Heterogeneous data source integration for smart grid ecosystems based on metadata mining

Juan I. Guerrero, Antonio García, Enrique Personal, Joaquín Luque, Carlos León

Electronic Technology Department, EPS, University of Seville, C/ Virgen de Africa 7, 41011, Seville (Spain)

Keywords:

Smart grids
Large-scale integration
Data mining Standards
Metadata mining
Big data

ABSTRACT

The arrival of new technologies related to smart grids and the resulting ecosystem of applications and management systems pose many new problems. The databases of the traditional grid and the various initiatives related to new technologies have given rise to many different management systems with several formats and different architectures. A heterogeneous data source integration system is necessary to update these systems for the new smart grid reality. Additionally, it is necessary to take advantage of the information smart grids provide. In this paper, the authors propose a heterogeneous data source integration based on IEC standards and metadata mining. Additionally, an automatic data mining framework is applied to model the integrated information.

1. Introduction

The traditional systems in power distribution grids usually have databases with different data structure. The new technologies related to Smart Grids have provided the opportunity of new and advanced functions. Although these new systems are based on the usage of sensor networks and information systems, the systems need the information from older systems, integrating information from heterogeneous data sources. In this sense, there are several problems which need to be solved:

- Information integration. The new systems need to take advantage of old and new data sources. Thus, the integration of these heterogeneous data sources is very difficult, because each database has their own structure. This data source should be translated to a common format. In this way, the information standards provide a good source for a Common Information Models (CIM).
- Data model definition incomplete. The data structures and models of relational databases are not often completely implemented. Frequently, there are several things lacking in the database structure: foreign keys, primary keys, constraints of specific columns, etc. The lack of any of these components makes it more difficult to understand stored information, although, these lacks make the implementation of interfaces easier, because, for example, the joining of tables can be performed in queries.
- Understanding of database models. Each system involved in power distribution grids usually has a different structure: charging management for electrical vehicles (Richardson, Flynn, & Keane, 2012; Sousa, Morais, Vale, Faria, & Soares, 2012), energy management systems for buildings (La, Chan, & Soong, 2016; Wang, Wang, & Yang, 2012), and distribution systems (Zidan & El-Saadany, 2012). The use of information standards simplifies the understanding information stage, in any process of system, data mart or modelling development. The information standards provide a CIM to store all information about the power grid and management systems, for example: International Electrotechnical Committee (IEC) with 61,970, 61,968 and 62,325, and Distributed Management Task Force (DMTF).
- Evolution of technologies. Currently, the development of new technologies is faster than the market's ability to apply them, being more evident in the electrical distribution field. Particularly, the technologies related to information management developed for power distribution companies needs to evolve the systems to take advantage from the new functionalities.
- Modelling information. The new technologies based on Smart Grid systems increase the volume of databases. These databases need powerful algorithms to model the information. Additionally, the information from older system provides several references in order to evaluate the impact of these new technolo-

Corresponding author.
E-mail addresses: juaguealo@us.es (J.I. Guerrero), antgar@us.es (A. García), epersonal@us.es (E. Personal), jluque@us.es (J. Luque), cleon@us.es (C. León).

gies, i.e. by means of Key Performance Indicators (KPI), or to get better models.

In this paper, a novel method to solve these problems is proposed. The system is based on the automatic characterization of metadata in order to discover structural and semantic relationships which were previously unknown. Additionally, this process discovers information about parameters and their patterns in order to establish the corresponding level of importance. This definition is very similar to data mining concept. Thus, this process is named metadata mining. The system includes several data mining tools to model information for classification, outlier detection, pattern detection, forecasting, or information retrieval based on the level of importance established by metadata mining process. Although, the process could get a low success ratio with some models, the results of this process will help in understanding the information and focus the manual modelling, decreasing the economic and time cost of the modelling process.

The following section gives a bibliographical review of some references related to the topic. Next, the metadata mining process is described with the characterization of each entity on the data source. Additionally, the integration process and data mining application are described. Finally, an application to a case in the power sector is shown.

2. Bibliographical review

The main research related to metadata mining applies to documents (Campos & Silva, 2000) and multimedia contents (Wong, 1999). The goal of these studies is knowledge discovery (Yi, Sundaresan, & Huang, 2000) or content classification (Yi & Sundaresan, 2000). Additionally, there are many references about usage of metadata over several types of contents: (Şah & Wade, 2012) proposed a novel automatic metadata extraction framework, which is based on a novel fuzzy based method for automatic cognitive metadata generation and uses different document parsing algorithms to extract rich metadata from multilingual enterprise content. (Asonitis, Boundas, Bokos, & Poulos, 2009) proposed an automated tool for characterizing news video files, using metadata schemas.

Alemu and Stevens (2015) proposed an efficient metadata filtering in order for users to effectively utilise metadata and thus enhance the findability and discoverability of information objects. Fermoso et al. (2009) proposed a new software tool called XDS (eXtensible Data Sources) that integrates data from relational databases, native XML databases, and XML documents. This framework integrates all information from heterogeneous databases to a XML-based format, such as MODS (Metadata Object Description Schema).

Models and algebras are proposed by some references, in order to provide tools for heterogeneous data integration. For example, in the case of models, Liu, Liu, Wu, and Ma (2013) proposed a Heterogeneous Data Integration Model (HDIM) based on the comparison and analysis of the current existing data integration approaches. On this HDIM, a pattern-mapping-based system called UDMP is designed and implemented. This approach tries to improve the rapid development of the Internet of Things (IoT), and Lu and Song (2010) proposed a heterogeneous data integration for smart grids. The authors described a model based on XML and ontology combined with cloud services to solve the heterogeneous problem from the syntax and semantics. The authors tested with Supervisory Control and Data Acquisition (SCADA) data to validate the model.

In the case of algebras, Tang, Zhang, and Xiao (2005) proposed a capability object conceptual model to capture a rich variety of query-processing capabilities of sources and outline an algebra to compute the set of mediator-supported queries based on the capability limitations of the sources they integrate. This algebra is used in several references.

Additionally, there are a number of studies and research related to heterogeneous data integration based on, for instance, XML (Fengguang, Xie, & Liqun, 2009) (Su, Fan, & Li, 2010) (Lin, 2009), Lucene and XQuery (Tianyuan, Meina, & Xiaoqi, 2010), and OGSA-DAI (Gao & Xiao, 2013). In the same way, heterogeneous data integration has applications to many areas, such as Livestock Products Traceability (Chen & Liu, 2009), safety production (Han, Tian, & Wu, 2009), management information systems (Hailing & Yujie, 2012), medical information (Shi, Liu, Xu, & Ji, 2010), and web environments (Fan & Gui, 2007).

There are also examples of the application of data mining mixed with heterogeneous data source integration. Cao, Chen, and Jiang (2007) proposed a framework of a self-Adaptive Heterogeneous Data Integration System (AHDIS), based on ontology, semantic similarity, web service and XML techniques, which can be regulated dynamically. Merrett (2001) used OLAP and data mining to illustrate the advantages for the relational algebra of adding the metadata type attribute and the transpose operator.

In relation with the application of data mining techniques automatically over integrated information, Li, Kang, and Gao (2007) proposed a high-level knowledge modelled by ordinary differential equations (ODEs) discovered in dynamic data automatically by an Asynchronous Parallel Evolutionary Modelling Algorithm (APHEMA). The data mining techniques are mainly used to forecast parameters. Hoiles and Krishnamurthy (2015) proposed a nonparametric demand forecasting based on Least Squares Support Vector Machine (LS-SVM). Chen, Li, Lau, Cao, and Wang (2010) proposed detecting automated load curve data cleansing based on the B-Spline smoothing and Kernel smoothing to automatically cleanse corrupted and missing data.

Some commercial-strength DataBase Management Systems (DBMS) and their On-Line Analytical Processing (OLAP) extensions provide very good solution to model information. However, all these applications implement solutions for modelling with supervision of an expert user. This software cannot make an automatic modelling because the software does not have any information about the problem or the information stored in the database. The proposed metadata mining method provides this information, determines what parameters or columns are a possible objective of the data mining model, and what the best technique is to get a good model.

3. General description

The proposed system provides a solution for the described problems, these problems are related to heterogeneous data sources and data analysis from smart grids ecosystems. The primary advantages of this solution are:

- The integration of information can be performed over any relational data source.
- The deployment of a smart grid ecosystem or a specific system in a smart grid is quicker because the system designs a specific ETL (Extract, Transform, and Load) for the new system based on standard information.
- The integration of information can be optionally stored in a data warehouse, with star or snowflake structure.
- The integration of information from different systems in a smart grid can be performed by the proposed system.
- The integration process can be applied in any distributed system with a high security level because the system only uses metadata.

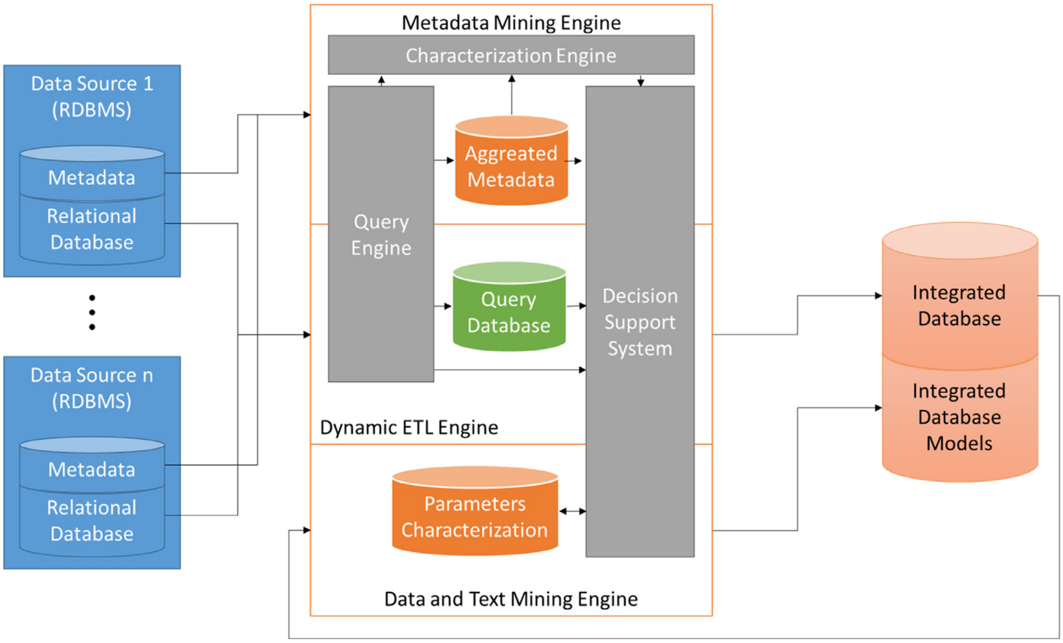

Fig. 1. Flow and architecture overview.

