, prominent examples were selected based on their frequency of support. Here it should be mentioned that the results show a snapshot at a certain point in time of the study. It is possible that the support for a database will be added or terminated in the future by a vendor. The results have been divided into two tables, Table 4 and Table 5, for a better overview. First, results for supported databases by Data Mining tools are presented in Table 4. If one combination is marked by an 'X', the database is officially supported by the tool, otherwise, it is marked by a hyphen. We selected five relational databases (IBM DB2, Maria DB, Microsoft SQL, MySQL and Oracle) and five NoSQL databases (Cassandra and HBase (Wide Column), Couchbase (Key-Value), MongoDB (Document) and Neo4j (Graph)), representing all four core types (see Section 2.2).

Table 4: Supported databases by Data Mining tools

Data Mining Tool / Database	IBM DB2	Maria DB	Microsoft	MySQL	Oracle	HBase	Couchbase	MongoDB	Cassandra	Neo4j
Altair	X	X	X	X	X	X	X	X	X	-
Alteryx	X	X	X	X	X	-	-	X	X	-
Anaconda	-	X	X	X	X	X	X	X	X	X
Bissantz	X	X	X	X	X	X	X	X	X	-
Databricks	X	X	X	X	X	X	X	X	X	X
Dataiku	X	X	X	X	X	X	X	X	X	X
Datarobot	X	X	X	X	X	-	-	-	-	-
Domino	X	X	X	X	X	X	X	X	X	X
H2O.ai	-	X	-	X	-	-	-	-	-	-
IBM SPSS	X	-	X	X	X	-	-	-	-	-
KNIME	X	X	X	X	X	X	X	-	X	X
MATLAB	X	X	X	X	X	X	X	X	X	X
Microsoft	X	X	X	X	X	X	X	X	X	-
Oracle	-	-	-	-	X	-	-	-	-	-
RapidMiner	X	X	X	X	X	X	X	X	X	-
SAS Miner	X	X	X	X	X	X	X	X	X	-
SOMine	X	X	X	X	X	X	X	X	X	-
Tibco	X	X	X	X	X	X	X	X	X	-
Weka	X	X	X	X	X	X	X	X	X	X

To complete the results obtained, Table 5 shows the database interfaces, which have been discussed briefly in Section 2.4, supported by the Data Mining tools. If a tool supports an interface, the combination is marked by an 'X', otherwise by a hyphen. As mentioned above, we list database and interface support, accordingly. One blur is that a database support may be listed, but an interface is used for connection. Nevertheless, we highlight supported interfaces because it is well possible to use this interface to extend the databases that are officially supported.

Table 5: Database interfaces supported by Data Mining Tools

Data Mining Tool / Interface	ODBC	JDBC	OLE-DB	ADO.net
Altair	X	-	-	-
Alteryx	X	-	-	-
Anaconda	-	-	-	-
Byssantz	X	-	X	-
Databricks	-	X	-	-
Dataiku	-	X	-	-
Datarobot	-	X	-	-
Domino	-	-	-	-
H2O.ai	-	X	-	-
IBM SPSS	-	-	-	-
KNIME	-	X	-	-
MATLAB	X	X	-	-
Microsoft	X	-	X	-
Oracle	-	-	-	-
RapidMiner	-	X	-	-

Data Mining Tool / Interface	ODBC	JDBC	OLE-DB	ADO.net
SAS Miner	X	-	-	-
SOMine	X	-	X	-
Tibco	X	-	X	X
Weka	-	X	-	-

3.3 Data Mining and Databases for Tasks in Supply Chain Management - First Experiments

The combination of Data Mining tools and NoSQL databases to answer tasks of Supply Chain Management seems to be beneficial, especially taking into account the developments discussed in Section 2.1 and Section 2.2. For this research and to highlight the practical use on the basis of an application case, we used MathWorks MATLAB (see Section 3.1.3) as the Data Mining tool. As the database we relied on the widespread Neo4j, which is a popular graph database implementing a property graph model (see Section 2.2). Neo4j uses a database language called Cypher instead of SQL to retrieve and manipulate data stored in the database. MATLAB officially supports Neo4j as a Database (see Table 4). We established a connection successfully using the implemented MATLAB Toolbox (see Section 3.1.3).

