The Role of Machine Learning in Big Data Analytics: Current Practices and Challenges

Hector A. Duran-Limon, Arturo Chavoya, and Martha Hernández-Ochoa

1 Introduction

Huge amounts of data are generated on a daily basis by diverse application domains. Social media, mobile phones, sensors, and medical imaging among others are examples of data sources. The exponential growth of both the Internet and data digitalization has fueled the generation of high volumes of data. According to the International Data Corporation, such generation of data will increase from 33 zettabytes in 2018 to 175 zettabytes in 2025 [1]. For instance, regarding social media data generated in 1 minute in October 2021 [2], we have that 694 million songs were streamed in the USA, there were 4.2 million Google searches, 210 million emails were sent, and 21 million snaps were created.

Big data analytics (BDA) enables the extraction of valuable information from large datasets that are obtained from multiple sources. Such valuable information involves patterns and correlations that can help organizations to make better decisions [3–5]. Laney defined big data in terms of Volume, Velocity, and Variety [6]. Two more Vs were added later: Value and Veracity [7]. Currently, the 5 Vs are the most widely accepted conceptualization of big data. Volume refers to large volumes of data that increase exponentially with time. Velocity regards the speed at which data are generated and processed. The diverse number of data sources and heterogeneity of the data denote Variety. In addition, Value refers to the extracted patterns

H. A. Duran-Limon (✉) · A. Chavoya
Information Systems, CUCEA, University of Guadalajara, Guadalajara, Mexico
e-mail: hduran@cucea.udg.mx; achavoya@cucea.udg.mx

M. Hernández-Ochoa
Knowledge Fundamentals, CUNORTE, University of Guadalajara, Guadalajara, Mexico
e-mail: martha.ochoa@cunorte.udg.mx

M. Mora et al. (eds.), *Development Methodologies for Big Data Analytics Systems*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-031-40956-1_2

and correlations that can help to make better decisions. Lastly, Veracity involves the level of confidence on the data.

The main elements of BDA include descriptive analytics, predictive analytics, and prescriptive analytics. Descriptive analytics is in charge of describing what has happened, where past and current patterns can be identified and highlighted. In contrast, predictive analytics identify correlations among different variables whereby the value of a variable can be forecasted when other variables suffer changes. On the other hand, prescriptive analytics helps to find the best option or recommendation under uncertainty conditions.

The data can be in different formats, namely, unstructured, semi-structured, and structured. Unstructured data do not have a structural organization and comprise videos, audios, pictures, and online text. Semi-structured data is data that is partially structured; an example of data in this format is XML data in the web, which employ an informal tag-type format for organizing the data. Lastly, structured data can be normally extracted from relational databases and spreadsheets. Crucially, most of the data is either unstructured or semi-structured.

Traditional approaches such as data warehousing and the use of a classic relational database management system (RDBMS) have become impractical to analyze such unstructured and semi-structured data [8]. On the other hand, machine learning (ML) algorithms have proven to be successful in analyzing vast amounts of data [4, 7, 9–11].

Machine learning is part of the artificial intelligence field, which involves algorithms and statistical models that are able to learn and adapt without following explicit instructions to do so [12]. ML algorithms can be categorized in three main classes: unsupervised learning, supervised learning, and reinforcement learning. Unsupervised learning is used to find a hidden structure on unlabeled data. This kind of algorithm groups data into clusters. Unsupervised learning can be used, for example, for customer segmentation and pattern classification. In contrast, with supervised learning, the data must be already labeled or structured. Algorithms of this kind infer a function from the labeled data that enables them to make either predictions or decisions. There are two subcategories of supervised learning: classification and regression. The former is used to identify the class of a data point. Classification can be used for speech recognition, image recognition, and fraud detection, among others. On the other hand, regression algorithms are employed for prediction. The value of the dependent variable is predicted from a continuous dataset. The independent variables are used for modeling or training. Regression has been applied, for example, for weather forecasting or predicting the value of a stock in the stock market. Lastly, reinforcement learning is an approach whereby each type of action is given a different reward. By using trial and error, the actions that have the greatest reward are learned. The goal of reinforcement learning is to find the policy that maximizes the reward function. This type of method is commonly applied in gaming and robotics.

Some of the most important domain areas of BDA include health and human welfare, weather forecasting, customer transactions, customer preferences, financial analysis, and social networking and the Internet. In this chapter, we focus on describing some of the most widely used ML algorithms and platforms for BDA, as

well as analyzing the role that the use of ML has played in some of these domain areas. More specifically, we present the use of ML for BDA in the areas of healthcare, weather forecasting, and social networking and the Internet.