- The information of metadata mining can be used to optimize the original data sources.
- The process provides basic models for each parameter identified in the data sources, using different data mining and text mining techniques.

The information flow and architecture is shown in Fig. 1. The metadata from data sources is gathered by the metadata mining engine using the query engine. The metadata are characterized and classified by means of a Decision Support System. The Decision Support System (DSS) has several rules that are based on the indicators generated in the metadata mining process and the results of queries. The DSS has 492 rules: 30 rules in Metadata Mining Engine, 352 rules in Dynamic ETL Engine, and 110 rules in Data and Text Mining Engine. Each of these rules has been obtained from experience in collaboration in around 20 research projects with utility companies. The common problem in these projects is the existence of different relational data sources (95% were relational databases), with different: data management systems, data model, scope, and, often, without defined foreign keys. The 30 rules in Metadata Mining Engine deal with technical metadata. The 352 rules in Dynamic ETL Engine deal with technical and informational metadata to create and run the ETL. These rules could be classified into:

- Dynamic rules. The antecedent and consequence of a dynamic rule is stored on a table. This really means that each dynamic rule is applied several times, depending of the coincidences between available information and the data stored in the dynamic rule antecedent. In this sense, several sets of rules could be identified:
  - ∘ 95 rules deal with IEC Common Information Model (CIM).
  - ∘ 83 rules deal with DMTF CIM.
  - ∘ 32 rules deal with IEC CIM extensions.
  - ∘ 36 rules deal with DMTF CIM extensions.
  - ∘ 53 rules deal with constraints.
  - ∘ 33 rules deal with foreign constraints.
- Static rules. These rules only have one antecedent and consequence. There are 20 rules which treat general topics to create and run the dynamic ETL.
- The 110 rules in Data and Text Mining Engine could be classified in:
- 96 dynamic rules deal with the selection and application of the most adequate method for each modelling process, according to technical and informational metadata and the characterization performed.
- 14 static rules deal with the analysis of the results of modelling methods applied.

The static and dynamic rules were generated from experience on several research projects related with utilities and Smart Grids. These projects are related to Smart Grids (Personal, Guerrero, Garcia, Peña, & Leon, 2014), non-technical losses detection (Guerrero et al., 2011) and (Guerrero et al., 2016a), Smart Grid ecosystem integration (Guerrero, Personal, Parejo, García, & León, 2016b), etc. The first prototype of this framework only has a semi-automatic process to integrate tables, and the researcher manually modifies it. After several applications of this prototype, several configuration rules were extracted and the main structure of rules (static and dynamic rules) was designed. The proposed solution tries to integrate all information from all provided data sources. If the proposed solution cannot integrate any part of any data source, it includes the information and trace it to manually define new rules for these new data sources. The proposed solution is the result of several iterations and the automatic generation of rules is in development, but was not applied for this solution.

When the system has classified all metadata from all data sources, the Dynamic ETL Engine performs the integration. There are two possibilities: according to an information standard or data warehouse (star or snowflake structure). If the user requires it, the integrated information can be modelled by the Data and Text Mining Engine. This engine performs an analysis according to the metadata mining information, in order to obtain the best model for each selected parameter. Thus, following the configuration provided by the user, the system gathers all metadata from data sources and provides:

- A database with integrated information in a specific format.
- A database with different models for each parameter identified.

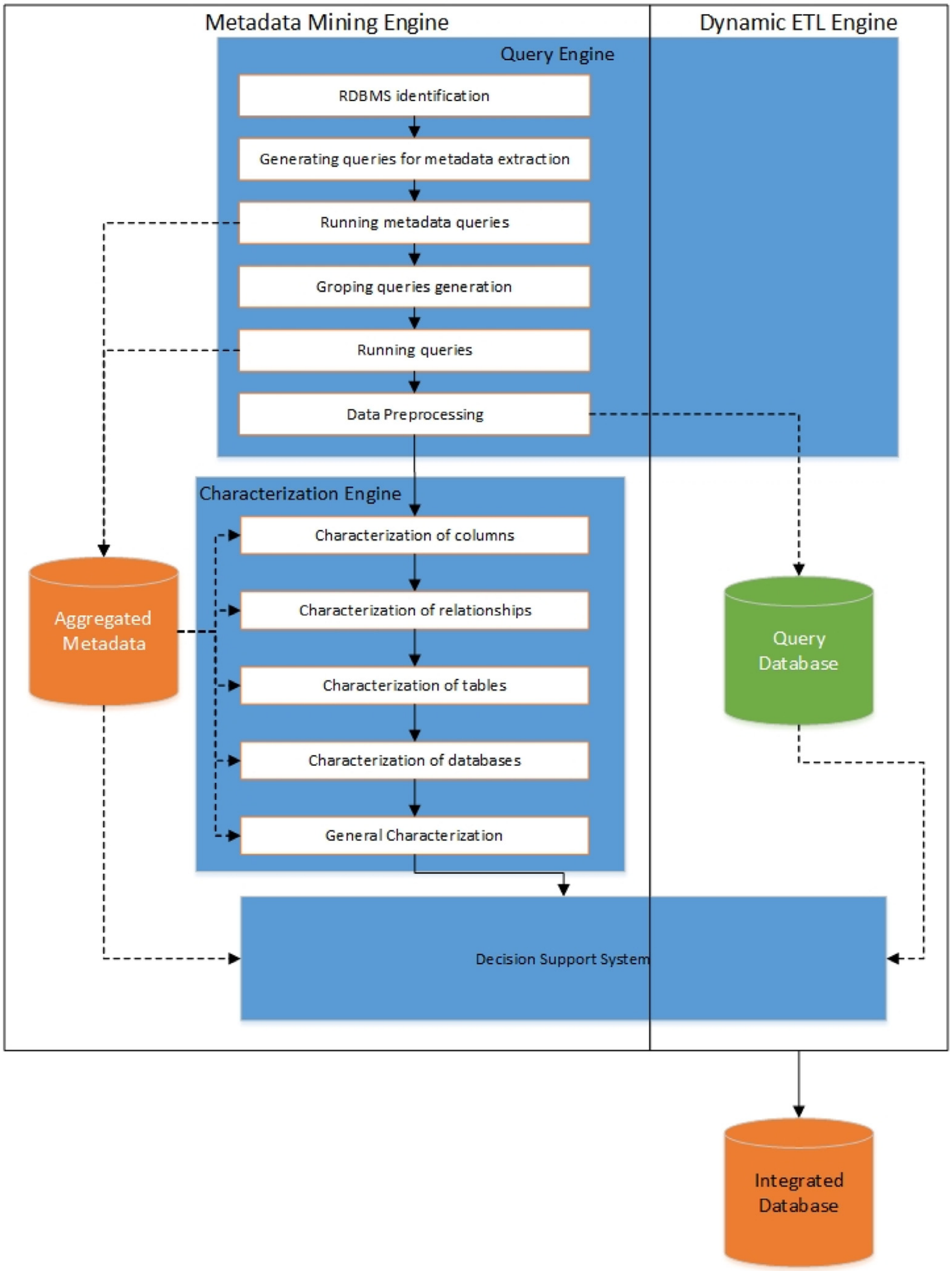

Fig. 2. Metadata mining process flow chart.

4. Metadata mining

The metadata mining process is based on the metadata in relational databases. Currently, the method has been successfully applied in several databases related to power distribution, power consumption, energy efficiency, health, and laboratory databases. The metadata mining methodology is the same in all these cases. The flow diagram is shown in Fig. 2. In the case of relational databases, this methodology has several steps that are shown in Fig. 2.

4.1. Relational database identification and metadata extraction

The proposed system has been tested with relational databases: MySQL, IBM DB2, Oracle Database, PostgreSQL, Microsoft SQL Server, and HBase. The identification of the relational database management system provides:

- Query language.
- Specific considerations about the RDBMS (Relational DataBase Management System).
- The name and structure of system tables.

Several queries to extract system tables are performed according to the database identified. These queries are automatically generated according to the identified RDBMS. The system provides two options to perform the queries:

- In system in which the RDBMS is not directly accessible, the system provides a sql script. The user has to run this script in the command line of the RDBMS. The results of the script usually are several text files (one per system table) and these text files are loaded in the system. This information is pre-processed in order to correct mistakes and format errors.
- The system runs the queries through a connection with RDBMS. The pre-processing step is easier than in the other case because the direct connection reduces the mistakes and errors in the interpretation of extracted information.

This process was simultaneously applied to several data sources.

4.2. Execution of grouping queries

The grouping queries are executed for each column of each table to obtain information regarding:

- The different values of the column.
- The frequency of each value.
- The absolute and relative frequencies of each value.
- Different statistical information about distribution of values.

This information is mixed with information from the previous step if there is statistical information regarding the column in the system tables. Occasionally, the statistical information in the system tables is empty because the database was not analysed by RDBMS tools.

The results of grouping queries are stored in different tables. For example, the grouping query for column$_{u,k,j}$ in table$_{u,k}$ from data source u (ds$_u$) in standard Statement Query Language (SQL) was

SELECT column$_{u,k,j}$, COUNT(*) AS counter
FROM table$_{u,k}$
GROUP BY column$_{u,k,j}$
ORDER BY column$_{u,k,j}$;

ds$_u$table$_k$column$_{k,j}$	
column$_{u,k,j}$	**counter**
value$_{u,k,j,i}$	v$_{u,k,j,i}$

Where, u is the data source index, k is the table index, j is the column index and i is the record index of ds$_u$table$_k$column$_{k,j}$. This query provides a table with two columns: column$_{k,j}$ and counter. The first column contains all the possible values of the column (value$_{u,k,j,i}$). The second column contains the number of the records which has the corresponding value. This table is stored in the target database. The name of the table will be a combination of the table and column names: ds$_u$table$_k$column$_{k,j}$.

These queries are separately performed on each column because of data protection laws. The execution of the queries in each column avoids crossing the data and obtaining the original register or original data source. However, these results provide enough information to know the possible values of column and the pattern or distribution of values.

4.3. Characterization process

The characterization process is executed in several stages by characterization engine (Fig. 2). In each characterization several key indicators are generated. These indicators will be used by DSS in order to stablish the semantic and structural relationships, performing the integration of information. Additionally, these key indicators also specify the available parameters for a Data and Text Mining Engine.

