UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

The application of artificial intelligence on big data generated by smart cities: A review of the preprocessing phase

Siebe Janssen

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

dr. Gonzalo NAPOLES RUIZ

UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

2019
2020

UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

The application of artificial intelligence on big data generated by smart cities: A review of the preprocessing phase

Siebe Janssen

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

dr. Gonzalo NAPOLES RUIZ

This master thesis was written during the COVID-19 crisis in 2020. This global health crisis might have had an impact on the (writing) process, the research activities and the research results that are at the basis of this thesis.

Abstract

As populations grow in size, new challenges arise in the big metropolitan cities. Resources need to be better distributed, traffic needs to be managed better, pollution needs to be kept at a minimum, etc. In order to facilitate the solutions for these kinds of problems cities are investing in new technologies with the aim to increase their sustainability and to improve the level of comfort for its citizens. This kind of cities are called "Smart Cities".

Through these new information and communication technology solutions, a lot of data is being generated at high speeds every day. We can classify this huge amount of data as big data. Data in smart cities comes from all kinds of sources in all kinds of formats. In most cases the data is not ready to be used for analysis when obtained, it can contain some noise or incorrect values. For cities to be able to analyze the data and to extract knowledge from the data, the data first needs to be preprocessed. This is a crucial step in the analysis process.

In this paper, we first define the concepts of "Smart Cities" and "Big Data". Thereafter, we will focus on the preprocessing steps in the analysis process. We explain what machine learning entails and what its dimensions are. Next, we will list the opportunities and challenges smart cities face and in the last section, we will present some preprocessing methods proposed in the literature to help with the challenges mentioned earlier.

1. Introduction

More than 55% of the population lives in urban areas and this number keeps growing (Camero & Alba, 2019). This introduces new problems into cities such as traffic congestions, pollution, resource scarcity, etc. Cities are investing in new solutions, using for example Internet of Things (IoT), sensors, cameras etc., to solve these problems.

New technologies that are being implemented in cities create huge amounts of data. This data can be classified as Big Data (Allam & Dhunny, 2019). After gathering the data, it needs to be processed and analyzed to gain insides. As mentioned by John Walker (2014) a data-driven decision is a better decision, it helps the managers make choices based on evidence rather than their intuition. The processing of big data can be done by using artificial intelligence and machine learning (Zhou, Pan, Wang, & Vasilakos, 2017). In this paper, we will be meanly focusing on the preprocessing steps of analyzing the data.

Research has been conducted with regards to the amount of literature that exists around the data gathering and data preprocessing phase in smart city analytics. It turns out that there are not a lot of publications at the moment that address the preprocessing phase of big data analytics. This is concerning because it can have a significant impact on the efficiency of the analysis process (Osman, 2019).

In the next section, we discuss the definition of a "Smart city". After that, we will look at big data and its characteristics. In the section that follows, we will be focussing on machine learning and its uses. Then we list some of the challenges, and benefits and opportunities of smart cities and finally we will list some of the latest solutions with regards to the preprocessing of big data generated by smart cities. We finalize the paper with a conclusion.

2. Smart Cities

Through the implementation of new technologies, smart devices and the use of IoT cities are evolving into what is referred to as smart cities. A shared definition of a smart city does not yet exist. Therefore, it is difficult to define a global standard. Through reviewing papers that try to define a definition of a smart city Eiman Al Nuaimi, Hind Al Neyadi, Nader Mohamed, and Jameela Al-Jaroodi (2015) formulate a global definition:

"We can view the smart city as an integrated living solution that links many life aspects such as power, transportation, and buildings in a smart and efficient manner to improve the quality of life for the citizens of such city."

Although smart cities do not have a fixed definition, they do have some characteristics that they all need to meet in order to be called a "Smart city". The main characteristics of a smart city include sustainability, resilience, governance, enhanced quality of life, and intelligent management of natural resources and city facilities (Eiman Al Nuaimi et al., 2015).

Besides defining what smart cities are we can also split them up into different dimensions. Camero and Alba (2019) did a literature review in a quantitative manner by automatically processing a large number of publications regarding smart cities, larger than any human could ever process by himself. Out of this analysis Camero and Alba (2019) found that smart cities can be divided into six dimensions. These dimensions are the following: Smart Economy, Smart Environment, Smart Governance, Smart Living, Smart Mobility and Smart People. Fig. 1. depicts these six dimensions.

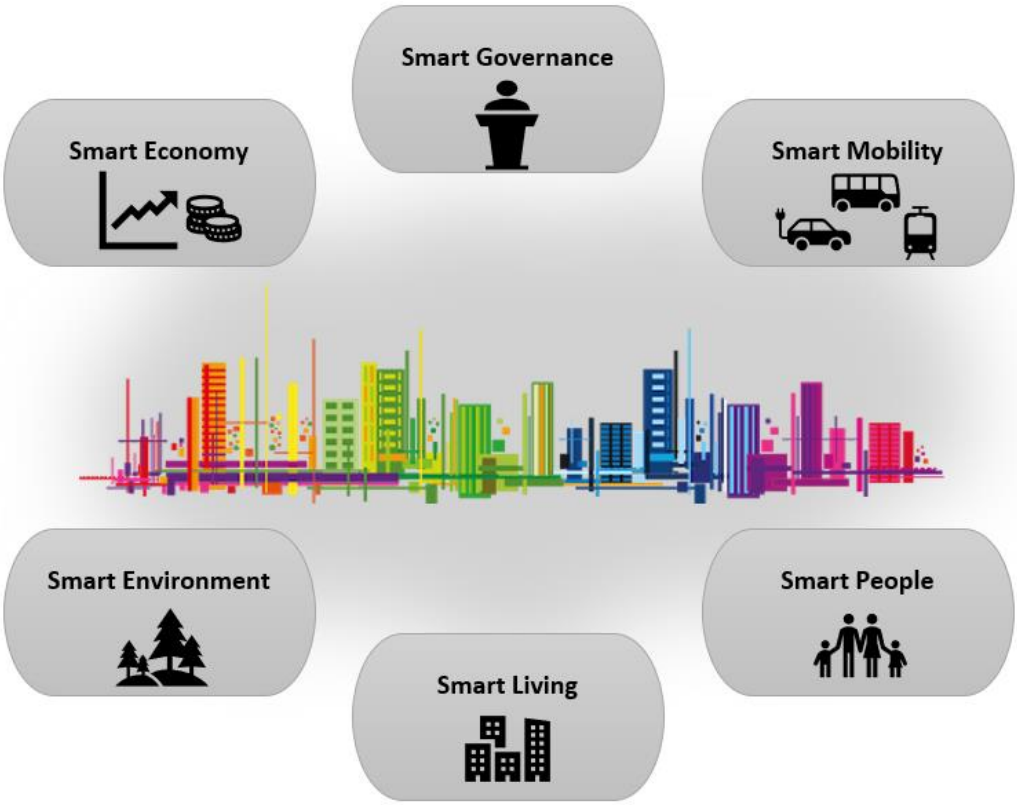

Fig. 1. The six smart city dimensions

Smart economy: a healthy economy helps cities function properly and it enables the cities to offer services to their citizens (Soomro, Bhutta, Khan, & Tahir, 2019). This dimension encompasses, for example, the development of new products, creation of jobs, e-business and e-commerce, etc. (Camero & Alba, 2019).

Smart environment: in this dimension there is a focus on sustainability. Urban cities face many problem with regards to the environment, for example, their carbon footprint, litter, etc. (Soomro et al., 2019). But it's not only the pollution that is included in this dimension, in a city there is also a need for efficient resource allocation. For example, smart energy can be categorized within the smart environment dimension. Here the focus is on better energy allocation and renewable energy (Camero & Alba, 2019).

Smart governance: this dimension focuses on better planning and faster decision making based on real-time facts delivered by IT solutions. Furthermore, there is a focus on trying to improve the democratic processes and public services (Camero & Alba, 2019).

Smart living: this dimension refers to the attempt of cities to increase the living standards in its city and become more attractive for the citizens. This is done, for example, by investing in green energy to increase air quality, better management of traffic to limit the annoyance of noise for its citizens, investing in security so that the citizens feel safer, etc. (Camero & Alba, 2019).

Smart mobility: in this dimension, the focus is on smart transportation and smart logistics. The goal is to get people and goods where they need to be in a quicker and safer manner. Some problems that need to be solved with regards to mobility are congestions, traffic accidents and air pollution (Soomro et al., 2019).

Smart people: this dimension focuses on the improvements of the citizens itself by giving them access to education and training programs, improving their creativity and fostering innovation (Camero & Alba, 2019).

All dimensions mentioned above try to resolve the upcoming problems with the help of technology, IoT and big data generated by smart cities. But Allam and Dhunny (2019) emphasize that we need to be careful not to overlook the sustainability and livability dimensions in favor of the technological ones when it comes to smart cities.

In de current framework for smart cities there is one dimension that is being overlooked and that is the livability dimension. Allam and Newman (2018) support the fact that smart cities need to focus on the technological aspects, but they believe that the overarching dimensions should be focused towards the people and improving the urban livability. With this in mind Allam and Dhunny (2019) propose three key dimensions namely Culture, Metabolism and Governance. Emphasizing too much on technology pose a threat to the livability and sustainability levels of a city. That is why Allam and Newman (2018) want the policy-makers to first focus on these three dimensions prior to the integration of technology.

The proposed framework can be seen in Fig. 2. In this framework, we can see that through IoT we get Big Data that can be processed and analyzed by artificial intelligent solutions. In the core we see the three dimensions: Culture, Governance and Metabolism. These need to make sure that the inclusion of big data and artificial intelligence is geared towards the human livability and not solely towards the implementation of new technologies that benefit the economic situation of corporations.