The chapter is organized as follows. We first present some of the most widely used ML algorithms in BDA. Then, we present the most commonly used distributed platforms for processing big data. This section is followed by a review of a selection of three important domain areas where BDA is employed. Finally, some concluding remarks are drawn.

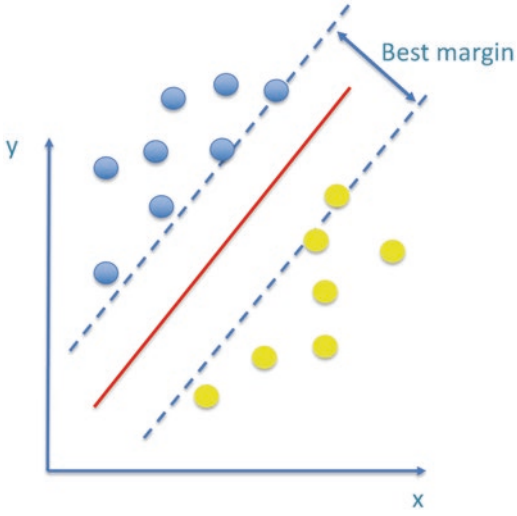
2 Machine Learning Techniques

In this section, we present some of the machine learning techniques that are frequently used in BDA [4].

2.1 Support Vector Machines

Support vector machines (SVMs) [13] are supervised learning models that are employed for binary classification and regression analysis of data. The training algorithm is a non-probabilistic binary linear classifier that classifies the trained data into one of two categories. The SVM maps training data into points in space whereby the width of the gap between the two categories is maximized. New data is then mapped to the same space in which it is predicted to which category it belongs and what position in space it takes. A data point involves a number of features. A hyperplane is used to classify these data points into two classes. It is desirable that the margin or distance from the hyperplane to the nearest point of each side is maximized, as shown in Fig. 1.

Fig. 1 Linear SVM



In many cases, it is not always possible to separate perfectly both classes. Soft margin classifiers (also called support vector classifiers) allow that certain data points be in the incorrect side, so that the distance of the hyperplane is maximized from the majority of the data points of both sides, obtaining a more robust classifier with a better predictive capacity when applied to new data points. In case the separation between the groups is nonlinear, the dimensions of the space can be expanded. In fact, the dimension of the hyperplane depends on the number of features (i.e., data inputs characterizing each data point). A kernel function can be used to efficiently map the input data into high-dimensional spaces.

Support vector regression (SVR) is a variant of SVM. This variant employs a regression scheme used for predicting values. In SVM, the margins do not include data, whereas in SVR the margin lines are chosen so that they cover all data (hard margin) or permit some violation (soft margin). These margins involve a tolerance error (epsilon). The aim here is to find the function that represents a line that is between the two margins, as shown in red in Fig. 2. SVR also allows a nonlinear regression analysis.

The parameter needed by SVMs is the so-called soft margin parameter, which is normally indicated with C, and the kernel function. In the case of SVRs, an additional parameter called ϵ is needed. In cases where there is much noise, ϵ must be selected accordingly to reflect the variance of noise. In cases where no noise is present, we have an interpolation problem and ϵ corresponds to the preset interpolation accuracy. The larger the value of ϵ, the smaller the number of support vectors required, and vice versa. In addition, a procedure of cross validation is commonly employed for the selection of both the kernel function and the optimal value of C.

A parallel implementation of SVM that employs MapReduce to reduce the training time is presented in [14].

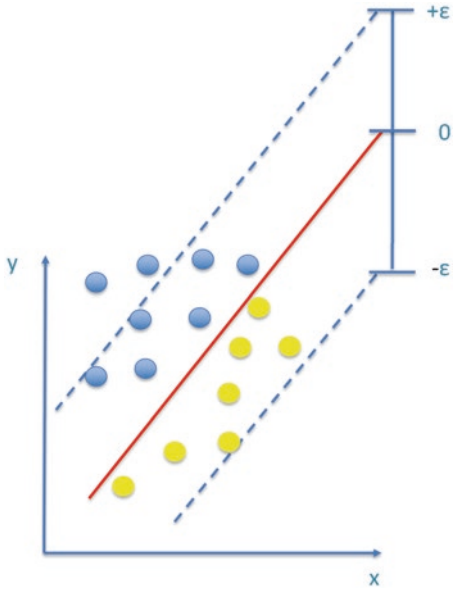


Fig. 2 Linear SVR

2.2 Decision Trees

Decision trees are nonparametric supervised learning models that are used for classification and regression analysis of data. Decision trees are able to carry out a multi-class classification on a dataset. There are various methodologies for generating decision trees. The classification and regression trees (CART) algorithm [15] is the most widely used, which is described below.