4.3.1. Characterization of columns

The characterization of a column depends on the data type. The first step is to classify the column in one of these categories: Numerical, Text, Timestamp, Object, Binary, and Other.

Each category is characterized according to different indexes and statistical information. These categories have the calculation of some indexes in common:

- Number of different values (TNV$_{u,k,j}$): This parameter is the number of records of ds$_u$table$_k$column$_{k,j}$.
- Total Number of Records (TNR$_{u,k,j}$): The total number of records in the original table. This value must be the same for all column$_{u,k,j}$ from table$_{u,k}$.

$$TNR_{u,k,j} = \sum_{i=0}^{NDV_{u,k,j}} v_{u,k,j,i}$$

- Analysis of null value:
  - Number of records with null value (NNV$_{u,k,j}$) is obtained by a query: *SELECT counter FROM table$_k$column$_{k,j}$ WHERE isNull(column$_{u,k,j}$)*; If query do not return any value NNV$_{u,k,j}$ will be 0.
  - Null value frequency (N$_{u,k,j}$). Number of records with null values divided by total number of records.

    $$N_{u,k,j} = \frac{NNV_{u,k,j}}{TNR_{u,k,j}}$$

  - Null value weight (NVW$_{u,k,j}$). *If NNV$_{u,k,j}$ > 0 then NVW$_{u,k,j}$ = 1/TNV$_{u,k,j}$ else NVW$_{u,k,j}$ = 0.*
- Analysis of blank value:
  - Number of records with Blank Value (NBV$_{u,k,j}$) is obtained by a query: *SELECT counter FROM table$_k$column$_{k,j}$ WHERE isBlank(column$_{u,k,j}$)*; If query do not return any value NBV$_{u,k,j}$ will be 0.
  - Blank value frequency (B$_{u,k,j}$). Number of records with blank values divided by total number of records. $B_{u,k,j} = \frac{NBV_{u,k,j}}{TNR_{u,k,j}}$
  - Blank Value Weight (BVW$_{u,k,j}$). If NBV$_{u,k,j}$ is greater than 0 then BVW$_{u,k,j}$= 1/TNV$_{u,k,j}$ else BVW$_{u,k,j}$ will be 0.
- Analysis of default value: the default values is extracted from metadata of table k:
  - Number of records with Default Value (NDV$_{u,k,j}$) is obtained by a query: *SELECT counter FROM table$_k$column$_{k,j}$ WHERE column$_{u,k,j}$= default_value*; If query does not return any value NDV$_{u,k,j}$ will be 0.
  - Default value frequency (D$_{u,k,j}$). Number of records with default values divided by total number of records. This index is calculated only if the default value is included in the constraints of the table.

    $$D_{u,k,j} = \frac{NDV_{u,k,j}}{TNR_{u,k,j}}$$

  - Default Value Weight (DVW$_{u,k,j}$). If NDV$_{u,k,j}$ is greater than 0 then DVW$_{u,k,j}$= 1/TNV$_{u,k,j}$ else DVW$_{u,k,j}$ will be 0
- Analysis of other values:
  - Relative Useful (RU$_{u,k,j}$). Number of different useful values divided by number of different values. No useful values are: blanks, nulls, and defaults.

    $$RU_{u,k,j} = NDV_{u,k,j} - \left(NV_{u,k,j} + BV_{u,k,j} + DV_{u,k,j}\right)$$

- Absolute Useful ($AU_{u,k,j}$). This indicator is calculated according to the value of previous indicators:

  $AU_{u,k,j} = 1 - -\left(NNV_{u,k,j} + NBV_{u,k,j} + NDV_{u,k,j}\right)$.
- Value Frequency ($VF_{u,k,j,i}$). For each value:

  $VF_{u,k,j,i} = v_{k,j,i}/TNR_{u,k,j}$
- Value Weight ($VW_{u,k,j,i}$). If $v_{u,k,j,i}$ is greater than 0 then $VW_{u,k,j,i}$= $1/TNV_{u,k,j}$ else $VW_{u,k,j,i}$ will be 0.

- Enumerable. This index identifies the number of identified categories in the values of the column. This is very common in discretized information or in parametric information. A value of zero determines a column with continuous values.
- Formatted. This indicator determines if there exists any format in the column values. If the column stored numerical or time information, the format of this column will be extracted from metadata information. If the format of the values is not specified in metadata, the format is inferred by values. Usually, in this case, a text data type is used in the column, and the value should be a code, for instance: serial number, identification code, etc. The $value_{u,k,j,i}$ is processed character by character, each number is replaced by N, each letter is replaced by L, and any symbol or special character (space character included) is replaced S.

The profile for numerical columns contains information about data type, length, precision, column description, and constraints. Some statistical information is calculated: histograms, maximum, minimum, standard deviation, average, median, mode, and variation coefficient.

The profile of text columns contains information about data type, length, char set, column description, and constraints. Some statistical information is calculated: histograms, maximum length, minimum length, average length, standard deviation length, maximum number of words, minimum number of words, and average word length. Additionally, a dictionary is generated using text mining techniques. This dictionary is used to calculate the relationship coefficient with each column. The text mining technique attempts to elicit the text field concepts, structured or otherwise. A concept can comprise one or more words which represent an entity (e.g., action, and event). Natural Language Processing (NLP) methods are used to extract linguistic (e.g., words and phrases) and non-linguistic (e.g., dates and numbers) concepts. An interesting review of this technique and its use in information management systems is proposed in Métais (2002). The following set of functionalities are included:

a. Recognition of punctuation errors. These types of mistakes include the incorrect use of the tilde, the period, the comma, the point and comma, the dividing bar, etc.
b. Recognition of spelling errors. A grouping fuzzy technology is applied. When concepts of the text are extracted, words with similar spelling (referring to the letters that compose it) or that are closely related are classified together. By applying this algorithm, mistakes of omission of letters, duplication of letters, or permutation of letters are corrected. This algorithm is used in the fuzzy relationship coefficient with each column calculation.

Although these mistakes are corrected before storing the concept in the dictionary, they are registered in the system in order to establish the level of wording of the column.

The profile for timestamp columns contains information about data type, format, column description, and constraints. Some statistical information is calculated: histograms, minimum, maximum, average time period between records, minimum time period between records, maximum time period between records, values with the maximum number of records, values with the minimum number of records, average number of records per value, standard deviation of number of records per value, and vales with the nearest number of records to average number of records per value. Additionally, a histogram of the number of records per value is created. This histogram is normalized from 0 to 1, dividing the number of records in each value by total number of records. This information is used to calculate the relationship coefficient with each column.

The profile of object columns is used when the column contains information in a specific datatype defined in the RDBMSs. These data types are composed by different primitive types. If the system table contains information about this data type (sometimes this information is not accessible) the system associates several profiles to the column one per primitive type, generating all the information previously described in each profile. Arrays are classified in this category.

The profile of a binary column is used when the data type of a column stores binary information, for example images, documents, etc. Currently, the metadata mining only classifies the type of contents into the following categories:

- Images: if the stored information is about image files
- Documents: if the stored information is about text file documents
- Video: if the stored information is about video files
- Technical: if the stored information is about technical files
- Other: if the information is not classified in the previous categories or is encrypted information.

The profile of other columns is used when the column cannot be classified in the categories above. Normally, these columns are not used in the metadata mining process, and they are manually handled in order to establish a new profile. The encrypted columns are usually classified in this category.

4.3.2. Characterization of relationships

The characterization of relationships is based on the constraint stored in metadata, and the similarity between registered values of columns. In the second case, several coefficients are calculated studying the column name and the values of columns:

- Fuzzy relationship coefficient with each column. This is an array of indexes, one per column in the database. Each element of this array establishes the relationship between different fields according to the name of the column. The index calculation is based in the application of a fuzzy algorithm to match the column name with other column names. The index can have a value between 0 and 1; zero indicates that there isn't any relationship, and one indicates that the columns are related. This algorithm was described previously, but additionally some rules in the DSS are used to detect some concepts or terms.
- Relationship coefficient with each column. This is an array of indexes, one per column in the database. Each element of this array establishes the relationship between different columns according to the registered values. First the algorithm compares the data type, then the values.
- Cardinality. The cardinality of relationship is calculated for each column. This cardinality is calculated based on the constraint stored in metadata and in the results of relationships coefficients previously described.

4.3.3. Characterization of tables

Each table on the selected data source is classified in one of the following categories:

- Parametric information table. The tables of this category contain indexed information about different characteristics. For example, the statistical classification of economic activities in the European Community (NACE) can be used in several tables, and it could have several columns: ID, SECTOR, SUBSECTOR, CODE, and DESCRIPTION. In this way, only using a reference to ID it is possible to obtain all information using a join query.
- Entity information table. These tables contain information about different entities in the system, for example: contracts, equipment, etc.
- Personal information table. These tables contain personal information that could require privacy protection.
- Historical information table. These tables are characterized by the utilization of timestamp columns, and they show any regularity in these columns. Some examples include historical data about consumption, historical data about tasks, etc.
- Complementary information table. These tables contain additional information for entity, personal, or historical information tables.
- Bridge table. This table category identifies tables that are only composed of indexed columns and are usually bridges between several tables. This is not a good practice in database structure definition, but it is possible to find some of these cases.
- Orphan table. This category represents the tables that did not show any relationship with other column tables.
- Dummy table. This category represents tables that cannot be classified in previously defined categories.

These categories are established by experience in several utility-related projects. Notwithstanding when the system finds any table which cannot be classified into these categories it is classified as a dummy table. In this case, an expert user could review the results (described in section V) and create a new category if it is necessary.