To use a graph database seems promising, since Supply Chains are networks and are of emergent, coherent nature (see Section 2.1). Supply Chain data are high in volume and heavily interconnected. To use a graph to map a Supply Chain is, from a storage technology point of view, the native form of storing Supply Chain data persistent. The authors state that it is clear that a graph database, e.g., is the appropriate storage solution for tracking

down correlations of interest to Supply Chain Management (see Section 2.1 and Section 2.3).

Figure 2 shows a small excerpt of the anonymized data stored in the database. The dataset consists of multi-level Supply Chain data, filtered to one product, which have been imported into the Neo4j database. Each node as well as every edge contains a plethora of properties, e.g., name or site location (in latitude and longitude).

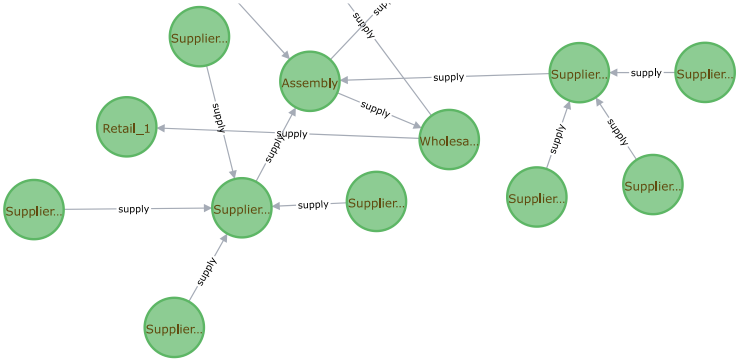
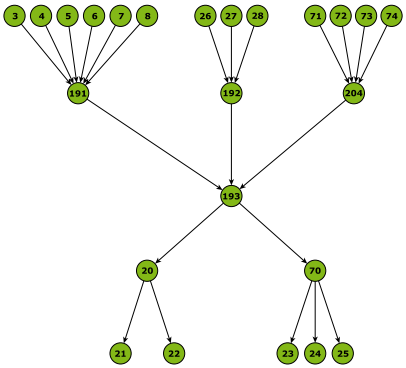


Figure 2: Excerpt of a Supply Chain Scenario in Neo4j

We conducted several tests with MATLAB to check for a correct and working database connection. MATLAB creates a Neo4j object which enables working with the graph directly. Second, we performed small experiments on the graph data, e.g., to track optimized routing or to find interesting patterns in the graph using, e.g., a segmentation algorithm (see Section 2.1 and Section 2.3). For example, exploring the whole structure of the Supply Chain and finding shortest paths can be done by graph traversing and using established algorithms like breadth-first graph search, directly on the data. Apart from this, e.g., to measure the importance of a node in the Supply Chain (centrality), Googles PageRank algorithm can be used, which makes use of the in- and outbound edges of a Node. Figure 3 shows the results generated by MATLAB of the application of PageRank directly onto the graph and a graph visualization.



Name	NodeLabels	NodeData	PageRank
{'191'}	{'Supplier_FirstTier'}	{1×1 struct}	0.088331
{'3' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'4' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'5' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'6' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'7' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'8' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'21' }	{'Retailer'}	{1×1 struct}	0.054531
{'20' }	{'Wholesaler'}	{1×1 struct}	0.094105
{'22' }	{'Retailer'}	{1×1 struct}	0.054531
{'192'}	{'Supplier_FirstTier' }	{1×1 struct}	0.051413
{'26' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'27' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'28' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'23' }	{'Retailer'}	{1×1 struct}	0.041186
{'70' }	{'Wholesaler'}	{1×1 struct}	0.094105
{'24' }	{'Retailer'}	{1×1 struct}	0.041186
{'25' }	{'Retailer'}	{1×1 struct}	0.041186
{'204'}	{'Supplier_FirstTier'}	{1×1 struct}	0.063719
{'71' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'72' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'73' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'74' }	{'Supplier_SecondTier'}	{1×1 struct}	0.014495
{'193'}	{'Assembly'}	{1×1 struct}	0.18727