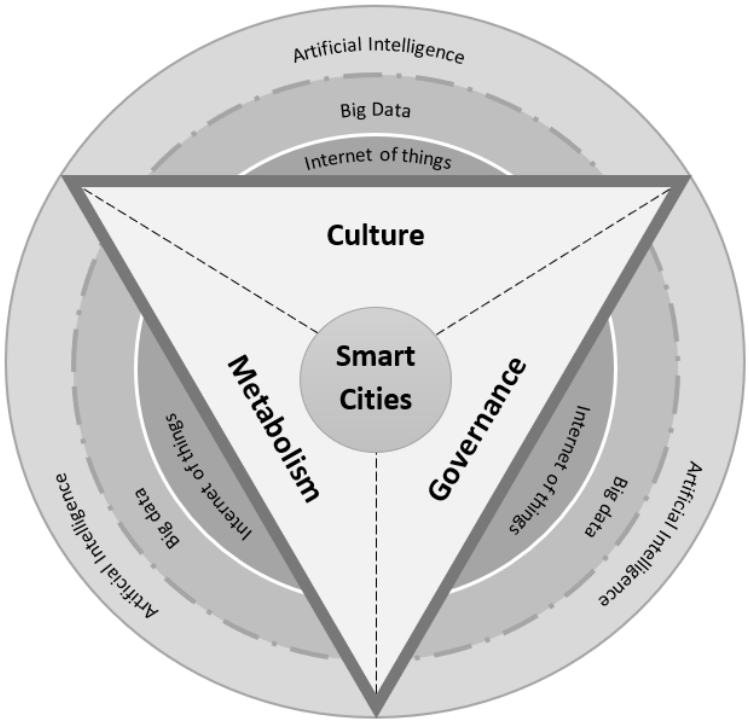

Fig. 2. The framework proposed by Allam and Newman (2018) for the Integration of AI and big data in smart cities to ensure livability

3. Big Data

In the previous section, we explained what a smart city entails. In smart cities there are a lot of sensors, smart devices, cameras, etc. that generate massive amounts of data also called big data.

Just as there is no single definition of a smart city, there is no single definition for big data. In general, it seems to be that big data is about problems so big that the traditional tools and models cannot handle them because they are not adequate or they take too much time (Torrecilla & Romo, 2018). There may be no fixed definition for big data but there can be found some characteristics that define the concept of big data in the literature. In this section, we describe the concept of big data and what its characteristics are.

Going through the literature we find that there are a lot of characteristics defined for big data, but not everybody takes into account all of the characteristics. Table 1. presents an overview of the reviewed papers that mentioned the characteristics of big data. The table illustrates which characteristics are being emphasized in which paper.

Table 1. Big data characteristics in literature

	Volume	Velocity	Variety	Variability	Value	Volatility	Validity	Veracity
(Al Nuaimi, Al Neyadi et al. 2015)	X	X	X	X	X	X	X	X
(Zhou, Pan et al. 2017)	X	X	X		X			X
(Mohammadi and Al-Fuqaha 2018)	X	X	X					
(Osman 2019)	X	X	X					X
(García-Gil, Luengo et al. 2019)	X	X	X		X			X
(Torrecilla and Romo 2018)	X	X	X					
(Ait Hammou, Ait Lahcen et al. 2020)	X	X	X		X			X
(Golov and Rönnbäck 2017)	X	X	X					
(AlNuaimi, Masud et al. 2019)	X	X	X	X				X
(Nguyen, Xue et al. 2020)	X	X	X		X			X
(Malhat, Menshawy et al. 2020)	X	X	X	X	X			X
(Ramírez-Gallego et al., 2016)	X	X	X		X			X
(Ridzuan & Wan Zainon, 2019)	X	X	X		X			X
(S. Ramírez-Gallego, García, Benítez, & Herrera, 2018)	X	X	X		X			X

Table 1. clearly shows that there are three main characteristics also called the main Vs (E. Al Nuaimi, H. Al Neyadi, N. Mohamed, & J. Al-Jaroodi, 2015) and some additional characteristics. Gartner (Laney, 2001) introduced these three main V's with regards to big data. These three V's include volume, velocity and variety.

The meaning of these three main V characteristics are as follows:

- Volume: refers to the size of the data
- Velocity: refers to the speed at which data is being generated
- Variety: refers to the different types of data being generated

Later Gartner (Laney, 2001) added two more V's, namely veracity and value. The meaning of these two V's can be defined as follows:

- Value: refers to the advantages the data can offer to, for example, a business
- Veracity: refers to the accuracy and truthfulness of the data and the meaningfulness of the results generated from the data for certain problems.

The meaning of the other V characteristics mentioned in Table 1. are the following:

- Variability: refers to the changes of structure and meaning of the data
- Volatility: refers to the retention policy of the data
- Validity: refers to the correctness, accuracy and validation of the data

The various characteristics of big data offer a huge potential to help in gaining knowledge and advancing in the different smart city dimensions (E. Al Nuaimi et al., 2015). But in order to be useful, there is a need for the right tool and methods to cope with big data. At the moment there is a lack of these tools and methods.

A study has shown that if data is not immediately used after obtaining it, it is unlikely that it will ever be used in the future (Mohammadi & Al-Fuqaha, 2018). The study also showed that only a small fraction of the available smart city data is being used today. This is caused by a lack of labeled data. This is why there is a need for Machine learning algorithms who make use of a combination between labeled and unlabeled data (Mohammadi & Al-Fuqaha, 2018).

Regarding data there is also a distinction that can be made with regards to static and streaming data. Both of these types of data will be present in smart cities. Sergio Ramírez-Gallego, Krawczyk, García, Woźniak, and Herrera (2017) focus on the main differences between static and streaming data. Some of the characteristics of a streaming scenario are:

- In a streaming scenario the data is not there beforehand, instead it is given piece by piece, in data chunks.
- The time intervals between arrivals of chunks my vary over time and this can go very fast.
- It is impossible to store all the data from a streaming scenario in memory as streams are infinite in size.
- Older instances may get discarded to free up memory space in a streaming scenario. The consequence of this is that instances can only be revisited a limited amount of times.

- In streaming scenarios there is only a limited time to process the instances to offer real-time responsiveness.
- Access to true class labels is limited in streaming scenarios due to the high cost of label queries.
- Access to true labels can also be delayed, they may come available after a long period of time.
- Statistical characteristics of instances can be subject to changes over time in streaming scenarios.

In a static scenario batch processing can be used, in case we have to deal with streaming data we can opt for stream processing. If batch processing is being used, the user must accept some delays on processing the data. If on the other hand, the user opts for a stream processing option, he can handle the data processing real time with minimum delay (Girtelschmid, Steinbauer, Kumar, Fensel, & Kotsis, 2013). The only problem here is selecting the right stream processing framework as there are many on the market offering different advantages and limitations (Isah et al., 2019; Tantalaki, Souravlas, & Roumeliotis, 2019).

In order to process the high volumes that come with big data we need platform scalability. There are two common scaling approaches (Osman, 2019):

- Vertical scaling: upping the computational power of a single operating system (for example, high performance computing clusters).
- Horizontal scaling: dividing the work into batches and processing it in parallel on different operating systems (for example, Apache Hadoop).

Vertical scaling has some drawbacks such as the cost of upscaling and the upper ceiling limitations.

Some of the scalable platforms used to process big data are (Gohar, Muzammal, & Ur Rahman, 2018):

- Apache Hadoop: this is a software that allows for distributed storage and data processing using MapReduce (Singh & Reddy, 2014).
- Apache Spark: this is a cluster-computer framework that is gaining popularity and replacing Apache Hadoop for certain tasks. One of the reasons for this is that Spark offers a programming interface for languages such as Python and R, which are commonly used in machine learning and statistics (Singh & Reddy, 2014).

In the next section we will discuss what machine learning entails and how it can be applied in a preprocess context with regards to big data.

4. Machine Learning

Artificial intelligence is not new. It already existed in the 1950's but there were not a lot of applications for it. It seems however that with the rise of big data in 2011 artificial intelligence began to rise again in popularity (Fig. 3.). From 2013 onwards the popularity of artificial intelligence rose. This trend suggests a correlation between the popularity of big data and artificial intelligence (Allam & Dhunny, 2019).

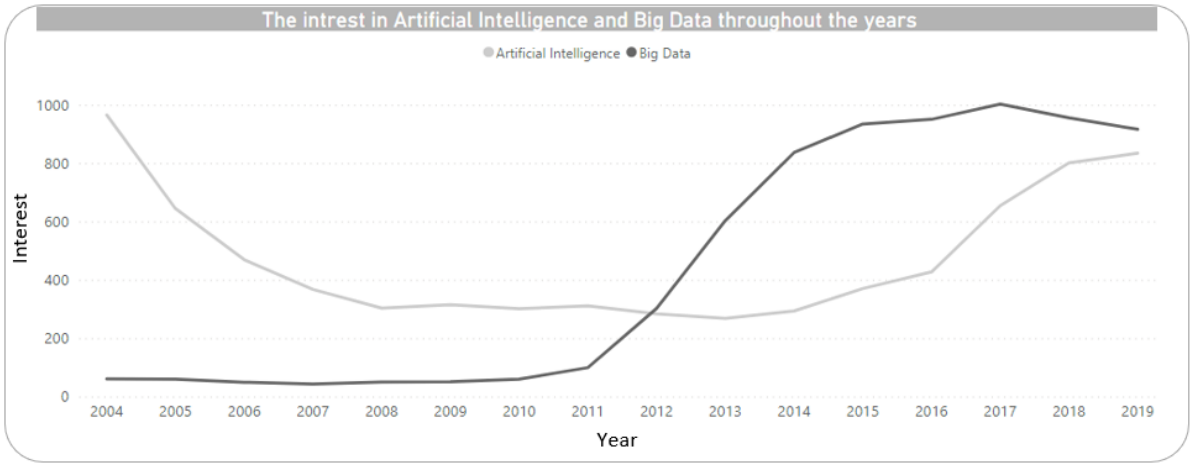

Fig. 3. The popularity of big data and artificial intelligence from 2004 to 2019 (Trends, 2020)

Analyzing big data can be done by using artificial intelligence and machine learning. But this brings some challenges with it as well. Machine learners are not designed to cope with these large amounts of data. Therefore, they need to be modified to be able to handle the huge amounts of data generated by smart cities. A machine learner is a computer system that improves itself through learning from past experience (Jordan & Mitchell, 2015). As we can see in Fig. 4., machine learning can be split up into multiple dimensions.