Additionally, some indicators are calculated for each table:

- Total Number of Tables of Data Source ($\mathrm{NTDS}_{\mathrm{u}}$, where u is the data source identification).
- Number of Table Columns ($\mathrm{NTC}_{\mathrm{u,k}}$, where k is the table identification from data source u).
- Number of Related Tables of data source ($\mathrm{NRT}_{\mathrm{u,k}}$). This number is calculated for each table k from data source u. The number of related tables includes the relationships with a high value in relationship coefficients (calculated in column characterization).
- Number of Columns with high rate of Relationship ($\mathrm{NCR}_{\mathrm{u,k}}$), this number is calculated for each table k from data source u. This number includes the columns with a high value in the relationship coefficients (calculated in column characterization).
- Number of Primary Key ($\mathrm{NPK}_{\mathrm{u,k}}$). Number of primary keys in the table k from data source u.
- Number of Self-Relationships ($\mathrm{NSR}_{\mathrm{u,k}}$). Number of self-relationships in table k from data source u.
- Table Relationship Indicator ($\mathrm{TRI}_{\mathrm{u,k}}$). The number of related tables divided by total number of tables in data source.

$$TRI_{u,k} = \frac{NRT_{u,k}}{NTDS_{u,k}}$$

- Column Relationship Indicator ($\mathrm{CRI}_{\mathrm{u,k}}$). This indicator is calculated for each table k from data source u. Number of columns with foreign or primary keys (this includes the columns with a high rate in the relationships index) divided by the total number of columns in the table.

$$CRI_{u,k} = \frac{NCR_{u,k}}{NCT_{u,k}}$$

- Key Indicator ($\mathrm{KI}_{\mathrm{u,k}}$). Number of primary keys divided by the total number of columns in the table.

$$KI_{u,k} = \frac{NPK_{u,k}}{NCT_{u,k}}$$

- Self-Relationship Indicator ($\mathrm{SRI}_{\mathrm{u,k}}$). Indicates the number of recursive relationships.

$$SRI_{u,k} = \frac{NSR_{u,k}}{NCT_{u,k}}$$

4.3.4. Characterization of data source

The characterization of the data source determines the coherence and reliability of the stored information, and it establishes the different indicators that will be used in automatic application of data mining techniques. These techniques try to establish models for prediction and classification of information. Additionally, the characterization includes information for automatic integration with other data sources.

- Total Number of Tables ($\mathrm{TNT}_{\mathrm{u}}$).
- Total Number of Columns ($\mathrm{TNC}_{\mathrm{u}}$). Total number of columns in all tables of the data source u.

$$TNC_u = \sum\nolimits_{k=1}^{TNT_u} NCT_{u,k}$$

- Database malleable indicator. This indicator establishes the potential for data analysis based on the information stored in the database. The number of columns with a high rate of useful information (columns with any possibility of application of any data mining technique) plus columns without useful information but with a high correlation coefficient with useful columns divided by the total number of columns.
- Database time analysis indicator. This indicator establishes the potential of temporal analysis. The calculation of this indicator is very similar to the “Database malleable indicator”. This indicator considers as useful columns those columns with any possibility of application of any time analysis technique.
- Database classification analysis indicator. This indicator establishes the potential of application of classification and clustering techniques. The calculation of this indicator is very similar to the “Database malleable indicator”. This indicator considers as useful those columns with any possibility of application of any classification or clustering technique.
- Database forecasting analysis indicator. This indicator establishes the potential of application of forecasting techniques. The calculation of this indicator is very similar to the “Database malleable indicator”. This indicator considers as useful those columns with any possibility of application of any forecasting technique.
- Database text analysis indicator. This indicator establishes the potential of application of text mining techniques. The calculation of this indicator is very similar to the “Database malleable indicator”. This indicator considers as useful those columns with any possibility of application of any text mining technique.
- Cohesion indicator. This indicator shows the information cohesion. The orphan records and tables are used to calculate this indicator. Additionally, if statistical information about the database is available, then this indicator is modified adding columns without queries.
- Replication indicator. This indicator shows the level of redundant information.

Table 1
Results of characterization of tables.

Data source	Table name	Classification	Relationship indicator	Column Rel. Ind.	Key Ind.	Auto-rel.	Number of cols.	Number of regs.
A	EconomicActivity	PARAMETRIC	0.25	0.17	0.17	0	6	996
A	Contract	ENTITY	0.5	0.13	0.06	0	16	11
A	HistoricalMeasures	HISTORICAL	0.5	0.33	0.33	0	6	11,037,600
A	SourceType	COMPLEMENTARY	0.25	0.17	0.17	0	6	21
B	EconomicActivity	PARAMETRIC	0.14	0.17	0.17	0	6	996
B	Contract	ENTITY	0.57	0.24	0.06	0	17	4
B	HistoricalRecharging	HISTORICAL	0.29	0.4	0.4	0	5	840,960
B	VehicleData	PERSONAL	0.29	0.25	0.13	0	8	4
B	Tariff	PARAMETRIC	0.29	0.25	0.13	0	8	12
B	RechargingStation	COMPLEMENTARY	0.29	0.14	0.07	0	14	3
B	Connector	COMPLEMENTARY	0.29	0.17	0.17	0	6	24
C	PowerResource	COMPLEMENTARY	0.33	0.11	0.11	0	9	3
C	HisMeasures	HISTORICAL	0.67	0.6	0.6	0	5	3,784,320
C	SourceType	COMPLEMENTARY	0.33	0.2	0.2	0	5	9

Table 2
Results of characterization of data sources.

Data source	Indicators and coefficients						
	Minable	Time analysis	Classification analysis	Forecasting	Text analysis	Cohesion	Replication
A	0.82	0.60	0.53	0.70	0.05	0.87	0
B	0.93	0.72	0.70	0.50	0.10	0.70	0.30
C	0.75	0.81	0.41	0.68	0.10	0.98	0

4.3.5. General characterization

The general characterization establishes the relationship between all the data sources characterized according to the method previously defined. In this characterization, all the previous steps are repeated, but considering all databases or data sources as the sole database. The new indicators calculated contain values according to all data sources. These new indicators have the prefix 'general'.

5. Decision support system and integration of information

The integration of information from heterogeneous databases is accomplished by the application of general characterization in all classified databases. This module creates queries to integrate all information from columns and tables based on a decision support system based on 352 rules. This Decision Support System is part of a Dynamic ETL Engine and it is based on the information generated in the characterization of metadata mining process and on the results of several queries. The rules provide the queries to build the final query that integrates the information from different tables from different data sources. These queries are packed into ETL according to the target RDBMS. All tables with similar characterization are checked to be grouped according to the calculated cardinality. These new tables are characterized using the process previously described. The new values are compared with the original values in order to check the integration.

An example of these rules that involves several queries is shown below. This rule is used in the characterization of columns, and this rule calculates the cardinality of one side of the relationship. Some queries are performed to calculate it. These queries are:

- *Select count*($_{table\ 1}column_{A,1}.column_{A,1}$) *AS* $count_A$ *from* $_{table\ 1}column_{A,1}$ *where not*($_{table\ 1}column_{A,1}$ *in* (*select* $_{table\ 1}column_{A,1}.column_{A,1}$ *from* $_{table\ 1}column_{A,1}$, $_{table\ 2}column_{B,2}$ *where* $_{table\ 1}column_{A,1}.column_{A,1}={}_{table\ 2}column_{B,2}.column_{B,2}$));
- *Select min*($counter_A$) *AS* min_A, *max*($counter_A$) *AS* max_A, *min*($counter_B$) *AS* min_B, *max*($counter_B$) *AS* max_B *from* (*select* $_{table\ 1}column_{A,1}.column_{A,1}$, $_{table\ 2}column_{B,2}.\mathrm{column}_{\mathrm{B},2}$, *sum*($_{table\ 1}column_{A,1}.\mathrm{counter}$) $counter_A$, *sum*($_{table\ 2}column_{B,2}.\mathrm{counter}$) $counter_B$ *from* $_{table\ 1}column_{A,1}$, $_{table\ 2}column_{B2}$ *where* $_{table\ 1}column_{A,1}.\mathrm{column}_{\mathrm{A},1} = {}_{table\ 2}column_{B,2}.\mathrm{column}_{\mathrm{B},2}$ *group by* $_{table\ 1}column_{A,1}.\mathrm{column}_{\mathrm{A},1}$, $_{table\ 2}column_{B,2}.\mathrm{column}_{\mathrm{B},2}$)

The Decision Support System uses the results of these queries and the calculated index to establish the cardinality of relation between column A of Table 1 and column of Table 2.

```
    If fuzzy_relationship $>=0.5$ or
  relationship_coefficient $>=0.9$ or
  exists defined constraint then
    If ($min_A==max_A$ and $min_A>1$) or
      $min_A<max_A$ then
        (maximum cardinality is N)
    endif
    If ($min_A==max_A$ and $min_A==1$) then
      (maximum cardinality is 1)
    endif
    If $count_A<>0$ then
      (minimum cardinality is 0)
    else
      (minimum cardinality is 1)
    endif
endif
```

Currently, the process of checking the validity of the integration is performed by using several threshold parameters. These parameters are specified by the user or analyst. The automatic threshold parameter adjustment is in the research stage. Additionally, the user can filter orphan tables and bridge tables, and avoid bridge tables.

The system can integrate information in two ways:

- According to the information of characterization. The system has been tested with several data sources. The intelligent ETL engine tries to create databases with star or extended-star architecture, in order to generate a data warehouse.
- According to the information of characterization and an information standard. Currently, the system only works with power distribution information standards. This system has been tested with information related to utilities, energy management, and information systems. The intelligent ETL engine can follow two standards: IEC CIM based on IEC 61,970 and 61,968 or DMTF CIM based on version 2.44.1 (but only applied to power grids).

Currently, the utilization of other standards for health (HL7 and OpenEHR) are in the research stage.

The integration of information includes several tables with information of characterization. This information was generated in metadata mining. The added tables are:

- GEN_CHAR. This table contains one record per data source, and contains information about the calculated indicators and data source description.
- DB_CHAR. This table contains one record per database, and contains information about the calculated indicators and database information. It is associated with data source described in GEN_CHAR.
- TAB_CHAR. This table contains one record per table, and contains information about the calculated indicators, relationship information, and table information. It is associated with data source (GEN_CHAR) and database (DB_CHAR).
- COL_CHAR. This table contains one record per column, and contains information about the calculated indicators, relationship information, and table information. It is associated with data source (GEN_CHAR), database (DB_CHAR), datatype (DT_CHAR), and table (TAB_CHAR)
- CONS_CHAR. This table contains one record per constraint, and contains information about constraints and the associated table and column. It is associated with column (COL_CHAR) and table (TAB_CHAR).
- DT_CHAR. This table contains one record per component of data type, and contains information about data types.

Additionally, the information from the integrated resource is described by similar tables with 'I' prefix: I_DB_CHAR, I_TAB_CHAR, I_COL_CHAR, I_CONS_CHAR, and I_DT_CHAR.

These tables have several additional columns to store information that will be generated in the data mining stage.

6. Data mining

The data mining module is guided by information generated in the characterization stage, supported by a DSS based on 110 rules. In the first place, a feature selection is performed to associate a support index to each column. This feature selection is performed for each column as a target. In this way, each column has one value associated to it.