A decision tree is a binary tree that can be constructed by splitting the data input into subsets of data based on an attribute evaluation. There are two kinds of nodes: decision nodes and leaf nodes. Decision nodes contain a condition to split the data, whereas leaf nodes help to decide the class of a new data point. Decision trees that classify data into categories are called classification trees, whereas decision trees that predict values are called regression trees. In the case of classification trees, the best split is found using the Gini impurity index, which is equivalent to using the entropy or information-gain criterion. On the other hand, in the case of regression trees, the best split is the split that minimizes the residual sum of squares (RSS) of the observed and predicted values.

A recursive partitioning is carried out in which this splitting process is performed on each derived branch. A decision tree is split down from the root to leaf nodes. The data points are located in axis-parallel (hyper-) rectangles, as shown in Fig. 3. In case of overfitting, there are mechanisms that help address this issue. Pruning is one of such mechanisms, which involves the process of removing a branch from a decision node.

Once the decision tree is constructed, the predictions are carried out on the leaves where the mode is taken for classification, whereas the mean is used for regression.

One of the main advantages of regression trees over other ML approaches is that the graphical model of a regression tree helps to understand the phenomenon represented in the data. That is, the features located in the upper nodes of the tree play a more important role in the prediction process. For instance, regarding weather

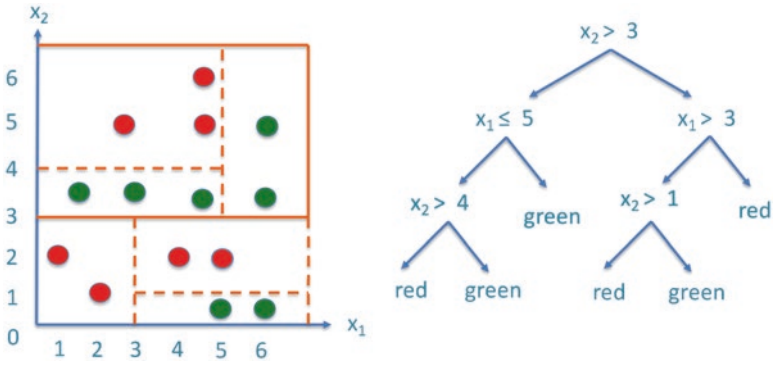


Fig. 3 Data points space of a classification tree

forecast, in case of having as an upper node, let us say wind speed, and as a lower node moisture, this would indicate that wind speed has a higher impact on the temperature than moisture.

There are a number of parallel versions of decision trees implemented with MapReduce, such as [16, 17].

2.3 Clustering Algorithms

Clustering algorithms create clusters of datasets, whose members are more closely related to each other than to members of other clusters. The main idea of these algorithms is to distribute input data into clusters without requiring labels for the training set [18]. The behavior of these algorithms is shown in Fig. 4. In this kind of algorithm, two requirements are met: (1) each cluster must have a set and (2) at least one element must exist in the cluster [19].

One popular clustering algorithm is *k*-means, which is described as follows. *K*-means groups datasets based on closeness to each other using the Euclidean distance [20], where the aim is to minimize the distance between the elements within the cluster, and *k* is the number of clusters. The *k*-means algorithm consists of assigning each element from the dataset to the defined *k*-th cluster closer to this element. For each iteration, the *k*-th cluster is calculated once the associated elements are observed to the related cluster. This process is iterative until all elements from the dataset are assigned to the clusters [21]. For *n* elements and a dimension *d*, the *k*-means algorithm complexity is $O(k*n*d)$, so it is computationally efficient [22]. The steps of the algorithm are defined as follows:

1. Define the number of clusters *k*.
2. Select *k* random elements from the dataset as centroids. In other words, select one element (called centroid) for each cluster.
3. All the elements are assigned to the closest cluster centroid.
4. Recalculate the *k*-th cluster once all the elements are associated with related clusters.
5. Repeat until one of the next criteria is met.

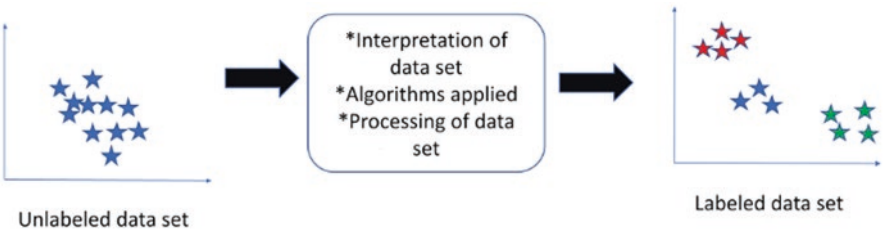


Fig. 4 Behavior of clustering algorithms

(a) The *k*-th value is reached.
(b) The elements remain in the same cluster.
(c) Once the new cluster is defined, the centroid is the same.