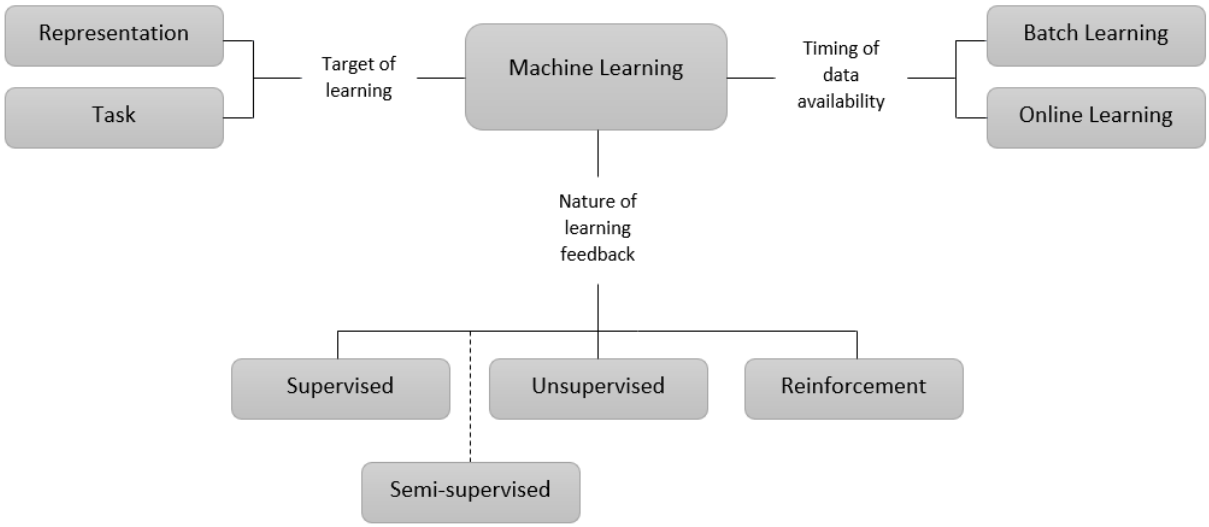

Fig. 4. The dimensions of machine learning

There are three different aspects we need to take into account when talking about Machine Learning: the timing of data availability, the target of learning and the nature of learning feedback (Zhou et al., 2017).

- Timing: batch learning means that the model learns from all the data in one batch whereas online learning updates the model based on each new input (Dekel, 2009).
- Target: the difference between Task and Representation learning is that in the first case the goal is to learn a task based on some inputs while in the second case its more about learning the features themselves (Bengio, Courville, & Vincent, 2013).
- Feedback: here we can distinguish three categories namely supervised, unsupervised and reinforcement learning. In supervised learning the learner will be presented with input-output pairs. In unsupervised learning the system searches for patterns in the input. Reinforcement learning also does not get the input-output pairs, instead it gets feedback on its previous experience. There is a fourth category and this is semi-supervised learning, it falls between supervised and unsupervised learning (Zhang, Harman, Ma, & Liu, 2020).

There are different phases in Machine Learning but for the purpose of this paper we will only focus on the preprocessing part. In the preprocessing phase, the raw data is being transformed into usable data for the next stages. In this phase we can come across all kinds of issues. These issues include data redundancy, inconsistency, noise, heterogeneity, transformation, labeling, data imbalance and feature representation/selection (Zhou et al., 2017). The explanation for these issues is given in Table 2.

Table 2. Explanation of the issues we can come across in the preprocessing phase

Issue	Explanation
Data redundancy	This can be, for example, duplicate data. The impact of redundant data on the results of the analysis can be severe (Zhou et al., 2017).
Data noise	This can entail missing or incorrect values but also data sparsity and outliers (Zhou et al., 2017).
Data heterogeneity	Data received from different sources and in different formats makes it highly heterogeneous, which makes it difficult to apply data mining methods (Jung, 2011).
Data discretization	Discretization is done, for example, by ranking numerical values and splitting them up into different class intervals. This is needed because some Machine Learning algorithms can only cope with discrete values (Zhou et al., 2017).
Data labeling	Most traditional data annotation methods are labor-intensive.
Imbalanced data	This can be addressed by traditional stratified random sampling methods but this can be time-consuming (Zhou et al., 2017).
Feature representation and selection	Machine Learning performance heavily relies upon the choice of data representation or features. Also feature selection improves the performance of a Machine Learning algorithm (Zhou et al., 2017).

So, in order for machine learning algorithms to analyze the data delivered by smart cities these issues need to be addressed. In a later section we will discuss some methods to help solve the issues.

After preprocessing the data, we can use deep learning models to get insight out of the data. Some of the models that are already being used in smart city applications include (Soomro et al., 2019):

- Deep Neural Networks (DNNs): these are used in, for example, social services diagnosis to automatically generate an alert if certain undesired social conditions emerge such as social exclusion (Serrano & Bajo, 2019). Another application domain for Deep Neural Networks is natural ventilation in smart buildings. Here high-fidelity models help to predict the thermal responses of buildings with regards to different environmental conditions (Chen, Tong, Zheng, Samuelson, & Norford, 2020).
- Convolutional Neural Networks (CNNs): this technique is used in, for example, 3D object detection for autonomous driving purposes in order to improve the sustainability of smart cities (L. Wang et al., 2020). Another example where they made use of this technique is in social media sentiment analysis which can be useful in many smart city applications (Alam, Abid, Guangpei, & Yunrong, 2020).
- Recurrent Neural Networks (RNNs): some examples of where this technique is being applied are in the prediction of electricity consumption for commercial and residential buildings (Rahman, Srikumar, & Smith, 2018), short-term load forecasting for energy consumption (J. Kim, Moon, Hwang, & Kang, 2019) and real-time pricing for smart grids (He, Huang, Li, Che, & Dong, 2015).
- Long Short Term Memory (LSTM): this is used in, for example, the prediction of wind speeds to ensure the reliable operation of power systems that relay on wind power (Pei et al., 2019). Another example where this technique is being utilized is in the streamflow and rainfall forecasting which is important in smart cities for water resources planning and management (Ni et al., 2020).
- Autoencoders (AEs): examples of the application of autoencoders are indoor air quality monitoring for subway systems as a key component of the ventilation systems (Loy-Benitez, Li, Nam, & Yoo, 2020) and energy disaggregation for smart homes to ensure energy sustainability by providing meaningful feedback to home owners (F. C. C. Garcia, Creayla, & Macabebe, 2017).
- Generative Adversarial Networks (GANs): these techniques are being used in, for example, the segmentation of images via an image augmentation approach for medical imaging domains because of the limited amounts of annotated samples (Pandey, Singh, & Tian, 2020). Another example is the generation of demand side load data for energy consumption based on historical data in order to help the energy providers to make informed decisions (Lan, Guo, & Sun, 2018).
- Deep Belief Networks (DBNs): examples of the application of DBNs are the reconstruction of images (Sabar, Turky, Song, & Sattar, 2019) and short-term power load forecasting (Fan, Ding, Zheng, Xiao, & Ai, 2020) which can be applied on the power grids of smart cities.

5. Smart City Challenges

In this section, we will address some of the challenges with regards to a smart city defined in the literature. With the rise of smart cities and big data, some new challenges emerge that need to be addressed in order to benefit from the generated data and the new technologies. In Table 3. we list the different challenges mentioned in the reviewed articles:

Table 3. A list of the smart city challenges

Challenge	Explanation
Managing data quality	Bad quality of the data will result in low trustworthiness (Lim, Kim, & Maglio, 2018).
Integrating different data	By connecting data we can produce higher levels of knowledge (Lim et al., 2018).
Addressing security and privacy issues	Many people are concerned with their privacy, so it is very important to find ways to use the data without violation the privacy norms (Lim et al., 2018). Also security is something to consider, there can be for example some falls data injections that cause validity issues (Mohammadi & Al-Fuqaha, 2018).
Understanding the needs of citizens, visitors and employees	Determining which information is useful to citizens, visitors and employees in order to deliver better benefits for them (Lim et al., 2018).
Enhancing geographic information delivery methods	It is important to make visualizations of the information available so it is easily accessible (Lim et al., 2018).
Integrating big and fast data analytics	Smart cities generate a lot of data and some of this data needs to be processed as fast as possible in order to make real-time decisions based on that data (Mohammadi & Al-Fuqaha, 2018).
On-device intelligence	There is a need for light-weight learning algorithms on resource-constraint devices (Mohammadi & Al-Fuqaha, 2018).
Big data shortage	Smart city applications need datasets in order to be developed and evaluated but these are not always available (Mohammadi & Al-Fuqaha, 2018).
Context-awareness	This can help get more value from data by adding contextual information (Mohammadi & Al-Fuqaha, 2018).
Data sources and characteristics	In smart cities there are many different sources and formats of data. Most current methodologies or data mining software tools cannot handle the size and complexity of the data (Eiman Al Nuaimi et al., 2015).
Cost	This focusses mainly on the cost to implement new technologies and the research costs to optimize the new systems implemented in smart cities (Eiman Al Nuaimi et al., 2015).
Smart city population	More people equals more data. The applications need to be able to handle the growing volumes of data (Eiman Al Nuaimi et al., 2015).
Designing smart city services	This integrates all the outcomes from analytics, ideation and information content design for a smart city (Lim et al., 2018).

Besides these challenges there are some considerations we need to keep in mind with regards to smart cities. Here follows a list of some considerations Lim et al. (2018) finds important:

- Adhere to a service-oriented perspective in collecting and analyzing urban data.
- Pay attention to the experience of people in collecting and using data and delivering information to them.
- Employ a data-oriented perspective in designing services.
- Create synergies and minimize conflicts between data-related stakeholders.
- Form a cross-functional team for big data use.

6. Smart City Benefits and Opportunities

After mentioning the challenges smart cities face, we now go over some of the benefits and opportunities that smart cities can deliver. These benefits and opportunities include:

- Efficient resource utilization: reducing the ecological footprint and controlling cost by better management of natural resources (Eiman Al Nuaimi et al., 2015) (Soomro et al., 2019).
- Better quality of life: delivering better services, efficient work and living models, and less waste (Eiman Al Nuaimi et al., 2015).
- Higher levels of transparency and openness: investing in more open datasets so that everybody can use them and enrich them with their own data (Eiman Al Nuaimi et al., 2015).
- Sustainable economic growth: Creating an economic growth rate that can be maintained without causing other significant economic problems for future generations (Soomro et al., 2019).
- Prediction of future scenarios for sustainable cities: predicting future scenarios with the application of machine learning and big data (Soomro et al., 2019).