Currently, a threshold is manually specified to use the different columns and it is based on experience. The variation of this threshold takes effect in the accuracy of data mining results, generating models that could not be useful, with a high error rate, and computational time wasting. Although, the threshold is based on experience it has not been optimized. The value of threshold is set in order to ensure good models, which addressed the studies over the data. Thus, the data analyst could use these models as a starting point. The application of automatic methods for optimization of this threshold is in the research stage and is focused in parametric optimization based on fuzzy techniques and evolutionary computation.

Additionally, according to the characterization performed, several methods are applied to obtain models. This module has been implemented in an SPSS Modeler (*IBM SPSS Modeler 16 Algorithms Guide*, n.d.) and Python (*IBM SPSS Modeler 16 Python Scripting and Automation Guide*, n.d.). In this way, the applied algorithms or techniques are: Anomaly detection (Chandola, Banerjee, & Kumar, 2009), apriori (Agrawal & Srikant, 1994), bayesian network (Pearl, 2000), C5.0,[1] Carma (Hidber, 1999), C&R Tree (Breiman, Friedman, Stone, & Olshen, 1984), Chi-squared Automatic Interaction Detector or CHAID (Kass, 1980), Cluster evaluation (based on silhouette coefficient, sum of squares error or SSE, sum of squares between or SSB, and predictor importance), COXREG (Cox, 1972), Decision List, Discriminant, Factor Analysis (PCA) (Geiger & Kubin, 2012), Generalized Linear Models, Generalized linear mixed models (Madsen & Thyregod, 2010), K-Means (MacQueen, 1967), Kohonen (Kohonen, 1982), Logistic Regression (Freedman, 2005), KNN (Pan, McInnes, & Jack, 1996), Linear modelling (Belsley, Kuh, & Welsch, 2013), neural network (Haykin, 1994), optimal binning (Usama M. Fayyad, 1993), "Quick, Unbiased, Efficient Statistical Tree" or QUEST (Loh & Shih, 1997), linear regression, Sequence, Self-learning response model or SLRMs, support vector machine (SVM), temporal casual modelling algorithms (Arnold, Liu, & Abe, 2007), time series (Box, Jenkins, & Reinsel, 2008), and TwoStep cluster (Chiu, Fang, Chen, Wang, & Jeris, 2001).

The selection and application of techniques is controlled by an implemented Python, based on thresholds over different Metadata Mining parameters. Thus, it is based on two criteria:

- The error rate of each generated method.
- The correlation between the model and the target.

Additionally, the generation of models can be personalized by:

- Specification of time limit in model generation.
- Specification of memory limit in model generation.
- Manual filtering of non-desired targets.
- Establishing a limit in the number of parameters to consider in the modelling process.
- Manual filtering of non-desired algorithms or techniques.

7. Experimental results

The proposed system was applied to several data sources related to power distribution. In different projects related to utilities, this framework has evolved until the framework presented in this paper. There are several problems in the application of this solution in companies. Firstly, the availability of database systems is very low. The commercial databases are busy with general management tasks (billing, memberships, withdrawals, field works, etc.). These tasks spend all the resources during daylight hours. During night hours, the backup process spends more time. It is very difficult to find an availability time window to perform any other task. Secondly, the data protection laws are a very serious issue. These laws ban sharing or crossing the data with other systems (or, in some cases, from other departments) or other companies. Thus, if the integration should be performed by an external system, this system must guaranteed that the data protection laws are accomplished. This means the original information could not be restored in external systems. Thirdly, the accessibility of information in this type of systems is very difficult because of the cybersecurity levels. The data extraction is essential to execute any integration of information. The data extraction stage depends on whether there is a direct connection to the data source. When the data source is protected, and it is not possible to have a direct connection or remote connection. In these cases in the proposed solution the queries are executed by a script generated by the system, and the user runs the script on an authorized client. The script provides several text files with information from each table. The user loads these files onto the system. However, when the system has a direct or remote connection with the data source, the extraction is automatically performed with authorization from the user, and the data protection laws are fulfilled.

[1] http://www.rulequest.com/.

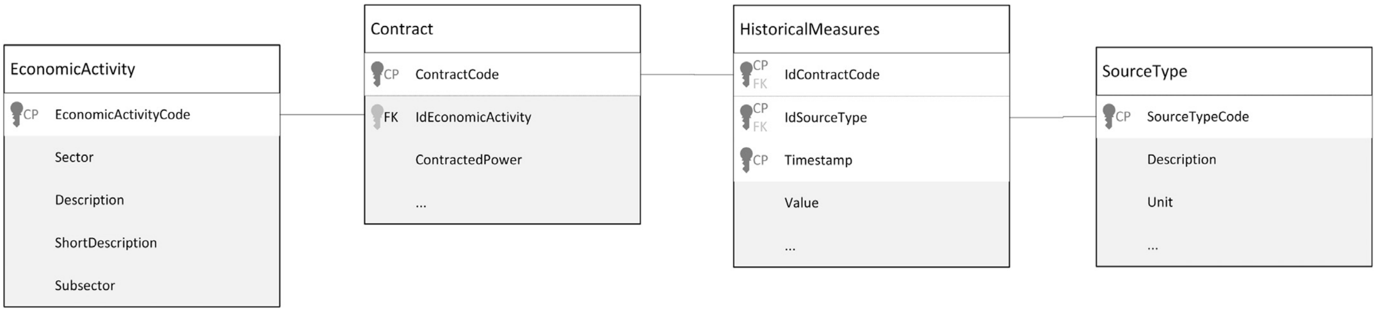

Fig. 3. UML diagram for Source A.

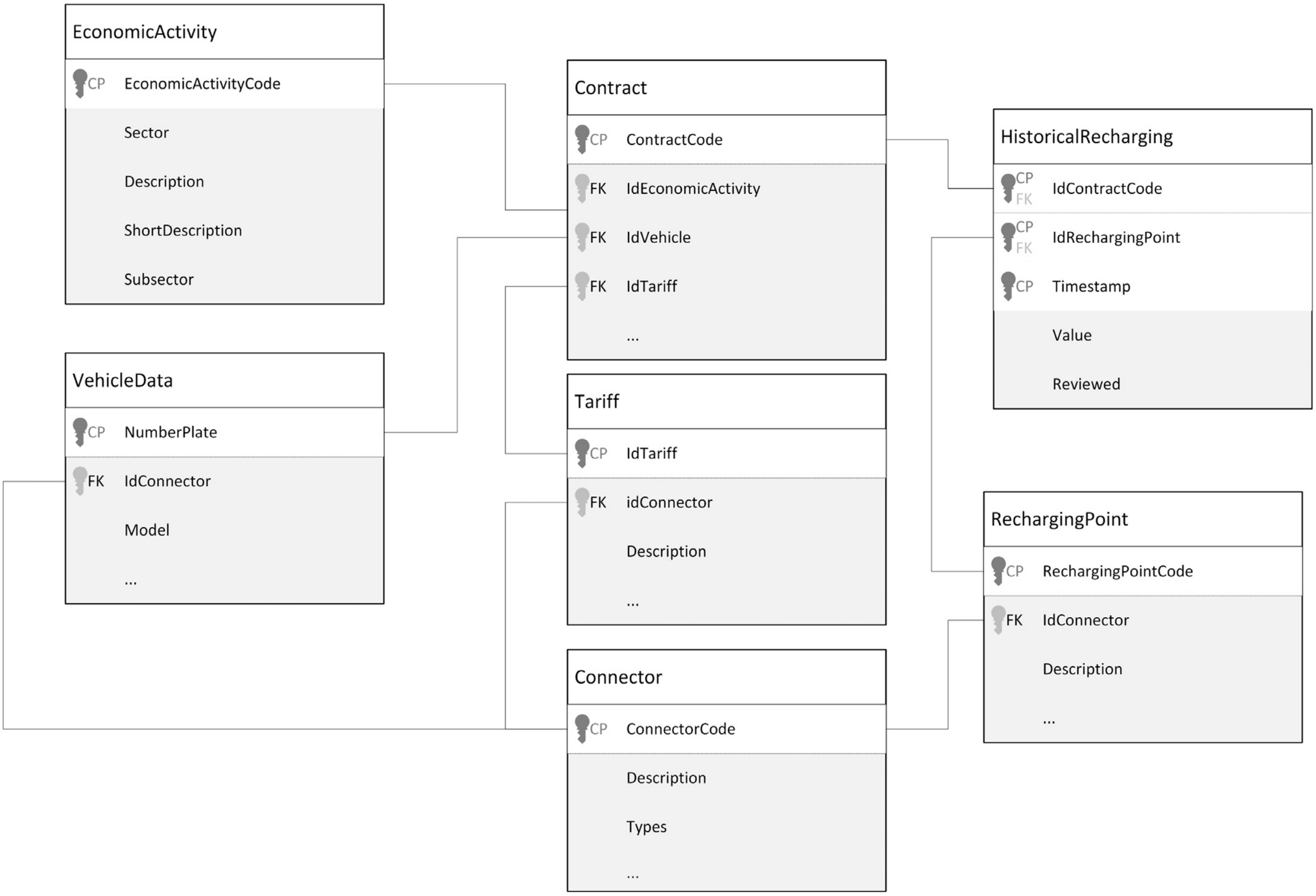

Fig. 4. UML diagram for Source B.