Some advantages of *k*-means are as follows: it is based on mathematical ideas, it is easily implemented, and it has fast convergence [23]. However, there are some drawbacks such as the following: when a global cluster is used, it is not effective; also, the size and density of the cluster is not handled by the algorithm [20]; with the traditional *k*-means algorithm, it is difficult to analyze a massive dataset; and prediction of a *k* value is hard. *K*-means is utilized for document classification, insurance fraud detection, customer segmentation, rideshare data analysis, automatic clustering of IT alerts, and call record details analysis, among other applications [22]. There are several improvements of this algorithm that have been made in different research works [21, 24–27], for example, "spectral clustering," which uses standard linear algebra methods. This algorithm is built on graph Laplacian matrices [21].

The main steps are described as follows:

1. Create a similarity graph to cluster between *N* objects.
2. Compute the first *k* eigenvectors of its Laplacian matrix.
3. Run *k*-means to separate objects into *k* classes.

Distributed clustering algorithms are classified into homogeneous and heterogeneous, based on the type of dataset they process. Most distributed clustering algorithms are focused on homogenous datasets. Some distributed clustering algorithms are described next. In [28], the authors proposed a distributed dynamic clustering algorithm (DDCA), which is based on *k*-means and a tree topology. Another article presented a Noise-based *k*-means that has better results for urban hotspots over *k*-means [29]. In [30] a new approach for very large spatial heterogeneous datasets is proposed, which is based on the *k*-means algorithm but generates clusters dynamically.

2.4 Artificial Neural Networks

Some of the most used ML techniques in BDA are the different variants of artificial neural networks (ANNs) [31]. ANNs are a family of models inspired in biological neural networks and consist of at least one input layer and one output layer of nodes, where each node corresponds to an artificial neuron and the nodes in one layer are connected to the nodes in the adjacent layer. The nodes in the input layer receive the values introduced in the model, and the nodes in the output layer produce the response of the model. There can also be one or more intermediate layers, which are known as "hidden" layers. The role of the hidden layers is to discover features that are informative for the desired goal. The connection among two nodes is represented by a function, whose parameters need to be adjusted by training the network with input data. An ANN that only contains an input layer and an output layer and all the nodes in one layer are connected to all nodes in the other layer is known as a perceptron [32].

Deep learning models are a special kind of ANNs that use a "deep" architecture, that is, one that contains more than one hidden layer. Deep learning methods are very effective when dealing with a large number of training samples. The current success of deep learning is due to a great extent to three factors: (1) recent advances in the development of high-performance central processing units (CPUs) and graphics processing units (GPUs), (2) the availability of big data, and (3) recent developments in ML algorithms. Unlike shallow architectures that depend on the availability of expert human knowledge to train the supervised models, deep models can discover useful features from data in a hierarchical way from fine to abstract in an unsupervised manner, where each layer in the network discovers new characteristics of the data in an incremental way. Deep learning models can be classified as either multilayer neural networks that take nonstructured vector values as input or convolutional neural networks (CNNs) that take multidimensional structured values as input. Within the first category, three widely used deep models are stacked autoencoders, deep belief networks, and deep Boltzmann machines. These models differ in the way the connections among layers are made, whether they are directed or undirected, and the direction of the connection (toward the output layer or toward the input layer). On the other hand, CNNs use the spatial and configurational information of adjacent data points that cannot exist in the vectorized data used by the multilayered neural networks. This characteristic makes CNNs especially suitable to analyze 2D or 3D data (such as images) to discover patterns of interest [32].

3 Open-Source Platforms for Big Data Analytics

We present some of the most used open-source platforms for BDA after a brief introduction to the MapReduce model.

3.1 MapReduce

MapReduce [33] is a programming model developed by Google for processing big data on a distributed platform. The data is processed in batches in parallel by using either clusters or grid systems.

This programming model involves two main operations: Map and Reduce. The former involves splitting and mapping the data, whereas the latter performs a summary operation. The Map function takes input key-value pairs (K1, V1), which are transformed to different key-value pairs (K2, V2). Afterward, a shuffling process is carried out, whereby all pairs with the same key (K2) are collected and grouped according to their key value. MapReduce then uses the Reduce function to process the data of each group, which is transformed into different key-value pairs (K3, V3).