7. Preprocessing techniques for big data

In this section, we will be focusing on some of the preprocessing techniques offered in literature with regards to big data generated by smart cities. Because data acquisition is loosely governed it can lead to bad quality of data, for example, impossible values, missing values, noise etc. (García, Luengo, & Herrera, 2015). Without addressing these issues, the raw data that is being mined can cause misleading results. This is why preprocessing raw data is a fundamental step to have a fast, precise and valid learning process.

Data preprocessing can take up more than 50% of the total effort of a knowledge discovery process (Sergio Ramírez-Gallego et al., 2017). It is considered as a mandatory step and there exist a lot of preprocessing methods which we can classify into two categories namely data preparation and data reduction (García et al., 2015). The data preparation category includes methods of integration (Do & Rahm, 2007), data cleaning (W. Kim, Choi, Hong, Kim, & Lee, 2003) and transformation (Hashem et al., 2015), and normalization (Hashem et al., 2015). The data reduction category encompasses discretization (Liu, Hussain, Tan, & Dash, 2002), feature selection (Liu & Motoda, 2012) and instance selection (Liu & Motoda, 2013). Fig. 5. gives an overview of these data preprocessing methods.

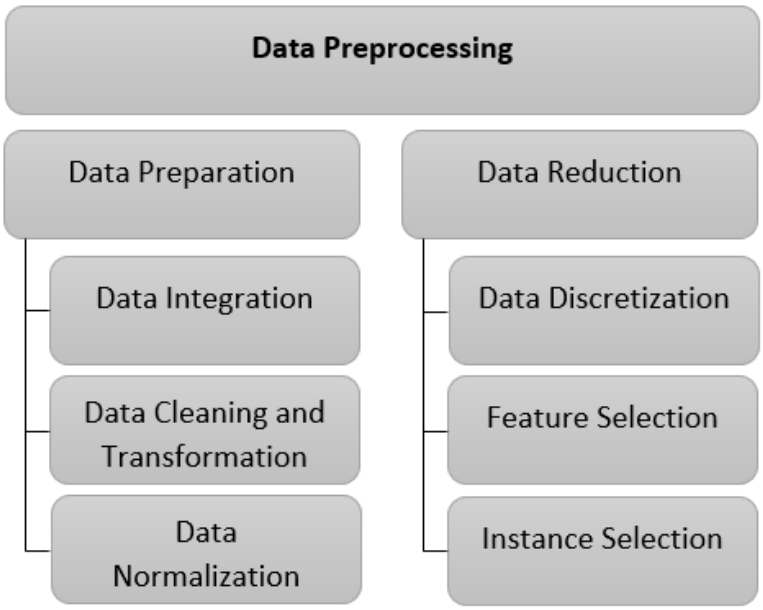

Fig. 5. The different data preprocessing methods

After executing the entire chain of preprocessing methods the data is excepted to be correct and ready to be analyzed by the appropriate algorithms (Silva, Souza, & Motta, 2016). In the first subsection, we will go over the different data preparation techniques. After this, the second subsection will address the data reduction techniques.

7.1 Data preparation

Data preparation refers to the steps that are performed in order for the data to be in the right format to serve as an input for data mining algorithms. In this step prior useless data is converted into data that fits the needs of the data mining algorithm. Without these data preparation steps the data mining algorithm would most likely return errors or even falls results (García et al., 2015).

7.1.1 Data integration

It's hard in a data mining context to collect a single dataset with information that comes from a variety of sources. In a data integration process, this data from different sources are combined in a single unified view. However, redundancies and inconstancies can be introduced into the dataset if the integration process for these different sources is not performed properly. This can result in a decrease of the accuracy and speed of the data mining algorithm (García et al., 2015).

In literature some automatic integration approaches can be found. These include, for example, techniques that match and find the schemas of the data (Doan, Domingos, & Halevy, 2003; Doan, Domingos, & Halevy, 2001) or automatic procedures that reunite the different schemas (Do & Rahm, 2007).

In a data integration context, we can distinguish two tasks namely finding redundant attributes and detecting tuple duplication and inconsistency (García et al., 2015).

The problem of redundant attributes should be avoided at all times. It increases the dataset size, incrementing the processing time of data mining algorithms. Besides this problem, it can cause the resulting model to be overfitted. The best way to counter redundancies is by using correlation analysis (García et al., 2015).

When tuples are obtained it is important to check whether these are not duplicate tuples. Duplicate tuples cause a waste of space, longer processing times of data mining algorithms and are a source for inconsistencies (García et al., 2015). Some techniques to counter this problem are probabilistic approaches (Fellegi & Sunter, 1969), supervised approaches (Cochinwala, Kurien, Lalk, & Shasha, 2001; Joachims, 1999), distance-based techniques (Monge & Elkan, 1996) and clustering algorithms (Verykios, Elmagarmid, & Houstis, 2000).

7.1.2 Data cleaning and transformation

Data cleaning is a necessary step in order to remove or repair dirty data from a dataset. Dirty data can be described as missing values, wrong data and non-standard representations of the same data. Dirty data is a consequence of data update errors, data transmission errors or bugs in data processing systems. If a big part of the dataset is dirty, the application of data mining algorithms will result in an unreliable model (García et al., 2015) this can, for example, affect business decisions (Ridzuan & Wan Zainon, 2019). According to a study (L. Li, 2012), 75% of the 599 companies surveyed have suffered losses that can be associated with data quality issues.

Data cleaning tries to offer a better data quality. Data quality can be described as the fitness of data to fulfill certain business requirements (Ridzuan & Wan Zainon, 2019). In order to obtain better quality data, data cleaning operations are performed on the existing data with the intentions to

remove anomalies and collect data that is an accurate and unique representation of the mini world (Müller & Freytag, 2005). The data cleaning process is complex and includes stages such as the specification of quality rules, the detection of data errors and the reparation of these errors (Khayyat et al., 2015).

Ridzuan and Wan Zainon (2019) divides the data cleaning process into five phases (Fig. 6.):

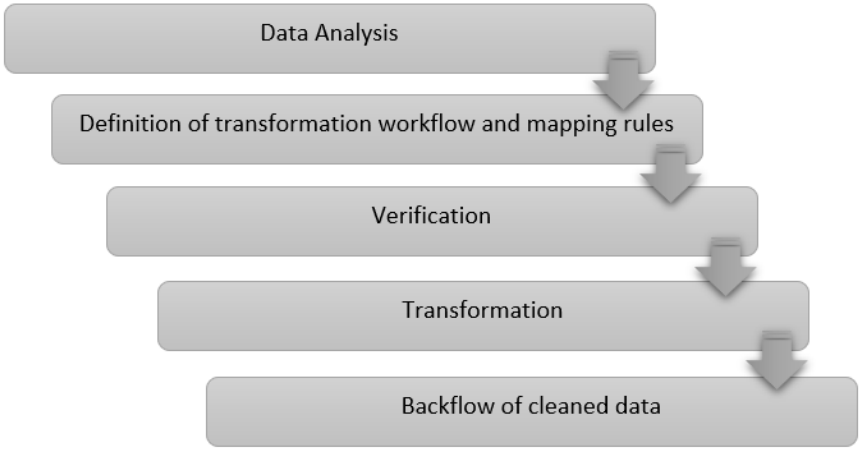

Fig. 6. The data cleaning process

1. The data analysis: here the errors and inconsistencies in the data are identified.
2. The definition of transformation workflow and mapping rules: in this phase the detection and elimination of anomalies executed by a sequence of operations is defined.
3. The verification: here the correctness and effectives of the second phase is evaluated.
4. The transformation: in this phase the data in the data warehouse is being refreshed.
5. The backflow of cleaned data: here the dirty data is being replaced with the cleaned data.

There are some problems regarding traditional data cleaning methods, they are not easy to adapt to a large dataset. Big data usually has, as mentioned before, five characteristics namely value, volume, variety, velocity and veracity (Ridzuan & Wan Zainon, 2019). Out of these five characteristics, volume and variety form the biggest problem with regards to data cleaning. With regards to the volume of big data, the current tools available for data cleaning are not scalable. Data variety is probably the biggest obstacle when trying to effectively use large volumes of data for analysis (Walunj Swapnil, Yadav Anil, & Gupta, 2016). Furthermore, the current data cleaning approaches do not guaranty the accuracy of the repaired data. There is still a need for domain experts that understand and implement the quality rules as well as verify the cleaned data. However these domain experts are expensive and limited (Chu et al., 2015). The last challenge mentioned by (Ridzuan & Wan Zainon, 2019) is the fact that in a big data environment, data is being generated constantly. In order to analyze and use this data in a machine learning context, the data needs to be cleaned first. These cleaning methods need to be able to handle the continuous influx of data without affecting the already existing data.

There are already some proposed solutions for the above mentioned challenges, for example, Cleanix (H. Wang et al., 2014), SCARE (Yakout, Berti-Équille, & Elmagarmid, 2013), KATARA (Chu et al.,

2015) and BigDansing (Khayyat et al., 2015). Except for KATARA that uses a sequential execution method, all of these methods make use of a parallel execution method.

There are two types of dirty data namely missing values and noise. The way these two get handled differs. For missing values, we can ignore them or fill them in manually or with a constant. This can be done, for example, by data transformation, the process of creating new attributes by using different mathematical formulas. For the identification of noise, we can use basic statistical and descriptive techniques such as scatter plots that reveal the outliers. However, in literature, it is recommended to perform a noisy detection and treatment approach usually by filtering (García et al., 2015).

García-Gil, Luengo, García, and Herrera (2019) introduce a new way of filtering noise out of a dataset called the Homogeneous ensemble technique (HME-DB). Because it is widely accepted that we are shifting towards Big Data the proposed method needs be scalable. In order to design the noise filtering process García-Gil et al. (2019) mainly focus on two characteristics of big data namely data veracity and data value as these two are related with data quality. Through there process they try to turn big data into smart data. In order for big data to be smart it needs to meet three criteria. The data needs to be accurate, actionable and agile.

To actually turn the big data into smart data the authors propose three different preprocessing techniques for noise filtering: Homogeneous Ensemble (HME-BD), Heterogeneous Ensemble (THE-BD) and Edited Nearest Neighbor algorithm (ENN-BD). All of which are implemented in a big data framework called Spark (García-Gil et al., 2019).

These three techniques where all tested and in the end the HME-BD technique was far superior to the other two techniques. HME-BD had faster run times, more correctly removed noisy instances and the highest accuracy improvement (García-Gil et al., 2019). Fig. 7. depicts the flowchart of the HME-BD noise filtering algorithm.