The utility companies have a lot of databases related to different aspects of business, and maybe volumes of several thousands of tables. However, it is very difficult to reach a number of several thousands of tables. Thus, to test the proposed solution a special case is selected. Although this is a real case, it has a low rate of data protection and confidentiality. This case shows the strengths and weaknesses of the proposed solution. The data sources were related to (some columns were omitted because of a confidentiality agreement):

- Source A: Consumer historical information. This data source contains information about consumers: historical consumption data and contract information. This data source has four tables: contract information, historical data, and two parametric tables. The UML diagram is shown in Fig. 3. The foreign keys and interrelations between tables were not established by constraints; the authors indicate the relations in order to make a better presentation of the data source.
- Source B: Recharging stations usage information. This data source contains information about consumption at a recharging station. This data source has six tables: recharging station information, contractual information, vehicle information, consumption information, and three parametric tables. The UML diagram is shown in Fig. 4. The foreign keys and interrelations between the tables were not established by constraints; the authors indicate the relations in order to make a better presentation of the data source.
- Source C: Generation data from different source types. This data source contains information about wind and photovoltaic generation data. This data source has three tables: historical information, source information, and a parametric table. The UML diagram is shown in Fig. 5. The foreign keys and interrelations

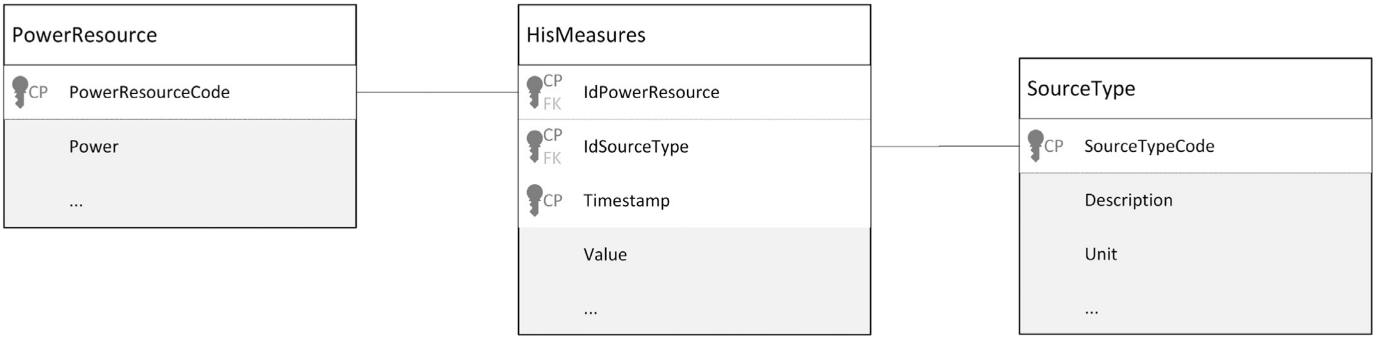

Fig. 5. UML diagram for Source C.

between the tables were not established by constraints; the authors indicate the relationship in order to make a better presentation of the data source.

After the metadata mining process and the characterization stage, the results for each data source are shown in Tables 1 and 2. The information in Tables 1 and 2 is only regarding tables and databases. This information is evaluated by a decision support system. The information about columns has been omitted because of a confidentiality agreement.

In Table 1, the number of records shows what the greatest tables in each source are. All these sources are historical information. The relationship indicator and the column relationship indicator show the interrelation level of the table. If the relationship indicator is near to 1, the structure of the database will be near a star or snowflake structure. The column relationship indicator identifies the number of columns that is needed to define a record. The system combines this indicator with the data type of each column to estimate the maximum size of the table. The self-relationship indicator shows the reflexive relationships. The sources do not have any reflexive relationships.

In Table 2, the coefficients and indicators were calculated according to the results of previous characterization processes. In this case, all the sources showed a high rate of possibilities for application of data mining techniques. They show a high rate of cohesion and low rate of replication. The best punctuation is for time analysis and forecasting. Thus, the decision support system selected the methods related to time analysis and forecasting to apply in the data mining stage.

These data sources were in different RDBMSs: Microsoft SQL Server, MySQL, and Oracle. The integration was performed in an HBase.

Following the IEC Standards, seventeen tables were created: PowerSystemResources, Mesaurement, Terminal, Analog, AnalogValue, AnalogLimitSet, Accumulator, AccumulatorValue, AccumulatorLimit, AccumulatorLimitSet, StringMeasurement, StringMeasurementValue, Discrete, DiscreteValue, ValueAliasSet, ValueToAlias, and MeasurementValueSource. There is no table about quality of measurement because there was no table about quality. Additionally the information about the different characterization process (metadata mining) was added to the database using the tables described in the Integration of Information section.

The data mining modelling was configured to forecasting methods. This configuration is selected by the system based on the nature of the parameters and the indexes calculated in the metadata mining process. This situation can be changed however, by the user adding options for outlier detection, classification, or visualizations. The results of data mining modelling in each parameter are shown in Table 3. In some cases, the system selected several methods because they have the same evaluation value. Nevertheless the different methods were ordered according to the time required for the model generation process.

A regression model was created for the Recharging stations, but in the test stage the generated model showed a very high error rate. The algorithm has no information about routes or drivers.

7.1. Performance test

The proposed solution was designed to work in an architecture based on Hadoop or Spark architecture, interacting with Hbase. However, it was not deployed in a real cluster of machines. The cluster was implemented with two virtualized servers. The first server has an Intel i7 (3 GHz), 16 GB RAM, GTX750 (2GB and 640 CUDA cores) and 8 TB of hard disk space. The second server has an Intel Xeon E5 (2 GHz), 64 GB RAM, Quadro K1200 (4 GB and 512 CUDA cores) and 10 TB of hard disk space. The proposed solution takes advantage of other solutions developed for other projects (Guerrero et al., 2016a) in order to integrate the architectures of Hadoop and Spark and to take advantage from CUDA cores in some operations.

The performance study is based on the application of metadata mining and data mining processes. The extraction process is performed but is not included in the performance study. The extraction process is executed in systems with a high control in data access. These systems have window times in which it is possible to execute extraction and backup processes. This window times are sometimes in night hours or, even weekends. Each system has its own window times. For these reasons, the performance study omitted the extraction processes. The metadata mining process is executed in Hadoop and Spark architectures, storing the results in Hbase. The data mining process is performed by an SPSS Modeler connected to the HBase server. Of course, this study is limited by the proposed hardware, in better and greater architectures the process will be faster.

The Table 4 shows the results for the proposed case, showing the size of each database, and the time investing in metadata mining. The integration and data mining process took 2.14 h.

The framework was applied on other sets of data sources in order to compare the evaluation test with bigger data sources. The results of this new data sources are shown in Table 5. The integration and data mining processes took 50.5 h.

The results of the performance test provide some interesting conclusions:

- The time of metadata mining process depends on the number of columns (Fig. 6), and the number of fields that contains dates, timestamps or long texts. The increase of these types of columns could increase the metadata mining process complexity in time. The influence of Size (Fig. 7) did not show any clear relation. However, the influence of number of tables (Fig. 8) shows some similarities, but it is not clear in low values.

Table 3
Results of data mining forecasting detected parameters.

Data source	Parameter	Modelling method	Correlation	Error
A	Authorised car dealer*	Linear regression generalized linear model	0.993	0.014
A	Hotel industry*	Regression generalized linear model	0.993	0.014
A	Technical advice office*	Regression generalized linear model	0.992	0.017
A	General services*	Regression generalized linear model	0.996	0.007
A	Communication office*	Regression generalized linear model	0.99	0.019
A	Power generation company office*	Regression generalized linear model	0.955	0.087
A	Authorised car dealer (without garage)*	Regression generalized linear model	0.971	0.058
A	Consulting office	Neural network (multilayer perceptron)	0.961	0.046
A	Main power distribution office*	Regression generalized linear model	0.992	0.015
A	Power distribution office	Neural network (multilayer perceptron)	0.977	0.046
A	Temporary employment agency office*	Regression generalized linear model	0.983	0.033
B	Recharging stations	Not useful model		
C	20 KW generation plant*	Regression generalized linear model	0.993	0.014
C	80 KW generation plant*	Linear regression generalized linear model	0.99	0.019
C	100 KW generation plant*	Linear regression generalized linear model	0.991	0.018

*: Several modelling techniques provides similar correlation and error rates. The different techniques based on regression usually provide the same model or similar.

Table 4
Results of performance test in the proposed case.

Data source	Number of tables	Number of columns	Size (GB)	Metadata mining time (Seconds)
A	4	34	27.46	1.34
B	7	73	0.38	2.34
C	3	19	9.17	0.49

Table 5
Results of performance test in new case.

Data source	Number of columns	Number of columns	Size (GB)	Metadata mining time (Seconds)
D	16	382	192.77	390.02
E	16	364	169.57	371.84
F	16	315	45.29	322.35
G	16	321	40.94	327.41
L	16	306	29.69	313.26
O	1	387	2.97	392.07
P	659	5732	16.46	5622.12
Q	521	2165	0.02	2291.85

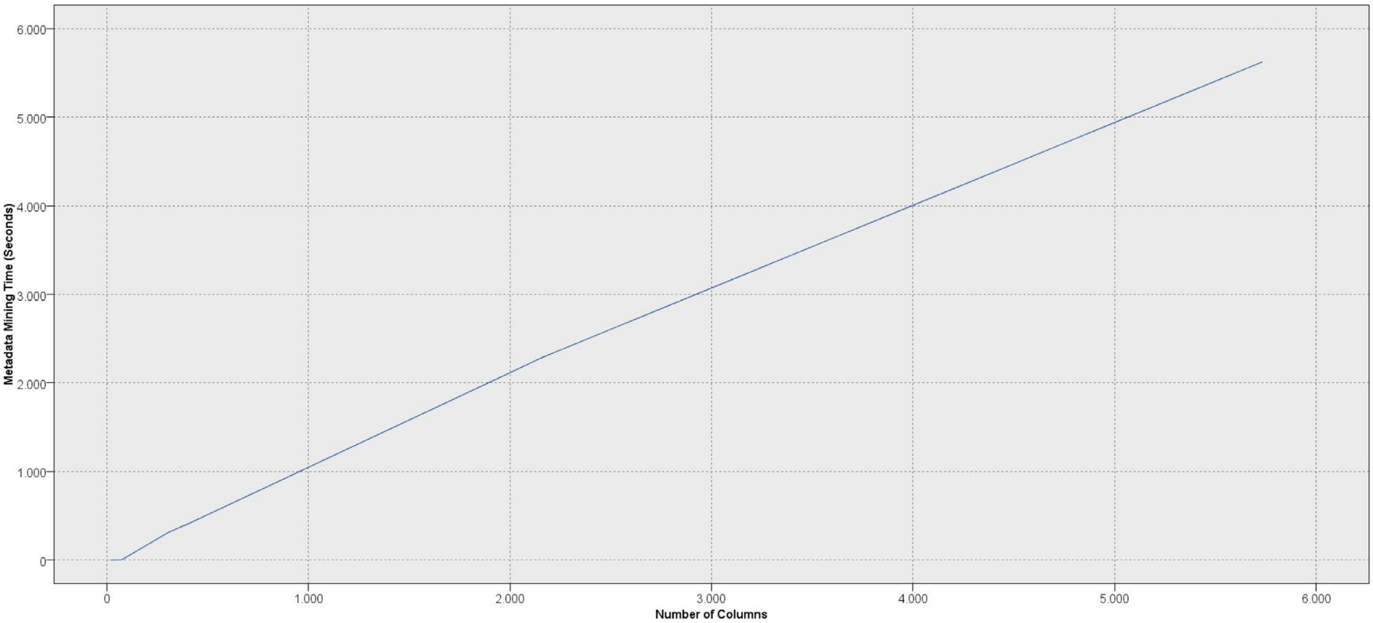

Fig. 6. Number of columns and metadata mining time (Seconds) chart.