Figure 3: MATLAB Results for the Centrality of Nodes in the Graph of a Supply Chain

4 Findings

In Table 4 and Table 5 of Section 3.2, we present, in a constructive selection of our results, databases and database interfaces supported by Data Mining tools. Overall, it can be seen across all Data Mining tools that databases can be connected very well. Although relational databases are dominating in practice and research over the last decades, our research shows that NoSQL databases are supported on a good level compared to their counterpart. For every highlighted Data Mining tool in Section 3.1.1, Section 3.1.2 and Section 3.1.3, the support for NoSQL databases can be identified. If we take into account further concepts like Cloud Object stores or Data Warehouses, relational databases are almost evenly well supported. The chart in Figure 4 summarizes the overall results and validates our initial intention. Also, it is visible from the research results of Section 3.2 that interfaces like ODBC and JDBC, although intended for relational databases using SQL (see Section 2.4), are used by NoSQL databases as well by adapting the interface to their own database language, e.g., translating SQL to Cypher when using the graph database Neo4j. Our small experiment in Section 3.3 showed good results on the basic possibility to use a NoSQL database as a basis for Data Mining and, therefore, generating knowledge to answer logistics questions (see Section 2.1). Furthermore, it was possible to use the advantages of the graph database, e.g., through directly applying graph techniques on the data without the need to transfer the Supply Chain as a graph into tables and back into a graph in the Data Mining tool, through a direct visualization of the results of Data Mining or, regarding data, the advanced storage of highly interconnected data.

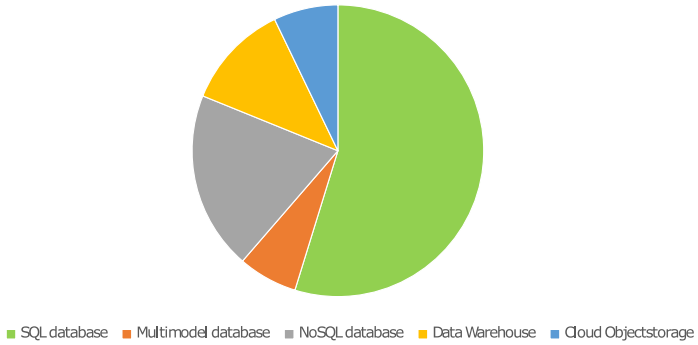


Figure 4: Data storage supported by Data Mining tools

As described in Section 2.3, the KDD phases for data preparation, selection, preprocessing, and transformation are essential for valid input data and running successful Data Mining (see Figure 1). Every phase of data preparation contains complex and time-consuming intermediate steps (see Section 2.3). Our experiments show promising results by using a suitable database to gain knowledge for answering tasks of Supply Chain Management (see Section 2.1) in light of data that fit into the Big Data paradigm (see Section 2.2). It is possible, in a specific Data Mining project, to reduce process steps in the data preparation phases and, therefore, increase overall performance and data quality in Data Mining.

5 Conclusion and Outlook

This paper discusses the possibilities of connecting databases and Data Mining tools for decision support in Supply Chains. To this end, relevant principles and resulting challenges, especially towards Big Data, were discussed. The identified subject areas were subsequently linked together. In a step-by-step modular analysis based on three established studies, relevant Data Mining tools were identified and examined for their interfaces and possibilities for connecting databases. On this basis, the possible databases were described and paired with the Data Mining tools. The results were presented in a comprehensive table using typical database representatives for every type. The associated subject-specific considerations regarding tasks in Supply Chain Management (see Section 2.1) were then discussed using a small logistics example. The example showed that the use of NoSQL databases in combination with Data Mining is worthwhile and in view of Supply Chain data which fit the Big Data paradigm an important component for the future.