Both the Map and the Reduce functions are run in parallel. Data inputs and data outputs are stored in a distributed file system.

The performance and scalability of MapReduce may be negatively impacted when there are large amounts of data that need to be written by the Map operation. Also, the communication costs commonly overcome computation costs given that many MapReduce implementations employ a distributed storage in order to address crash recovery.

MapReduce is useful for different kinds of applications, such as distributed pattern-based searching, distributed sorting, web access log stats, ML, and document clustering, among others. There are a number of frameworks implementing MapReduce such as Hadoop and Spark, which are presented below.

3.2 Apache Hadoop

Apache Hadoop is a parallel computing framework whose main function is to store and process large datasets across clusters of computers [34]. Hadoop is designed to scale up from single servers to thousands of nodes, each one having its own storage and processing. It consists of four modules: (1) Hadoop Common, which includes utilities to support the other Hadoop modules; (2) Hadoop Distributed File System (HDFS), which is a distributed file system that gives high-throughput access to application data; (3) Hadoop YARN (Yet Another Resource Negotiator), which is a framework that provides cluster resource management and job scheduling for managing the extensive storage resources and keeping track of the processing workload across clusters; and (4) Hadoop MapReduce, which is a YARN-based system for the parallel processing of large datasets contained on HDFS clusters; during a Map step, the master node divides the job into smaller tasks and distributes the resources depending on the task, and after the computations, the Reduce step aggregates all the partial results to produce an integrated solution to the problem [34, 35].

Even though Apache Hadoop is extensively used, it has some drawbacks. One problem with Hadoop is that it is strictly a batch computing platform, and as such, it is not suitable for real-time streaming applications where immediate results are expected. Another problem with Hadoop is the skew problem, which happens when during a Map and Reduce operation there is an imbalance in the time between a Map step and the corresponding Reduce step, which can cause a delay in the execution of one of the steps [34]. Some of the problems with Hadoop are solved with Spark, which is better suited for real-time data processing, but Hadoop is still considered more suitable for BDA in terms of cost, security, and fault tolerance when batch processing is involved [36].

3.3 Apache Spark

Spark is the topmost used tool (34.88%) for BDA among experts in this field, according to [4]. It is a parallel and open-source cluster computing framework developed as an Apache project. Spark was created in 2009 in UC Berkeley's AMPLab [37]. Spark runs on top of an HDFS (Hadoop Distributed File System) infrastructure. Spark also supports SparkQL, Spark Streaming, MLib, and GraphX libraries for ML and data mining. Multi-language and analytics are also supported. Spark is deployed in a big data hybrid (batch and real time) processing model [38]. Spark can access Hadoop Distributed File System (HDFS), Hbase, and Cassandra [39].

Some benefits of using this platform are that it is easier to use, programs run faster (up to 100 times quicker than Hadoop MapReduce [39]), and it has high processing speed. Also, Spark is highly efficient with massive amounts of data and has fault tolerance without replication, reducing read disk, write disk, and the network I/O cost and employing in-memory computation operations. Furthermore, it covers batch, streaming, interactive, and iterative workloads [40]. In Spark, resilient distributed datasets (RDDs) are the main abstraction and provide a way to treat all distributed RAM as a single memory, which provides robustness against data loss. In [38] the authors figured out that Spark performs better than MapReduce for all datasets due to its in-memory computation, less overhead in setting up jobs for every iteration, and lower network I/O cost. For these reasons, Spark is in general the preferred choice by experts in big data.

On the other hand, some drawbacks are that Spark consumes more memory in operation than Hadoop, and as such the cost is very high and the latency is higher, so results have lower throughput and iteration processing. Other problems involve that there is no file management system and that there are small file issues, among others.

3.4 Other Open-Source Platforms and Tools

Even though Hadoop and Spark are the main big data platforms currently in use, there are other platforms that can be useful under specific circumstances, and some of them can even interact with Hadoop or Spark. Some of these platforms are listed next.

Apache Storm can be an alternative to Hadoop MapReduce when there is a heavy need for real-time big data processing. The main difference between Hadoop and Storm is that the former runs jobs, whereas the latter runs topologies, and while a MapReduce job can finish, a topology continues processing incoming data until the user terminates the process [34].

Apache Flink is a platform that provides real-time processing of data streams, and at the same time it can process historical batch data. Flink offers many libraries,

including support for ML, a graph API, and a table API to process SQL operations, among others [34].

On the other hand, Apache Flume is an agent-based platform that provides reliable, distributed, and accessible web services from various sources to collect, aggregate, and transfer large amounts of streaming data to a centralized data store [41].