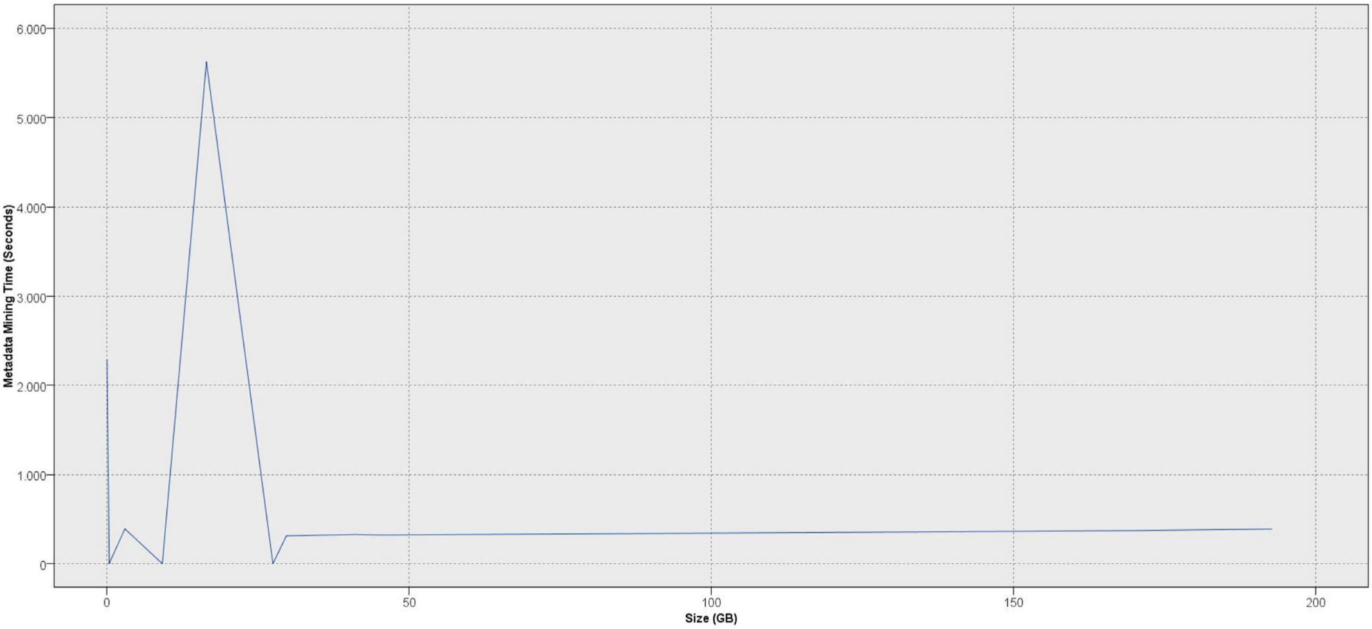

Fig. 7. Size (GB) and metadata mining time (Seconds) chart.

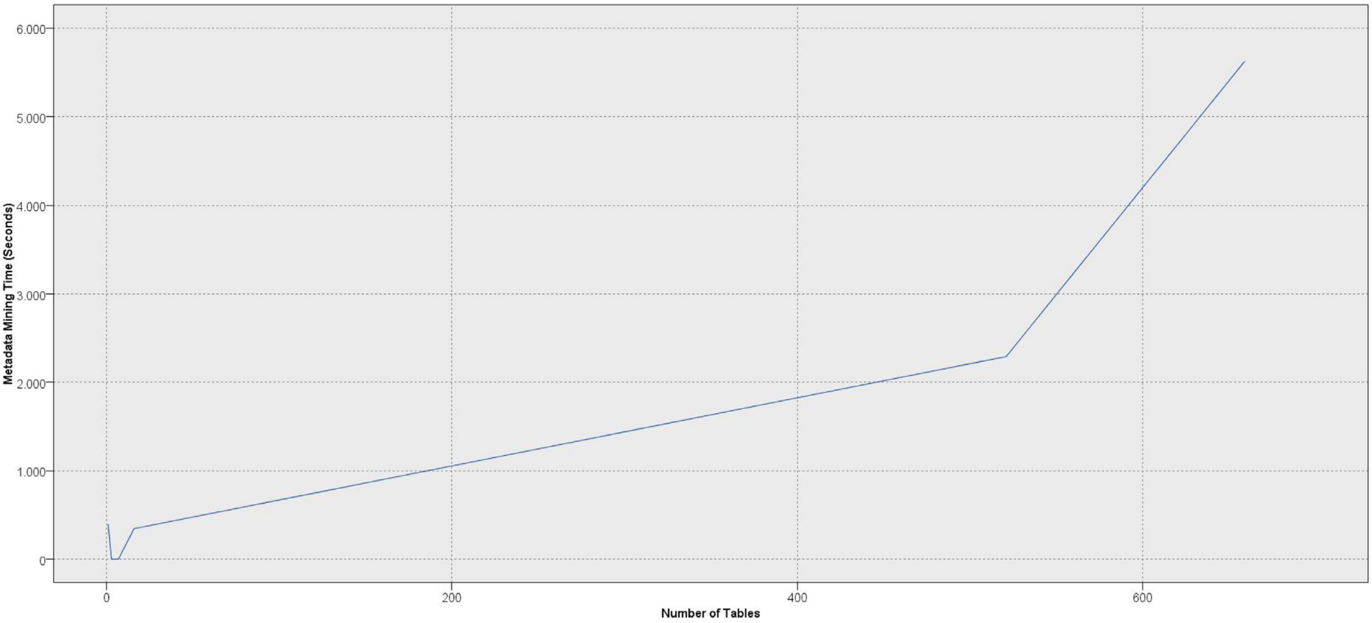

Fig. 8. Number of tables and metadata mining time (Seconds) chart.

- The time of integration depends strongly on the total number of tables and size. If the total number of tables is very high, the number of relationships is very high, too. If the size increases, the time invested to translate the information increases, too. Thus, the ETL Dynamic Engine takes more time to get the final integration.
- The time of data mining process depends strongly on the number of columns involved in the process.

8. Conclusions

Smart grids and the new technologies related to information management are the future of the new smart services and applications. Several services and applications of different technological levels coexist within the current utility grid. In this sense, it is necessary to establish techniques that provide the capability to integrate information from different architecture and technological levels. These technologies increase the robustness of the management systems related to the utility grid.

The metadata mining process is focused on metadata, and taking advantage of this technology it is possible to make systems that integrate the information, according to an information standard, star, or extended-star structure. Additionally, a system for automatic modelling is provided, based on a previous application of a metadata mining algorithm. In this way, this technology provides an easy-to-use and adaptive platform to integrate and model information. The models could be improved by adding new information, and performing the modelling algorithm.

In this paper, the proposed system is used in power distribution, but the future research lines include the application of this technology to other types of database, such as document-based and key-value databases.

9. Future research lines

The future research lines are:

- Test these techniques in other type of utilities.
- Extend these techniques with non-relational databases.
- Extend these techniques to use in health sector.
- Modelling the variation of threshold for automatic data mining techniques according to the results of metadata mining in order to increase the accuracy of generated models.

Additionally, the research team is currently researching about new techniques to integrate heterogeneous systems at web service level (Guerrero et al., 2016b).

Acknowledgments

The authors would like to thank the Smart Business Project (SBP) (Reference Number: P011-13/E24), which provided data sources. Additionally, the authors would like to thank the IDEA Agency for providing the funds for the project.

The authors are also appreciative of the backing of the SIIAM project (Reference Number: TEC2013-40767-R), which is funded by the Ministry of Economy and Competitiveness of Spain.

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th international conference on very large data bases* (pp. 487–499). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from. http://dl.acm.org/citation.cfm?id=645920.672836.

Alemu, G., & Stevens, B. (2015). 8 - The principle of metadata filtering. In *An emergent theory of digital library metadata* (pp. 89–96). Chandos Publishing. Retrieved from. http://www.sciencedirect.com/science/article/pii/B9780081003855000080.

Arnold, A., Liu, Y., & Abe, N. (2007). Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 66–75). New York, NY, USA: ACM. https://doi.org/10.1145/1281192.1281203.

Asonitis, S., Boundas, D., Bokos, G., & Poulos, M. (2009). Semi – automated tool for characterizing news video files, using metadata schemas. In M.-A. Sicilia, & M. D. Lytras (Eds.), *Metadata and semantics* (pp. 167–178). Springer US. Retrieved from. http://0-link.springer.com.fama.us.es/chapter/10.1007/978-0-387-77745-0_16.

Belsley, D. A., Kuh, E., & Welsch, R. E. (2013). *Regression diagnostics: Identifying influential data and sources of collinearity*. Hoboken, N.J: Wiley-Interscience.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). *Time series analysis: Forecasting and control* (4th ed.). Hoboken, N.J: Wiley.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Taylor & Francis.

Campos, J. P., & Silva, M. J. (2000). ActiveXML: Compound documents for integration of heterogeneous data sources. In J. Borbinha, & T. Baker (Eds.), *Research and advanced technology for digital libraries* (pp. 380–384). Berlin Heidelberg: Springer. Retrieved from. http://0-link.springer.com.fama.us.es/chapter/10.1007/3-540-45268-0_45.

Cao, Y., Chen, Y., & Jiang, B. (2007). A study on self-adaptive heterogeneous data integration systems. In L. D. Xu, A. M. Tjoa, & S. S. Chaudhry (Eds.), *Research and practical issues of enterprise information systems II* (pp. 65–74). US: Springer. Retrieved from. http://0-link.springer.com.fama.us.es/chapter/10.1007/978-0-387-75902-9_7.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), 15:1–15:58. https://doi.org/10.1145/1541880.1541882.

Chen, J., Li, W., Lau, A., Cao, J., & Wang, K. (2010). Automated load curve data cleansing in power systems. *IEEE Transactions on Smart Grid, 1*(2), 213–221. https://doi.org/10.1109/TSG.2010.2053052.

Chen, X. d, & Liu, J. z. (2009). Research on heterogeneous data integration in the livestock products traceability system. In international conference on new trends in information and service science, 2009. NISS '09 (pp. 969–972). https://doi.org/10.1109/NISS.2009.94.

Chiu, T., Fang, D., Chen, J., Wang, Y., & Jeris, C. (2001). A robust and scalable clustering algorithm for mixed type attributes in large database environment. In *proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 263–268). New York, NY, USA: ACM. https://doi.org/10.1145/502512.502549.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological), 34*(2), 187–220.