The research field of Polyglot Persistence (see Section 2.2) in combination with Data Mining (see Section 2.3) is of central importance for further research, since the research presented in this paper assumes that one database is substituted by another, in our case a relational database by a graph database. At this point, there could be more beneficial opportunities to answer logistical tasks from Supply Chain Management, since the different types of databases could contribute their strengths regarding particular tasks and their specific requirements on data.

References

Adriaans, P. and Zantinge, D., 1996. *Data Mining*. Boston: Addison Wesley Professional.

Alteryx Inc., 2020. *Data Sources: Supported Data Sources and File Formats* [online]. Available at: <https://help.alteryx.com/current/designer/data-sources?tocpath=Data%20Sources%7CSupported%20Data%20Sources%7C_____0> [Accessed 19 May 2020].

Batra, R., 2018. *SQL Primer*. Berkeley, CA: Apress.

Borgi, T., Zoghlami, N., and Abed, M., 2017. Big Data for Transport and Logistics: A Review. In: *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*: IEEE, pp. 44–49.

Bousonville, T., 2017. *Logistik 4.0: Die digitale Transformation der Wertschöpfungskette*. Wiesbaden: Springer Fachmedien.

Christopher, M., 2016. *Logistics & Supply Chain Managem*ent. Harlow, England, New York: Pearson Education.

Codd, E.F., 1970. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13 (6), pp. 377–387.

Connolly, T.M. and Begg, C.E., 2015. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed. Boston, Mass.: Pearson.

Dill, M., 2009. *Data Mining Software 2009: Funktionsvergleich und Benchmarkstudie*.

Edlich, S., Friedland, A., Hampe, J., Brauer, B., and Brückner, M., 2011. *NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken*. 2nd ed. München: Hanser.

Elmasri, R. and Navathe, S., 2011. *Fundamentals of Database Systems*. 6th ed. Boston: Addison-Wesley.

Fayyad, U.M., Piatetsky-Shapiro, G., and Smyth, P., 1996. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17 (3), pp. 37–54.

Garcia-Molina, H., Ullman, J.D., and Widom, J., 2009. *Database Systems: The Complete Book*. 2nd ed. Upper Saddle River, NJ: Pearson/Prentice Hall.

Gürez, E., 2015. Zuordnung von Data Mining-Methoden zu problemspezifischen Fragestellungen von Supply Chain Management-Aufgaben. Bachelor Thesis. Technical University Dortmund, Faculty Mechanical Engineering, Department ITPL.

Harland, C.M., 1996. Supply Chain Management: Relationships, Chains and Networks. *British Journal of Management*, 7 (s1), pp. 63–80.

Hecht, R. and Jablonski, S., 2011. NoSQL Evaluation: A Use Case Oriented Survey. In: *2011 International Conference on Cloud and Service Computing: IEEE*, pp. 336–341.

Jose, B. and Abraham, S., 2017. Exploring the Merits of NoSQL: A Study Based on MongoDB. In: 2017 International Conference on Networks & Advances in Computational Technologies (NetACT): IEEE, pp. 266–271.

Kelly, J., 2015. *Big Data Vendor Revenue and Market Forecast, 2011-2026* [online]. Available at: <https://wikibon.com/big-data-vendor-revenue-and-market-forecast-2011-2026/> [Accessed 6 May 2020].

Krensky, P., den Hamer, P., Brethenoux, E., Hare, J., Idoine, C., Linden, A., Sicular, S., and Choudhary, F., 2020. *Magic Quadrant for Data Science and Machine Learning Platforms* [online]. Available at: <https://www.gartner.com/doc/reprints?id=1-1YCTPMUL&ct=200213&st=sb&utm_medium=Website+> [Accessed 24 May 2020].

Lambert, D.M., 2014. *Supply Chain Management: Processes, Partnerships, Performance*. 4th ed. Ponte Vedra Beach, Florida: Supply Chain Management Institute.

Li, C., 2009a. Java Database Connectivity. In: *L. Liu and M.T. Özsu, eds. Encyclopedia of Database Systems*. Boston, MA: Springer US, pp. 1577–1578.

Li, C., 2009b. Open Database Connectivity. In: *L. Liu and M.T. Özsu, eds. Encyclopedia of Database Systems*. Boston, MA: Springer US, pp. 1977–1978.