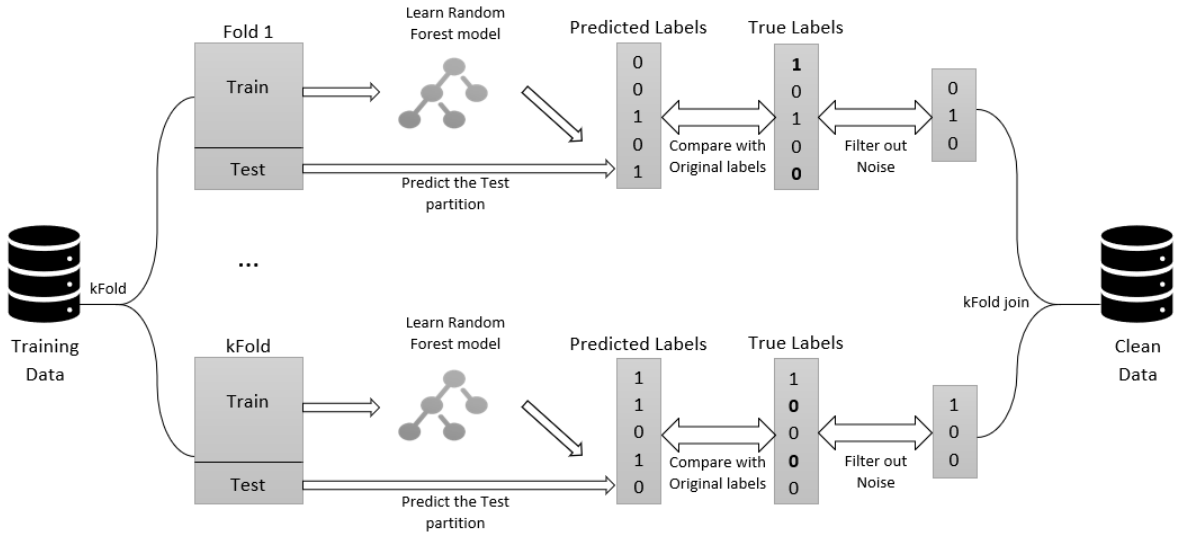

Fig. 7. The flowchart of the HME-BD noise filtering algorithm

First, a kFold partitioning of the dataset is being performed. After this, the algorithm goes through each partition learning a random forest model from the training set. Then the test set is being predicted by the learned model. If the predicted class is different from the original the instance is marked as noise. Lastly, the noisy instances are being removed via a filter and the cleaned dataset is being returned.

This noise filtering technique can handle problems with a large amount of data in a very short timeframe. Some future research that is interesting for this problem can be the classification of multiclass or imbalanced problems with cost-sensitive filters. Another research topic could be renaming or relabeling noisy instances instead of removing them (García-Gil et al., 2019).

7.1.3 Normalization

There is more than one definition for data normalization depending on the study domain. In the domain of databases, we see data normalization as the process where attributes are organized in such a manner that it increases the cohesion and efficiency of managing the data. In statistics however, there is a different view on normalization. Here normalization means the adjustment of the values that are measured on different scales to a common scale, mostly between zero and one (Vafaei, Ribeiro, & Camarinha-Matos, 2016). This makes it easier to compare the values.

First, we will look at the solutions for de database domain. Big data analytics has become a hot topic over the last few years. Business users such as banks and big web companies such as Google or Facebook are increasingly trying to monetize big data. Yet there remains an issue with the implementation of platforms for big data that are fast enough to load, store and execute analytical queries (Oussous, Benjelloun, Ait Lahcen, & Belfkih, 2018).

Hadoop has been considered a good solution for the problems that emerge from the use of big data, because of its almost unlimited horizontal scaling. The data lake approach has been developed as an ideal method to store data. But it does not offer what is expected of it. Instead of delivering a schema-less structure it delivers a schema-on-read structure whereas a traditional database delivers a schema-on-write structure. So instead of structuring the data "on write" the problem is just moved forward because in a data lake setup the data still needs to be structured "on read". Another drawback of the data lake method is the likelihood that it becomes a data swamp that is unstructured, ungoverned and out of control where it is hard to find and use the data (Golov & Rönnbäck, 2017).

The business intelligence team of Avito, a Russian e-commerce site, were challenged with the task to develop a data warehouse that was scalable, could cope with the changing business model and was able to support analytical workloads. Here the developers opted to use the Anchor model (Rönnbäck, Regardt, Bergholtz, Johannesson, & Wohed, 2010) that uses a modeling technique where tables are in the sixth normal form.

In the Anchor modeling technique there are four main constructions that are being used (Golov & Rönnbäck, 2017):

- Anchor: this is a table that holds the surrogate keys for all the instances of an ensemble.
- Attribute: this is a table that contains all the property values of an ensemble.
- Tie: in this table the relationship is being stored between the ensembles.

- Knot: this table holds a set of enumerated values.

Table 4. lists the pros and cons of choosing a high level of normalization for big data.

Table 4. A list of the pros and cons of choosing a high level of normalization for big data

Pros	Cons
It is very easy to expand the Anchor model.	Because the Anchor model uses a single date, time series merge joins are required for loading new date.
Only actual values will be sorted in the Anchor model, "nulls" will be represented by absented rows.	Multiple projection support is required when tables are distributed across multiple servers.
Anchor models only use one date, the "from date". Single date approaches require only inserts, which is better suited for a big data environment.	Because the Anchor model uses highly normalized tables, there is a need for efficient merge joins in order to be able to perform analysis on the data.
The Anchor model contains uniform rules of data distribution which makes it a lot easier to expand the data model.	Even efficient merge join operations inside database management systems cannot guarantee optimal query execution. Therefore, there is a need for efficient query execution plans for ad-hoc analytical queries
It is easier to maintain a big number of comparable normalized tables than a few enormous denormalized tables.	

Although the Anchor model is a very good data warehousing option with regards to efficiency, it is not very well suited for reporting tasks because of its highly normalized structure. Therefore there is a need for an extra layer, a presentation layer, where data marts are implemented. This way the denormalized tables can be used to do analysis and use reporting tools (Golov & Rönnbäck, 2017).

After looking at the domain of databases we can also take a look at data normalization in the domain of statistics. Vafaei et al. (2016) analyzes the effect of five of the most promising normalization techniques on the Analytic Hierarchy Process (AHP) method, a Multi-Criteria Decision Making (MCDM) method. The AHP method is made up of five steps (Vafaei et al., 2016):

1. First the problem is decomposed into a hierarchical structure.
2. Then a pairwise comparison is employs.
3. After this the logical consistency is determined.
4. next the relative weights are estimated.
5. Lastly the priority of the alternatives is determined.

The five normalization techniques that were being studied are: linear max, linear max-min, linear sum, vector normalization and logarithmic normalization. The results of the study show that the linear max method combined with the linear sum method has the best results. We need to combine the two because with the linear max method the columns of the pairwise matrices do not sum to 1 and this is required in the AHP method. This is why we need to re-normalize the results of the linear max method using the linear sum method (Vafaei et al., 2016).

7.2 Data reduction

If the data that is being used has low quality, most of the time the generated models will also have low quality. This can cause the decision-maker to make the incorrect decision. To remove the negative factors in the raw data we can use a technique called data reduction, that is a family of preprocessing techniques that tries to reduce the representation of data without changing its original structure (S. Ramírez-Gallego, García, & Herrera, 2018).

Data reduction methods are a set of techniques that reduce the representation of an original dataset into a smaller subset. Most data mining algorithms have memory constraints and if we exceed these constraints the algorithm can be prohibitive (García et al., 2015). Therefore, data reduction becomes crucial when working with big data generated by smart cities.

7.2.1 Data discretization

Within the data preprocessing phase, discretization is considered one of the most relevant techniques to improve data quality (S. Ramírez-Gallego, García, & Herrera, 2018). Data discretization reduces the data by converting complex continuous attributes into a finite set of discrete intervals.

We can identify four main steps in a discretization process (Fig. 8.). The first step is to sort the continuous values for feature selection. After this cut points are determined based on an evaluation measure. In the third step the intervals are either being merged or being split depending on the operation method of the discretizer. Finally, the discretization process stops when a certain criteria is being met (Ramírez-Gallego et al., 2016).

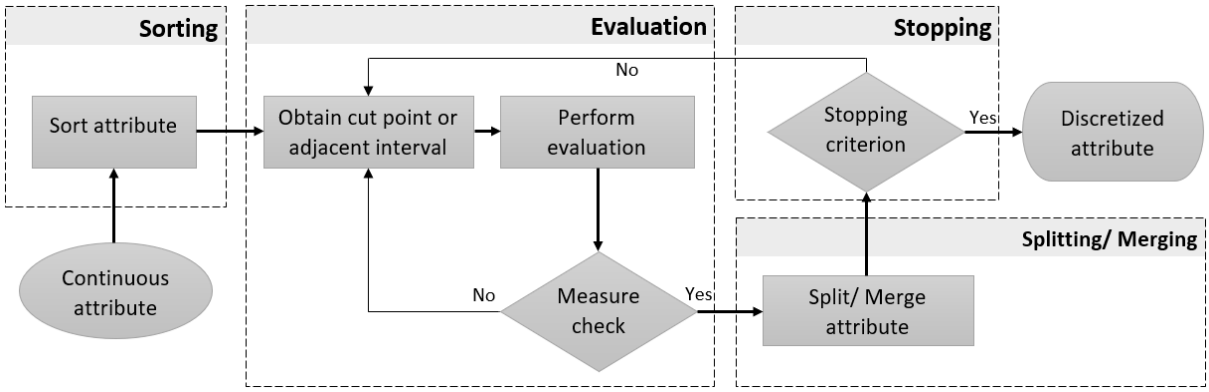

Fig. 8. The steps in a discretization process

Only a few methods of discretization have been developed that can handle big data generated by smart cities (Ramírez-Gallego et al., 2016). In standard discretization algorithms, the whole dataset needs to be in memory. With the introduction of sensors, logs, etc. that continuously generate streams of data it is not possible to have the whole dataset in memory all at ones. Another problem that occurs in online discretization is the introduction of concept drifts. This occurs when the data distribution changes with the introduction of new data. Ideally, the discrete intervals should adapt to the incoming drifts so that the model does not lose accuracy. This can be done in one of three ways (Sergio Ramírez-Gallego et al., 2017):

- By retraining the classification system from scratch every time a new instance or chunk arrives. This is considered a bad option because of the high cost to retrain the system.
- By detecting the changes and retraining the classifier only if changes are significant.
- By using adaptive learning methods that can follow the shifts and drifts in data stream.