Regarding storage systems for BDA, traditional SQL database systems are not suitable for storing large quantities of unstructured data, such as text documents. Consequently, in these cases, there has been a need to transition to NoSQL databases for storing this kind of data to be processed by BDA systems. In a recent systematic literature review, the NoSQL storage tools most cited in the publications were MongoDB, Hbase, CouchDB, Cassandra, and Neo4J, although other storage systems such as BigTable, HyperTable, and SimpleDB were also cited [42].

As for other tools used for big data, WEKA (Waikato Environment for Knowledge Analysis) is an open-source software that contains, among other things, a collection of implementations of well-known ML algorithms [43]. Another popular tool for the analysis of big data is the R language, which provides a wide variety of statistical and graphics techniques [44]. Finally, some ML algorithms cited in the literature are implemented using general-purpose programming languages, such as Python and C++.

4 Domain Areas of Big Data Analytics

We present some of the main domain areas to which BDA is currently applied. The articles presented in this section were considered based on some inclusion-exclusion criteria, which are described next. The inclusion criteria were as follows: (i) the article is written in the English language; (ii) the article must relate to ML algorithms, BDA, and related platforms and/or tools; (iii) the article was published between the years 2012 and 2022; (iv) the article was published in a journal or conference; (v) the article addresses one of the considered domains; and (vi) the article was selected from a subset of high quality journals and conferences, such as those supported by IEEE or ACM. On the other hand, the exclusion criteria were (i) articles not published within the period 2012–2022, (ii) papers not published in a journal or conference, and (iii) papers with not enough relevance to the main topic of this chapter.

4.1 Healthcare

Healthcare systems produce the largest and fastest growing datasets corresponding mainly to electronic medical records (EMRs) and imaging data, which are considered clinical data [45]. Other types of healthcare-related data are patient behavior and sentiment data such as those coming from wearable sensors and social sites;

administration and cost activity data, such as financial and operational data, and patient profiles including dietary habits, exercise patterns, and environmental factors; and pharmaceutical and research and development data, including mechanism of action of drugs, and their side effects and toxicity [46].

Collected patient information is growing both in volume and complexity. For instance, neuroimaging currently produces more than 10 petabytes (10^{15}) of data each year, and genomic sequencing data is expected to reach exabyte (10^{18}) proportions per year within the next decade, exceeding other big data fields such as astronomy [47]. Given that healthcare is a data-intensive field and that health data comes from numerous sources and in different formats, traditional software systems are not able to handle this kind of data [34]. It is therefore justified to use the tools provided by BDA to collect, organize, analyze, and evaluate massive datasets from healthcare systems in order to identify patterns and other information of interest that can lead as an ultimate goal to improve human welfare [48].

Health BDA has mainly four challenges:

1. Data aggregation, as health big data come from different sources, it has to be put together from warehouses located in different places and in real time.
2. Data maintenance and storage, which require both SQL and NoSQL databases systems, as the data are growing at an exponential rate and come in different formats.
3. Data integration and interoperability, as data come in structured, semi-structured, and unstructured formats, and a way has to be found to standardize all these data so that systems can operate together.
4. Data analysis, as the time and resource requirements increase exponentially as the number of records increases, the hardware and software needed to analyze health data have to grow in size and complexity to provide robust analytical tools to perform analyses that extract knowledge from the data [34].

The three types of analytics are of interest in healthcare: descriptive, predictive, and prescriptive [46]. In a 2020 review of 804 articles that applied BDA to healthcare data, almost half of the articles used predictive analytics, approximately a third used prescriptive analytics, and nearly a quarter used descriptive analytics [46]. These results emphasize the fact that in healthcare, predicting outcomes is more valuable than building an explanatory model, as delaying action waiting for a complete model can cost lives [49]. In this same review, 70% of studies used clinical data, many articles (40%) included experiments with the hope that the proposed predictive and prescriptive models be incorporated in systems used by decision-makers in healthcare organizations, and nearly 65% of the articles focused on ML and data mining techniques applied to the field of health, such as the classification of medical data and symptoms and diagnosis and prediction of diseases [46]. In general, ML and statistical methods such as data mining are among the main approaches used in predictive analysis in order to make informed decisions on patient care by examining current and historical facts to predict future outcomes [50].

Machine learning techniques can be valuable for the prediction of disease occurrences or their complications. Although many ML algorithms can be applied to

solve health related problems, each type of problem might best be solved using a particular technique or a certain combination of techniques. For instance, deep learning has been successfully applied to the classification of medical images and videos, frequently in combinations with the processing of EMRs [47]. In healthcare, the following ML algorithms have been used on big data [34, 48]: *K*-nearest neighbor, support vector machines, neural networks, *k*-means clustering techniques, ensemble learning, Markov decision process, decision trees, and naïve Bayes.