Li, Y. and Manoharan, S., 2013. A Performance Comparison of SQL and NoSQL Databases. In: *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*: IEEE, pp. 15–19.

Marakas, G.M., 2003. *Decision Support Systems in the 21st Century*. 2nd ed. Upper Saddle River, NJ: Prentice Hall.

MathWorks, 2020. *Database Toolbox* [online]. Available at: <https://de.mathworks.com/help/database/index.html> [Accessed 19 May 2020].

Meier, A. and Kaufmann, M., 2019. *SQL & NoSQL Databases*. Wiesbaden: Springer Fachmedien.

Moniruzzaman, A.B.M. and Hossain, S.A., 2013. NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison. *International Journal of Database Theory and Application*, 6 (4), pp. 1–11.

North, K. and Maier, R., 2018. Wissen 4.0 – Wissensmanagement im digitalen Wandel. *HMD Praxis der Wirtschaftsinformatik*, 55 (4), pp. 665–681.

Rahman, F.A., Desa, M.I., and Wibowo, A., 2011. A Review of KDD-Data Mining Framework and Its Application in Logistics and Transportation. In: *Y. Cho, S. Kawata, and F. Ko, eds. 2011 7th International Conference on Networked Computing and Advanced Information Management (NCM)*. Piscataway: IEEE, pp. 175–180.

RapidMiner Inc., 2020. *Database Connectors* [online]. Available at: <https://docs.rapidminer.com/latest/studio/connect/database/> [Accessed 19 May 2020].

Reinsel, D., Gantz, J., and Rydning, J., 2018. The Digitization of the World – From Edge to Core.

Rellensmann, T., 2019. Data Mining-Werkzeuge und ihre Schnittstellen zu Datenbankmanagementsystemen. Project Thesis. Technical University Dortmund, Faculty Mechanical Engineering, Department ITPL.

Rowley, J., 2007. The Wisdom Hierarchy: Representations of the DIKW Hierarchy. *Journal of Information Science*, 33 (2), pp. 163–180.

Runkler, T.A., 2010. *Data Mining: Methoden und Algorithmen intelligenter Datenanalyse*. Wiesbaden: Vieweg+Teubner.

Sadalage, P.J. and Fowler, M., 2013. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Upper Saddle River, NJ: Addison-Wesley/Pearson.

Scheidler, A.A., 2017. *Methode zur Erschließung von Wissen aus Datenmustern in Supply-Chain-Datenbanken*. Göttingen: Cuvillier.

Serdarasan, S., 2013. A Review of Supply Chain Complexity Drivers. *Computers & Industrial Engineering*, 66 (3), pp. 533–540.

solid IT GmbH, 2020. *DBMS Popularität pro Datenbankmodell: Vollständiger Trend, beginnend mit Jänner 2013* [online]. Available at: <https://db-engines.com/en/ranking_categories> [Accessed 5 Feb 2020].

Strozzi, C., 2017. *NoSQL: A Relational Database Management System* [online]. Available at: <http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/NoSQL/Home%20Page> [Accessed 4 May 2020].

Teniwut, W.A. and Hasyim, C.L., 2020. Decision Support System in Supply Chain: A Systematic Literature Review. *Uncertain Supply Chain Management*, pp. 131–148.

Turban, E. and Volonino, L., 2011. *Information Technology for Management: Improving Strategic and Operational Performance*. 8th ed. Hoboken, N.J.: John Wiley.

Weskamp, M., Tamas, A., Wochinger, T., and Schatz, A., 2014. *Einsatz und Nutzenpotenziale von Data Mining in Produktionsunternehmen*. Stuttgart: Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA.

Wrobel, S., et al., 1996. User Interactivity in Very Large Scale Data Mining. In: *W. Dilger, et al., eds. Beiträge zum 9. Fachgruppentreffen Maschinelles Lernen der GI Fachgruppe 1.1.3. Chemnitz: Technische Universität Chemnitz-Zwickau*, pp. 125–130.