Fan, H., & Gui, H. (2007). Study on heterogeneous data integration issues in web environments. In *international conference on wireless communications, networking and mobile computing, 2007. WiCom 2007* (pp. 3755–3758). https://doi.org/10.1109/WICOM.2007.929.

Fengguang, X., Xie, H., & Liqun, K. (2009). Research and implementation of heterogeneous data integration based on XML. *9th international conference on electronic measurement instruments, 2009. ICEMI '09* (pp. 4-711–4–715) *https://doi.org/10.1109/ICEMI.2009.5274686* .

Fermoso, A. M., Berjón, R., Beato, E., Mateos, M., Sánchez, M. A., García, M. M., et al. (2009). A new proposal for heterogeneous data integration to XML format. Application to the environment of libraries. *Metadata and semantics*, 143–153 Retrieved from http://0-link. springer. com. fama. us. es/chapter/10. 1007/978-0-387-77745-0_14.

Freedman, D. (2005). *Statistical models: Theory and practice*. Cambridge University Press.

Gao, J., & Xiao, J. (2013). Research on heterogeneous data access and integration model based on OGSA-DAI. In *2013 fifth international conference on computational and information sciences (ICCIS)* (pp. 1690–1693). https://doi.org/10.1109/ICCIS.2013.441.

Geiger, B. C., & Kubin, G. (2012). Relative information loss in the PCA. *arXiv:1204.0429 [Cs, Math]*, 562–566. https://doi.org/10.1109/ITW.2012.6404738.

Guerrero, J. I., León de Mora, C., Biscarri Triviño, F., Monedero, I., Biscarri Triviño, J., & Millán, R. (2011). A real application on non-technical losses detection: The MIDAS project. In *The 7th international conference on data mining proceedings* (pp. 77–83). Retrieved from. https://idus.us.es/xmlui/handle/11441/23491.

Guerrero, J. I., Parejo, A., Personal, E., Biscarri, F., Biscarri, J., & Leon, C. (2016a). Intelligent information system as a tool to reach unapproachable goals for inspectors - high-performance data analysis for reduction of non-technical losses on smart grids. In *Presented at the INTELLI 2016, the fifth international conference on intelligent systems and applications* (pp. 83–87). Retrieved from. https://www.thinkmind.org/index.php?view=article&articleid=intelli_2016_4_10_60123.

Guerrero, J. I., Personal, E., Parejo, A., García, A., & León, C. (2016b). Forecasting the needs of users and systems - a new approach to web service mining. In *the fifth international conference on intelligent systems and applications* (pp. 96–99). Barcelona, Spain: IARIA.

Hailing, W., & Yujie, H. (2012). Research on heterogeneous data integration of management information system. In *2012 international conference on computational problem-solving (ICCP)* (pp. 477–480). https://doi.org/10.1109/ICCPS.2012.6384220.

Han, X. b, Tian, F., & Wu, F. b. (2009). Research on heterogeneous data integration in the safety production and management of coal-mining. In *2009 first international workshop on database technology and applications* (pp. 87–90). https://doi.org/10.1109/DBTA.2009.60.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. MacMillan Publishing Company.

Hidber, C. (1999). Online association rule mining. In *Proceedings of the 1999 ACM SIGMOD international conference on management of data* (pp. 145–156). New York, NY, USA: ACM. https://doi.org/10.1145/304182.304195.

Hoiles, W., & Krishnamurthy, V. (2015). Nonparametric demand forecasting and detection of energy aware consumers. *IEEE Transactions on Smart Grid, 6*(2), 695–704. https://doi.org/10.1109/TSG.2014.2376291.

IBM SPSS Modeler 16 Algorithms Guide. (n.d.). IBM Press.

IBM SPSS Modeler 16 Python Scripting and Automation Guide. (n.d.). IBM Press.

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 29*(2), 119–127. https://doi.org/10.2307/2986296.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*(1), 59–69. https://doi.org/10.1007/BF00337288.

La, Q. D., Chan, Y. W. E., & Soong, B. H. (2016). Power management of intelligent buildings facilitated by smart grid: A market approach. *IEEE Transactions on Smart Grid, 7*(3), 1389–1400. https://doi.org/10.1109/TSG.2015.2477852.

Li, Y., Kang, Z., & Gao, H. (2007). Automatic data mining by asynchronous parallel evolutionary algorithms. In L. Kang, Y. Liu, & S. Zeng (Eds.), *Advances in computation and intelligence* (pp. 485–492). Berlin Heidelberg: Springer. Retrieved from. http://0-link.springer.com.fama.us.es/chapter/10.1007/978-3-540-74581-5_53.

Lin, Y. (2009). Study and technological realization about heterogeneous data integration based on XML schema. In *international conference on test and measurement, 2009. ICTM '09: 2* (pp. 394–397). https://doi.org/10.1109/ICTM.2009.5413020.

Liu, H., Liu, Y., Wu, Q., & Ma, S. (2013). A heterogeneous data integration model. In F. Bian, Y. Xie, X. Cui, & Y. Zeng (Eds.), *Geo-informatics in resource management and sustainable ecosystem* (pp. 298–312). Berlin Heidelberg: Springer. Retrieved from. http://0-link.springer.com.fama.us.es/chapter/10.1007/978-3-642-45025-9_31.

Loh, W.-Y., & Shih, Y.-S. (1997). SPLIT selection methods for classification trees. *Statistica Sinica, 7*(4), 815–840.

Lu, B., & Song, W. (2010). Research on heterogeneous data integration for smart grid. In *2010 3rd IEEE international conference on computer science and information technology (ICCSIT): 3* (pp. 52–56). https://doi.org/10.1109/ICCSIT.2010.5564620.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *proceedings of the fifth Berkeley symposium on mathematical statistics and probability*: 1. Statistics, The Regents of the University of California Retrieved from http://projecteuclid.org/euclid.bsmsp/1200512992.

Madsen, H., & Thyregod, P. (2010). *Introduction to general and generalized linear models*. CRC Press.

Merrett, T. H. (2001). Attribute metadata for relational OLAP and data mining. In

G. Ghelli, & G. Grahne (Eds.), *Database programming languages* (pp. 97–118). Berlin Heidelberg: Springer. Retrieved from. http://0-link.springer.com.fama.us.es/chapter/10.1007/3-540-46093-4_6.

Métais, E. (2002). Enhancing information systems management with natural language processing techniques. *Data & Knowledge Engineering, 41*(2–3), 247–272. https://doi.org/10.1016/S0169-023X(02)00043-5.

Pan, J. S., McInnes, F. R., & Jack, M. A. (1996). Fast clustering algorithms for vector quantization. *Pattern Recognition, 29*(3), 511–518. https://doi.org/10.1016/0031-3203(94)00091-3.

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge, U.K.; New York: Cambridge University Press.

Personal, E., Guerrero, J. I., Garcia, A., Peña, M., & Leon, C. (2014). Key performance indicators: A useful tool to assess smart grid goals. *Energy, 76*, 976–988. https://doi.org/10.1016/j.energy.2014.09.015.

Richardson, P., Flynn, D., & Keane, A. (2012). Local versus centralized charging strategies for electric vehicles in low voltage distribution systems. *IEEE Transactions on Smart Grid, 3*(2), 1020–1028. https://doi.org/10.1109/TSG.2012.2185523.

Şah, M., & Wade, V. (2012). Automatic metadata mining from multilingual enterprise content. *Web Semantics: Science, Services and Agents on the World Wide Web, 11*, 41–62. https://doi.org/10.1016/j.websem.2011.11.001.

Shi, Y., Liu, X., Xu, Y., & Ji, Z. (2010). Semantic-based data integration model applied to heterogeneous medical information system. In *2010 The 2nd international conference on computer and automation engineering (ICCAE): 2* (pp. 624–628). https://doi.org/10.1109/ICCAE.2010.5451697.

Sousa, T., Morais, H., Vale, Z., Faria, P., & Soares, J. (2012). Intelligent energy resource management considering vehicle-to-grid: A simulated annealing approach. *IEEE Transactions on Smart Grid, 3*(1), 535–542. https://doi.org/10.1109/TSG.2011.2165303.

Su, J., Fan, R., & Li, X. (2010). Research and design of heterogeneous data integration middleware based on XML. In *2010 IEEE international conference on intelligent computing and intelligent systems (ICIS): 2* (pp. 850–854). https://doi.org/10.1109/ICICISYS.2010.5658689.

Tang, J., Zhang, W., & Xiao, W. (2005). An algebra for capability object interoperability of heterogeneous data integration systems. In Y. Zhang, K. Tanaka, J. X. Yu, S. Wang, & M. Li (Eds.), *Web technologies research and development - APWeb 2005* (pp. 339–350). Berlin Heidelberg: Springer. Retrieved from. http://0-link.springer.com.fama.us.es/chapter/10.1007/978-3-540-31849-1_34.

Tianyuan, L., Meina, S., & Xiaoqi, Z. (2010). Research of massive heterogeneous data integration based on Lucene and XQuery. In *2010 IEEE 2nd symposium on web society (SWS)* (pp. 648–652). https://doi.org/10.1109/SWS.2010.5607370.

Usama, M., & Fayyad, K. B. I. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th international joint conference on artificial intelligence* (pp. 1022–1029).

Wang, L., Wang, Z., & Yang, R. (2012). Intelligent multiagent control system for energy and comfort management in smart and sustainable buildings. *IEEE Transactions on Smart Grid, 3*(2), 605–617. https://doi.org/10.1109/TSG.2011.2178044.

Wong, R. K. (1999). Heterogeneous data integration and presentation in multimedia database management systems. In *IEEE international conference on multimedia computing and systems, 1999: 2* (pp. 666–671). vol.2. https://doi.org/10.1109/MMCS.1999.778563.

Yi, J., & Sundaresan, N. (2000). Metadata based web mining for relevance. In *database engineering and applications symposium, 2000 international* (pp. 113–121). https://doi.org/10.1109/IDEAS.2000.880569.

Yi, J., Sundaresan, N., & Huang, A. (2000). Metadata based web mining for topic-specific information gathering. In K. Bauknecht, S. K. Madria, & G. Pernul (Eds.), *Electronic commerce and web technologies* (pp. 359–368). Berlin Heidelberg: Springer. Retrieved from. http://0-link.springer.com.fama.us.es/chapter/10.1007/3-540-44463-7_31.

Zidan, A., & El-Saadany, E. F. (2012). A cooperative multiagent framework for self-healing mechanisms in distribution systems. *IEEE Transactions on Smart Grid, 3*(3), 1525–1539. https://doi.org/10.1109/TSG.2012.2198247.