S. Ramírez-Gallego, García, and Herrera (2018) propose a solution for the previously mentioned problems by introducing the so called Local Online Fusion Discretizer (LOFD) that is able to smoothly and efficiently adapt to incoming drifts. They also lowered the response times which makes LOFD suitable for high-speed streaming systems.

Together with big data, scalable distribution techniques have emerged. The first programs that were being introduced were MapReduce (Dean & Ghemawat, 2008) and its open source variant Apache Hadoop (Ghazi & Gangodkar, 2015). Later Apache Spark has been introduced to face the challenges of big data by processing large-scale data based on in-memory computation (Karau, Konwinski, Wendell, & Zaharia, 2015).

Ramírez-Gallego et al. (2016) introduces another solution to the problem of data discretization with regards to big data. They discuss the possibilities of Apache Spark to parallelize and use standard discretization methods for big data processing. By parallelizing the process, Apache Spark helps in boosting the performance and accuracy of the discretization methods. García-Gil, Ramírez-Gallego, García, and Herrera (2018) for example, introduced a new method for big data discretization based on Apache Spark called principal components analysis random discretization ensemble.

7.2.2 Feature selection

With the emergence of big data, some technical difficulties arise with regards to the analysis of this data. As a consequence reducing the cardinality of the data without or with minimum loss of information has become a hot topic in the data science community (Cordeiro de Amorim, 2019). There are a lot of feature selection algorithms introduced in the literature but these have two main issues. Most of the time they require labelled samples and secondly, they are not designed to be able to handle the volume of data we have nowadays. Trough recent technologies and new government policies, such as the introduction of open datasets, the amount of data has drastically increased. However, quantity does not imply quality in the world of big data (Kaisler, Armour, Espinosa, & Money, 2013). It is highly likely that this data contains irrelevant features. Features stand for the measurements used to describe entities in a dataset (Cordeiro de Amorim, 2019). Not all of the features in a dataset are useful. Relevant features provide useful information about the learning task, in contrast, the irrelevant features can provide misleading information which can influence the learning performance (Zhao, Sinha, & Ge, 2009). In order to deal with these kinds of datasets, feature selection is introduced. Feature selection reduces the number of features by removing irrelevant features (Guyon & Elisseeff, 2003). In feature selection, a small and more informing subset of features is extracted from the original features. This can lead to shorter processing times and lower memory usage. On top of this, it can help avoid overfitting and the curse of dimensionality.

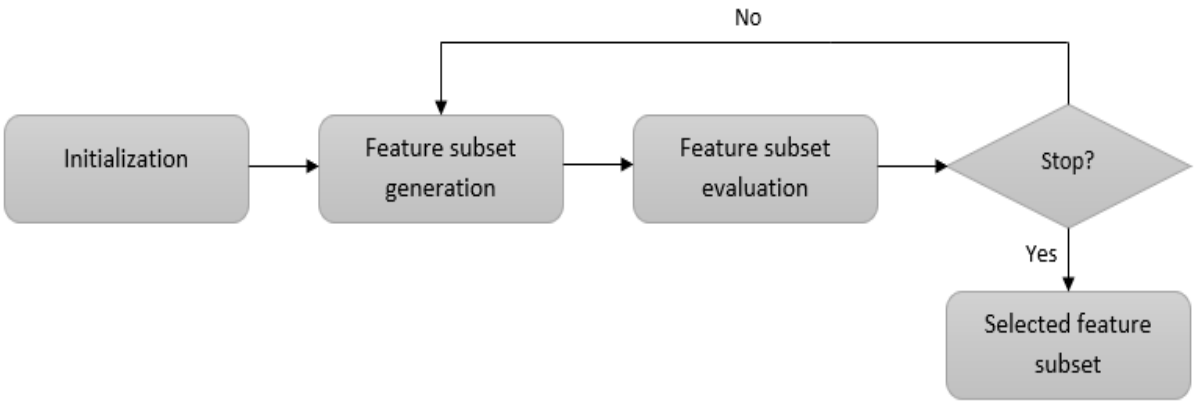

Fig. 9. The steps in a feature selection algorithm

There are four steps that can be identified in a feature selection algorithm. These steps include the initialization, feature subset generation, feature subset evaluation and selecting a feature subset. In Fig. 9. this process is depicted (Nguyen, Xue, & Zhang, 2020).

The two most important steps in this process are the feature subset generation and the feature subset evaluation. In the feature subset generation step, candidates are being generated using a search method. In the feature subset evaluation the goodness of the generated subset in the previous step is measured by an evaluation function (Nguyen et al., 2020).

Furthermore, feature selection approaches can be split up into flat features and structural features. Flat features contain filter models, wrapper models and embedded models (J. Li et al., 2017; Sergio Ramírez-Gallego et al., 2017). In a wrapper approach features subsets are evaluated by looking at their classification performance. In the filter approach, measures are being used to evaluate the subsets. In an embedded approach, the selection process will take place during the training process of the classification algorithm (Nguyen et al., 2020). Among these three approaches, the filter approach is the most efficient because it does not need to go through a learning process. However, wrapper and embedded approaches most of the time have a better classification performance. This is caused by the fact that they consider the interaction between the selected features and the classification algorithm (Nguyen et al., 2020).

Structural features include graph structures, tree structures and group structures (AlNuaimi, Masud, Serhani, & Zaki, 2019). Graph structures contain a set of objects, where some objects are connected to each other. The nodes represent the features and the edges represent the relationship between those features. In a tree structure, the features simulate a hierarchical tree with a root value and subtrees which are in essence children of parent nodes. In a group structure, the most informative subgraphs are extracted from a set of graphs (AlNuaimi et al., 2019).

Most of the feature selection method that exists up to this point focus on supervised methods (Cordeiro de Amorim, 2019). This is a problem in terms of big data because labelling a large sample of entities is becoming increasingly more expensive. This is why we need to start looking at unsupervised feature selection methods, that do not need labelled samples to learn from. Although unsupervised feature selection methods are better, they are not yet perfect. Most of the unsupervised feature selection methods need the full dataset to be in memory. With the growing sizes of datasets this is not feasible. That is why Cordeiro de Amorim (2019) introduces a novel unsupervised feature

selection algorithm that is designed to cope with large datasets that do not fit in the main memory of a computer.

Another problem is the fact that in a smart city context we come across a lot of streaming data. This is a problem because we do not have the full dataset available when we want to perform feature selection. This is where streaming feature selection can offer a solution. Streaming feature selecting helps in reducing the learning complexity of machine learning algorithms. It has been considered as a superior technique for selecting relevant subset features from highly dimensional streaming data (AlNuaimi et al., 2019).

Streaming feature selection is a popular technique used to reduce the size of streaming data. It is mostly used in application such as weather forecasting, transportation, stock markets, etc. It helps with the preparation of big data for real-time analysis processes (AlNuaimi et al., 2019).

If we compare traditional feature selection algorithms with streaming feature selection algorithms, we can clearly see three additional benefits with regards to streaming feature selection. Firstly, the streaming feature selection leads to better and easier models for users and researchers to understand. Secondly, it reduces the training time and avoids the challenges and issues related to high dimensionality. Lastly, it ensures greater generalization by reducing over-fitting (James, Witten, Hastie, & Tibshirani, 2013).

Streaming feature selecting also introduces some new challenges. In a static feature selection scenario, we can assume that all features and instances of the data are captured well in advance. In a streaming context this is not the case, streaming data has an unknown number of instances and features. Other challenges include the extremely high dimensionality of big data (Hilbert, 2016), scalability issues with regards to computational performance (Bolón-Canedo, Sánchez-Maroño, & Alonso-Betanzos, 2015), the stability of feature selection (Tang, Alelyani, & Liu, 2014) and sustainability by means of optimizing resource usage.

7.2.3 Instance selection

Machine learners have high execution times and storage requirements which makes them unusable when dealing with big data (Hernandez-Leal, Carrasco-Ochoa, Martínez-Trinidad, & Olvera-Lopez, 2013). By reducing the size of the dataset into a representative subset we can reduce the memory requirements to store the data and we can accelerate the classification process (Dornaika & Kamal Aldine, 2015). This reduction can be accomplished by using an instance selection. Instance selection increases the efficiency and accuracy of algorithms that are being used for mining big data. Instance selection helps to scale down big data by removing irrelevant, redundant and unreliable data. This in turn helps to reduce the computational resources that are needed to perform the mining task. All this while making a trade-off between the reduction rate and the classification accuracy metrics (Malhat, Menshawy, Mousa, & Sisi, 2020).

With regards to the order in which instances are processed, we can identify five methods (S. Garcia, Derrac, Cano, & Herrera, 2012):

- Incremental methods: this method begins with an empty set and adds instances to the selected subset.
- Decremental methods: here the method starts with the original training dataset and remove instances.
- Batch methods: in this method no instances are removed before all of them are analyzed. They are simply marked for removal and at the end only the unmarked instances are kept.
- Mixed methods: this method starts with a preselected set of instances, then decides to add or delete instances.
- Fixed methods: this last method is a sub-family of the mixed method. Here the removal and additions are the same, keeping the number of instances in the subset fixed.

Considering the type of selection, we can also distinguish three categories. This has to do with the points they remove: whether they are border points, central points or otherwise (S. Garcia et al., 2012). These three categories are:

- Condensation methods: these try to retain the border points.
- Edition methods: these are the opposite of condensation, they try to remove all points not well-classified by their nearest neighbors. These methods achieve smoother boundaries and noise removal.
- Hybrid methods: these stand in the middle of condensation and edition methods. They remove both internal and border points.

The most popular instance selection methods in literature at the moment are the density-based approaches namely the local density-based instance selection algorithm (Carbonera & Abel, 2015) and the centroid density-based instance algorithm (Carbonera & Abel, 2015). The problem with conventional instance selection methods is that they have a high computational complexity (of at least O (n log n)) and cannot handle big amounts of data (Arnaiz-González, Díez-Pastor, Rodríguez, & García-Osorio, 2016). When they have to deal with millions of records, they face problems such as long execution times and large memory requirements. Therefore, there is a need for a scalability option in terms of instance selection.