Regarding the platforms, the following big data platforms are popular in health informatics: Hadoop, Spark, High Performance Computing (HPC) cluster, Flink, and Storm [34]. The Hadoop ecosystem has been used in the following applications: treatment of cancer and genomics; monitoring of patient vitals; collection of real-time data related to patient care; processing of large datasets related to drugs, diseases, symptoms, and other factors to extract meaningful information for insurance companies; and prevention and detection of frauds [50].

The selection of the big data platform as solution for a specific healthcare problem depends on a number of factors, such as real-time requirements, speed, data size, scalability, and throughput, among others. Some applications, such as EMR collection, might not require real-time processing, and a platform that does not require live streaming such as Hadoop MapReduce will suffice, but for other applications, a real-time response will be a must, such as the analysis of an electrocardiogram in order to determine a possible intervention. For other applications such as diagnosis suggestion support, scalability and storage of huge amounts of data is a necessity, in which case a scaling system like Spark would be the right choice [34].

Issues and future directions concerning big data in healthcare involve the increased volumes of health data at an intense rate, which demand an increment in IT infrastructure to allow healthcare organizations and researchers to safely manage and exploit the ever-increasing quantities of datasets and enable clinical decision-making in real time based on personalized data from patients [46]. Another concern in healthcare is the high heterogeneity of data sources, the noise introduced in high-throughput experiments, and the variety of experimental techniques and environmental conditions; these heterogeneous data must frequently be collected and preprocessed before applying the data mining methods to extract valuable knowledge. Big data privacy and security of healthcare data are also two important issues that must be addressed in BDA software, by, for example, using advanced encryption algorithms and pseudo-anonymization of the personal data; these software solutions must offer security on the network level and authentication of all users handling these data, as well as appropriate governance standards and practices [51]. Given the sensitive nature of healthcare data, attempts to protect medical and clinical data have been provided by legal provisions such as the Health Insurance Portability and Accountability Act (HIPAA) in the USA, which safeguards the collection, storage, and disclosure of identifiable healthcare data. However, this protection is provided only for so-called covered entities, such as insurance companies and healthcare facilities, but does not cover firms that own social networks such as Facebook, Google, and Twitter, which in some cases have been known to make illegal use of personal information from users. Data protection laws should be

extended beyond healthcare settings and encompass systems—such as social network services—that allow the amassment, storage, and analysis of personal information [52].

Regarding the application of ML techniques on big data in various fields within healthcare, some examples are shown in Table 1.

The sample applications from Table 1 were chosen to cover different healthcare fields from a number of datasets from patients from various parts of the world. A more detailed description of the examples given in Table 1 follows.

Gulshan et al. [53] used a deep convolutional neural network for the detection of diabetic retinopathy and macular edema in US patients. The CNN was trained using a dataset of 128,175 retinal images that were classified in a scale of 3 to 7 for diabetic retinopathy and macular edema by a set of 54 US ophthalmologists. The trained neural network was validated using two separate datasets of 9963 and 1748 images. At an operating point selected for high sensitivity for the detection of diabetic retinopathy and diabetic macular edema, the algorithm had a sensitivity of 97.5% and 96.1% and a specificity of 93.4% and 93.9% for the two respective

Table 1 Some applications of machine learning algorithms on big data in healthcare

Ref.	Desired goal	Platform and/or tools	Machine learning algorithm	Datasets	Issues
[53]	Detection of diabetic retinopathy and diabetic macular edema in US patients	An implementation of the Inception-v3 architecture	Deep convolutional neural network	128,175 retinal images	Research needed on applicability in clinical settings
[54]	Prediction of diabetes on Indian populations	Hadoop, R	Decision tree, naïve Bayes, random forest	Data from 75,664 patients with 13 attributes	A higher number of Hadoop nodes could be useful
[55]	Prediction of multimodal cerebral infarction risk in Chinese populations	Not mentioned	Convolutional neural networks	20,320,848 records from patients containing structured and unstructured text data	The accuracy of the algorithms depends on the feature description of the disease
[56]	Prediction of obesity in children in the USA	WEKA	Decision tree algorithm ID3	Data from 7519 patients	Clinical data can have missing or erroneous values
[57]	Disease detection in populations from Saudi Arabia	Spark	Naïve Bayes, logistic regression	18.9 million tweets	Privacy risks concerning public data need to be addressed

validation datasets. The authors state that the feasibility of using the algorithm in a clinical setting for the detection of these diseases requires further research.