Provost and Kolluri (1999) mention three ways we can scale up to certain algorithms: the first one is by designing a faster algorithm, the second one is by partitioning the data and the third one is by using relational representation.

Arnaiz-González et al. (2016) introduced two new instance selection algorithms namely LSH-IS-S and LSH-IS-F. Both make use of a technique called locality-sensitive hashing which can be described as an efficient method for checking similarity between elements. The algorithms introduced have a linear complexity of O(n). On top of this, not all instances need to fit in memory. Both algorithms can be seen as incremental methods. The main advantages of the introduced algorithms are low execution times and low memory consumption, which makes them suitable for big data processing.

de Haro-García, García-Pedrajas, and del Castillo (2012) introduce a parallel methodology based on democratization that can scale up any instance selection algorithm in a simple way. This method reduces the execution time of any instance selection algorithm significantly all while keeping its performance in terms of reduction rate and accuracy. Another important benefit of this method is that it reduces the memory storage requirements.

Here follows a list of the most important features of the proposed method:

- The algorithm is executed in parallel.
- There is no need to perform a task using the whole dataset.
- The instance selection algorithm that is being used by each task is a parameter, this means that we can chose any algorithm suitable for our dataset.
- The algorithm has a linear complexity.
- The proposed method can also be used in other data reduction methods such as feature selection, etc.

8. Conclusion

In order to capture the opportunities and benefits of a smart city, the use of big data and artificial intelligence is inevitable. By learning to use and analyze the big data produced in smart cities and gather insides out of the data by applying artificial intelligence and machine learning, we can increase the efficient use of resources in the city, increase the quality of life for the citizens of the city, create a sustainable economic growth and predict future scenarios to help the city in its goal to become more sustainable. But in order to obtain these insights, the data first needs to be preprocessed in order to ensure a fast, precise and valid learning process.

After thoroughly studying the literature with regards to preprocessing techniques for big data generated by smart cities, we came to the conclusion that there are still a lot of gaps that need to be filled. Although there already exist some solutions in the different preprocessing categories, these are scares. A solution that is mentioned a lot is working with scalable platforms like, for example, Apache Spark. This way we can still use already existing techniques for data preprocessing and scale them so they can handle the big amounts of data generated by smart cities. Without such a scalable platform the solutions that already exist would give a time out because of the memory constraints most of these solutions have.

Another challenge regarding big data preprocessing that was prominent in literature is the fact that in a smart city context there is a lot of streaming data which means that not all data is available when we want to start preprocessing it. This is a problem because most preprocessing solution requires the whole dataset to be in memory in order to function properly. This is why stream preprocessing could be an interesting research topic in future works.

A third challenge that can cause a lot of problems is the heterogeneity of the data being generated in smart cities. Because smart cities have a lot of sensors, camera's, smart devices, etc. The amount of different data sources is immense and all these different sources may have different formats in which they store data. This causes problems when we need to integrate all the data together to be processed and extract useful information out of it. In order to overcome this issue, future research should focus on a way to homogenize the way data is being stored from different sources in smart cities. This will result in less errors that are being introduced in the integration process which makes the analysis process quicker and more accurate.

9. References

Al Nuaimi, E., Al Neyadi, H., Mohamed, N., & Al-Jaroodi, J. (2015). Applications of big data to smart cities. *Journal of Internet Services and Applications, 6*(1), 25.

Al Nuaimi, E., Al Neyadi, H., Mohamed, N., & Al-Jaroodi, J. (2015). Applications of big data to smart cities. *Journal of Internet Services and Applications, 6*(1), 1-15. doi:10.1186/s13174-015-0041-5

Alam, M., Abid, F., Guangpei, C., & Yunrong, L. V. (2020). Social media sentiment analysis through parallel dilated convolutional neural network for smart city applications. *Computer Communications, 154*, 129-137. doi:https://doi.org/10.1016/j.comcom.2020.02.044

Allam, Z., & Dhunny, Z. A. (2019). On big data, artificial intelligence and smart cities. *Cities, 89*, 80-91.

Allam, Z., & Newman, P. (2018). Redefining the Smart City: Culture, Metabolism and Governance. *Smart Cities, 1*, 4. doi:10.3390/smartcities1010002

AlNuaimi, N., Masud, M. M., Serhani, M. A., & Zaki, N. (2019). Streaming feature selection algorithms for big data: A survey. *Applied Computing and Informatics*. doi:https://doi.org/10.1016/j.aci.2019.01.001

Arnaiz-González, Á., Díez-Pastor, J.-F., Rodríguez, J. J., & García-Osorio, C. (2016). Instance selection of linear complexity for big data. *Knowledge-Based Systems, 107*, 83-95. doi:https://doi.org/10.1016/j.knosys.2016.05.056

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798-1828.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-Based Systems, 86*, 33-45.

Camero, A., & Alba, E. (2019). Smart City and information technology: A review. *Cities, 93*, 84-94. doi:https://doi.org/10.1016/j.cities.2019.04.014

Carbonera, J. L., & Abel, M. (2015). *A density-based approach for instance selection.* Paper presented at the 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI).

Chen, Y., Tong, Z., Zheng, Y., Samuelson, H., & Norford, L. (2020). Transfer learning with deep neural networks for model predictive control of HVAC and natural ventilation in smart buildings. *Journal of Cleaner Production, 254*, 119866. doi:https://doi.org/10.1016/j.jclepro.2019.119866

Chu, X., Morcos, J., Ilyas, I. F., Ouzzani, M., Papotti, P., Tang, N., & Ye, Y. (2015). KATARA: reliable data cleaning with knowledge bases and crowdsourcing. *Proceedings of the VLDB Endowment, 8*(12), 1952-1955.

Cochinwala, M., Kurien, V., Lalk, G., & Shasha, D. (2001). Efficient data reconciliation. *Information Sciences, 137*(1-4), 1-15.

Cordeiro de Amorim, R. (2019). Unsupervised feature selection for large data sets. *Pattern Recognition Letters, 128*, 183-189. doi:https://doi.org/10.1016/j.patrec.2019.08.017

de Haro-García, A., García-Pedrajas, N., & del Castillo, J. A. R. (2012). Large scale instance selection by means of federal instance selection. *Data & Knowledge Engineering, 75*, 58-77. doi:https://doi.org/10.1016/j.datak.2012.03.002

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Commun. ACM, 51*(1), 107–113. doi:10.1145/1327452.1327492

Dekel, O. (2009). *From online to batch learning with cutoff-averaging.* Paper presented at the Advances in neural information processing systems.

Do, H.-H., & Rahm, E. (2007). Matching large schemas: Approaches and evaluation. *Information Systems, 32*(6), 857-885. doi:https://doi.org/10.1016/j.is.2006.09.002

Doan, A., Domingos, P., & Halevy, A. (2003). Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning, 50*(3), 279-301.

Doan, A., Domingos, P., & Halevy, A. Y. (2001). *Reconciling schemas of disparate data sources: A machine-learning approach.* Paper presented at the Proceedings of the 2001 ACM SIGMOD international conference on Management of data.

Dornaika, F., & Kamal Aldine, I. (2015). Decremental Sparse Modeling Representative Selection for prototype selection. *Pattern Recognition, 48*(11), 3714-3727. doi:https://doi.org/10.1016/j.patcog.2015.05.018

Fan, C., Ding, C., Zheng, J., Xiao, L., & Ai, Z. (2020). Empirical Mode Decomposition based Multi-objective Deep Belief Network for short-term power load forecasting. *Neurocomputing, 388*, 110-123. doi:https://doi.org/10.1016/j.neucom.2020.01.031

Fellegi, I. P., & Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association, 64*(328), 1183-1210.

García-Gil, D., Luengo, J., García, S., & Herrera, F. (2019). Enabling Smart Data: Noise filtering in Big Data classification. *Information Sciences, 479*, 135-152. doi:https://doi.org/10.1016/j.ins.2018.12.002

García-Gil, D., Ramírez-Gallego, S., García, S., & Herrera, F. (2018). Principal Components Analysis Random Discretization Ensemble for Big Data. *Knowledge-Based Systems, 150*, 166-174. doi:https://doi.org/10.1016/j.knosys.2018.03.012

Garcia, F. C. C., Creayla, C. M. C., & Macabebe, E. Q. B. (2017). Development of an Intelligent System for Smart Home Energy Disaggregation Using Stacked Denoising Autoencoders. *Procedia Computer Science, 105*, 248-255. doi:https://doi.org/10.1016/j.procs.2017.01.218

Garcia, S., Derrac, J., Cano, J., & Herrera, F. (2012). Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(3), 417-435. doi:10.1109/TPAMI.2011.142

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining*: Springer.

Ghazi, M. R., & Gangodkar, D. (2015). Hadoop, MapReduce and HDFS: A Developers Perspective. *Procedia Computer Science, 48*, 45-50. doi:https://doi.org/10.1016/j.procs.2015.04.108

Girtelschmid, S., Steinbauer, M., Kumar, V., Fensel, A., & Kotsis, G. (2013). *Big data in large scale intelligent smart city installations.* Paper presented at the Proceedings of International Conference on Information Integration and Web-based Applications & Services.

Gohar, M., Muzammal, M., & Ur Rahman, A. (2018). SMART TSS: Defining transportation system behavior using big data analytics in smart cities. *Sustainable Cities and Society, 41*, 114-119. doi:https://doi.org/10.1016/j.scs.2018.05.008

Golov, N., & Rönnbäck, L. (2017). Big Data normalization for massively parallel processing databases. *Computer Standards & Interfaces, 54*, 86-93. doi:https://doi.org/10.1016/j.csi.2017.01.009

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research, 3*(Mar), 1157-1182.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Ullah Khan, S. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems, 47*, 98-115. doi:https://doi.org/10.1016/j.is.2014.07.006

He, X., Huang, T., Li, C., Che, H., & Dong, Z. (2015). A recurrent neural network for optimal real-time price in smart grid. *Neurocomputing, 149*, 608-612. doi:https://doi.org/10.1016/j.neucom.2014.08.014

Hernandez-Leal, P., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Olvera-Lopez, J. A. (2013). InstanceRank based on borders for instance selection. *Pattern Recognition, 46*(1), 365-375. doi:https://doi.org/10.1016/j.patcog.2012.07.007

Hilbert, M. (2016). Big data for development: A review of promises and challenges. *Development Policy Review, 34*(1), 135-174.

Isah, H., Abughofa, T., Mahfuz, S., Ajerla, D., Zulkernine, F., & Khan, S. (2019). A Survey of Distributed Data Stream Processing Frameworks. *IEEE Access, 7*, 154300-154316.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112): Springer.

Joachims, T. (1999). Making large-scale support vector machine learning practical, Advances in Kernel Methods. *Support vector learning*.
John Walker, S. (2014). Big Data: A Revolution That Will Transform How We Live, Work, and Think. *International Journal of Advertising, 33*(1), 181-183. doi:10.2501/IJA-33-1-181-183
Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science, 349*(6245), 255-260.
Jung, J. J. (2011). Semantic preprocessing for mining sensor streams from heterogeneous environments. *Expert Systems with Applications, 38*(5), 6107-6111. doi:https://doi.org/10.1016/j.eswa.2010.11.017
Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). *Big data: Issues and challenges moving forward.* Paper presented at the 2013 46th Hawaii International Conference on System Sciences.
Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2015). *Learning Spark: Lightning-Fast Big Data Analytics*: O'Reilly Media, Inc.
Khayyat, Z., Ilyas, I. F., Jindal, A., Madden, S., Ouzzani, M., Papotti, P., . . . Yin, S. (2015). *Bigdansing: A system for big data cleansing.* Paper presented at the Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data.
Kim, J., Moon, J., Hwang, E., & Kang, P. (2019). Recurrent inception convolution neural network for multi short-term load forecasting. *Energy and Buildings, 194*, 328-341. doi:https://doi.org/10.1016/j.enbuild.2019.04.034
Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K., & Lee, D. (2003). A Taxonomy of Dirty Data. *Data Min. Knowl. Discov., 7*(1), 81–99. doi:10.1023/a:1021564703268
Lan, J., Guo, Q., & Sun, H. (2018). Demand Side Data Generating Based on Conditional Generative Adversarial Networks. *Energy Procedia, 152*, 1188-1193. doi:https://doi.org/10.1016/j.egypro.2018.09.157
Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Retrieved from http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf
Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys (CSUR), 50*(6), 1-45.
Li, L. (2012). *Data quality and data cleaning in database applications.* Edinburgh Napier University,
Lim, C., Kim, K.-J., & Maglio, P. P. (2018). Smart cities with big data: Reference models, challenges, and considerations. *Cities, 82*, 86-99. doi:https://doi.org/10.1016/j.cities.2018.04.011
Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data mining and knowledge discovery, 6*(4), 393-423.
Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* (Vol. 454): Springer Science & Business Media.
Liu, H., & Motoda, H. (2013). *Instance selection and construction for data mining* (Vol. 608): Springer Science & Business Media.
Loy-Benitez, J., Li, Q., Nam, K., & Yoo, C. (2020). Sustainable subway indoor air quality monitoring and fault-tolerant ventilation control using a sparse autoencoder-driven sensor self-validation. *Sustainable Cities and Society, 52*, 101847. doi:https://doi.org/10.1016/j.scs.2019.101847
Malhat, M., Menshawy, M. E., Mousa, H., & Sisi, A. E. (2020). A new approach for instance selection: Algorithms, evaluation, and comparisons. *Expert Systems with Applications, 149*, 113297. doi:https://doi.org/10.1016/j.eswa.2020.113297
Mohammadi, M., & Al-Fuqaha, A. (2018). Enabling cognitive smart cities using big data and machine learning: Approaches and challenges. *IEEE Communications magazine, 56*(2), 94-101.
Monge, A. E., & Elkan, C. (1996). *The Field Matching Problem: Algorithms and Applications.* Paper presented at the Kdd.
Müller, H., & Freytag, J.-C. (2005). *Problems, methods, and challenges in comprehensive data cleansing*: Professoren des Inst. Für Informatik.

Nguyen, B. H., Xue, B., & Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation, 54*, 100663. doi:https://doi.org/10.1016/j.swevo.2020.100663

Ni, L., Wang, D., Singh, V. P., Wu, J., Wang, Y., Tao, Y., & Zhang, J. (2020). Streamflow and rainfall forecasting by two long short-term memory-based models. *Journal of Hydrology, 583*, 124296. doi:https://doi.org/10.1016/j.jhydrol.2019.124296

Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. *Future Generation Computer Systems, 91*, 620-633. doi:https://doi.org/10.1016/j.future.2018.06.046

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences, 30*(4), 431-448. doi:https://doi.org/10.1016/j.jksuci.2017.06.001

Pandey, S., Singh, P. R., & Tian, J. (2020). An image augmentation approach using two-stage generative adversarial network for nuclei image segmentation. *Biomedical Signal Processing and Control, 57*, 101782. doi:https://doi.org/10.1016/j.bspc.2019.101782

Pei, S., Qin, H., Zhang, Z., Yao, L., Wang, Y., Wang, C., . . . Yi, T. (2019). Wind speed prediction method based on Empirical Wavelet Transform and New Cell Update Long Short-Term Memory network. *Energy Conversion and Management, 196*, 779-792. doi:https://doi.org/10.1016/j.enconman.2019.06.041

Provost, F., & Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Data mining and knowledge discovery, 3*(2), 131-169.

Rahman, A., Srikumar, V., & Smith, A. D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy, 212*, 372-385. doi:https://doi.org/10.1016/j.apenergy.2017.12.051

Ramírez-Gallego, S., García, S., Benítez, J. M., & Herrera, F. (2018). A distributed evolutionary multivariate discretizer for Big Data processing on Apache Spark. *Swarm and Evolutionary Computation, 38*, 240-250. doi:https://doi.org/10.1016/j.swevo.2017.08.005

Ramírez-Gallego, S., García, S., & Herrera, F. (2018). Online entropy-based discretization for data streaming classification. *Future Generation Computer Systems, 86*, 59-70. doi:https://doi.org/10.1016/j.future.2018.03.008

Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., & Herrera, F. (2017). A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing, 239*, 39-57. doi:https://doi.org/10.1016/j.neucom.2017.01.078

Ramírez-Gallego, S., García, S., Mouriño-Talín, H., Martínez-Rego, D., Bolón-Canedo, V., Alonso-Betanzos, A., . . . Herrera, F. (2016). Data discretization: taxonomy and big data challenge. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 6*(1), 5-21.

Ridzuan, F., & Wan Zainon, W. M. N. (2019). A Review on Data Cleansing Methods for Big Data. *Procedia Computer Science, 161*, 731-738. doi:https://doi.org/10.1016/j.procs.2019.11.177

Rönnbäck, L., Regardt, O., Bergholtz, M., Johannesson, P., & Wohed, P. (2010). Anchor modeling—Agile information modeling in evolving data environments. *Data & Knowledge Engineering, 69*(12), 1229-1253.

Sabar, N. R., Turky, A., Song, A., & Sattar, A. (2019). An evolutionary hyper-heuristic to optimise deep belief networks for image reconstruction. *Applied Soft Computing*, 105510. doi:https://doi.org/10.1016/j.asoc.2019.105510

Serrano, E., & Bajo, J. (2019). Deep neural network architectures for social services diagnosis in smart cities. *Future Generation Computer Systems, 100*, 122-131. doi:https://doi.org/10.1016/j.future.2019.05.034

Silva, D. A. N. S., Souza, L. C., & Motta, G. H. M. B. (2016). An instance selection method for large datasets based on Markov Geometric Diffusion. *Data & Knowledge Engineering, 101*, 24-41. doi:https://doi.org/10.1016/j.datak.2015.11.002

Singh, D., & Reddy, C. K. (2014). A survey on platforms for big data analytics. *Journal of Big Data, 2*(1), 8. doi:10.1186/s40537-014-0008-6

Soomro, K., Bhutta, M. N. M., Khan, Z., & Tahir, M. A. (2019). Smart city big data analytics: An advanced review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9*(5), e1319. doi:10.1002/widm.1319

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. *Data classification: Algorithms and applications*, 37.

Tantalaki, N., Souravlas, S., & Roumeliotis, M. (2019). A review on big data real-time stream processing and its scheduling techniques. *International Journal of Parallel, Emergent and Distributed Systems*, 1-31.

Torrecilla, J. L., & Romo, J. (2018). Data learning from big data. *Statistics & Probability Letters, 136*, 15-19. doi:https://doi.org/10.1016/j.spl.2018.02.038

Trends, G. (2020). The popularity of big data and AI. Retrieved from https://trends.google.com/trends/explore?date=all&q=big%20data,artificial%20intelligence

Vafaei, N., Ribeiro, R. A., & Camarinha-Matos, L. M. (2016). *Normalization techniques for multi-criteria decision making: analytical hierarchy process case study.* Paper presented at the doctoral conference on computing, electrical and industrial systems.

Verykios, V. S., Elmagarmid, A. K., & Houstis, E. N. (2000). Automating the approximate record-matching process. *Information Sciences, 126*(1-4), 83-98.

Walunj Swapnil, K., Yadav Anil, H., & Gupta, S. (2016). Big Data: Characteristics, Challenges and Data Mining. *International Journal of Computer Applications, 975*, 8887.

Wang, H., Li, M., Bu, Y., Li, J., Gao, H., & Zhang, J. (2014). *Cleanix: A big data cleaning parfait.* Paper presented at the Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management.

Wang, L., Fan, X., Chen, J., Cheng, J., Tan, J., & Ma, X. (2020). 3D object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities. *Sustainable Cities and Society, 54*, 102002. doi:https://doi.org/10.1016/j.scs.2019.102002

Yakout, M., Berti-Équille, L., & Elmagarmid, A. K. (2013). *Don't be SCAREd: use SCalable Automatic REpairing with maximal likelihood and bounded changes.* Paper presented at the Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data.

Zhang, J. M., Harman, M., Ma, L., & Liu, Y. (2020). Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*.

Zhao, H., Sinha, A. P., & Ge, W. (2009). Effects of feature construction on classification performance: An empirical study in bank failure prediction. *Expert Systems with Applications, 36*(2), 2633-2644.

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing, 237*, 350-361. doi:https://doi.org/10.1016/j.neucom.2017.01.026