In another study, Yuvaraj and SriPreethaa [54] compared three ML algorithms for their ability to predict diabetes using data from Indian populations. A dataset from 75,664 patients obtained from the Indian National Institute of Diabetes was used, with each record having 13 attributes related to diabetes. From this dataset, 70% of the data was used for training the algorithm, and the remaining 30% was used for validation of the model. The following ML algorithms were compared against each other in terms of precision, recall, F-measure, and accuracy on a Hadoop cluster with four nodes running R language scripts: decision tree, naïve Bayes, and random forest. Under the conditions tested, the random forest algorithm yielded a better precision for predicting diabetes by at least 3% than the other two algorithms for all evaluation measurements. The authors propose to use a Hadoop cluster with more nodes to speed up the process and to compare other ML algorithms.

Chen et al. [55] used a convolutional neural network algorithm to predict the risk of cerebral infarction using data from 31,919 hospitalized patients in Central China from the years 2013 to 2015. The data consisted of 20,320,848 records in total and was composed of structured and unstructured data. The structured data included laboratory data and the basic information from the patient, such as age, gender, and life habits, whereas the unstructured text data included the patients' narration of their illness, as well as the doctors' notes on the case. A CNN-based multimodal (using both structured and unstructured data) disease risk prediction algorithm was designed based on a unimodal (using only unstructured text data) CNN prediction algorithm. The multimodal disease risk prediction algorithm achieved 94.8% accuracy and a faster convergence speed than the unimodal disease risk prediction algorithm. The authors found out that the accuracy of the algorithms depended on the quality of the descriptions of the diseases in the data available.

Dugan et al. [56] compared six ML algorithms to predict obesity in children from the USA after the age of 2 using only data collected before this age. The ML techniques analyzed were the WEKA implementations of the random tree, random forest, J48, ID3, naïve Bayes, and Bayes algorithms. The data was collected from a US pediatric clinical support system and consisted of records from 7519 patients. Results showed that the decision tree algorithm ID3 accurately predicted obesity in children after the age of 2. These authors emphasized that clinical data might have missing or erroneous values that can affect the accuracy of the prediction.

In another study, Alotaibi et al. [57] developed a symptoms and disease detection tool using Twitter data in Arabic and proposed its use by the healthcare system in the Kingdom of Saudi Arabia. The data consisted of 18.9 million tweets collected from November 2018 to September 2019. The proposed tool implemented the naïve Bayes and the logistic regression algorithms and ran on a Spark platform. The tool detected that the top 5 diseases in Saudi Arabia according to the available Twitter data were dermal diseases, heart diseases, hypertension, cancer, and diabetes. The results were evaluated using numerical criteria (Accuracy and F1-score) and validated against available healthcare statistics. The data obtained by the proposed system could be used by healthcare officials, among other things, to create awareness

in the public about the top diseases and how to prevent them. On the other hand, the availability of healthcare data in public social networks raises privacy concerns that need to be addressed.

From these examples, we can see that the main focus of the analysis of big data using ML techniques lies on the detection of present diseases or the prediction of future diseases. Another commonality is that these studies consist of proposals to be used in clinical settings, rather than descriptions of working systems currently in use in healthcare facilities. Furthermore, distributed platforms such as Hadoop and Spark in these reports are not as widely used as they should in order to process the large amounts of data required by BDA systems. The above suggests that the use of ML in BDA is still mainly in an exploratory phase before its adoption in real-world applications in the healthcare field. On the other hand, although the ML algorithm used depends largely on the kind of application desired in healthcare, it is noticeable that deep learning algorithms are steadily being used more frequently in BDA in these and other works, instead of the more traditional ML algorithms. Finally, a concern that is emphasized is the matter of privacy of healthcare data, since the records and other clinical data from patients frequently require to be processed in a different location from the one where it was produced and can also require to be accessed by different people in the BDA systems.

In general, it can be concluded that although there is still room for improvements in a number of aspects, ML techniques will be indispensable tools in the extraction of knowledge from big data derived from healthcare systems in order to improve the well-being of humans at the individual and the population level.

4.2 Weather Forecasting

Weather forecasting has gained attention in the last decades due to its potential to save lives. For instance, forecasting hurricanes, cyclones, heavy rains, and tornados can help in implementing evacuation plans more efficiently. Weather forecasting is also important in agriculture as it allows farmers to prepare their lands for any anticipated weather changes. Furthermore, social events and sport events can be organized based on weather predictions.

Currently, weather forecasting primarily relies on model-based methods, in which the atmosphere is modeled as a fluid. Partial differential equations of fluid dynamics and thermodynamics [58] are solved using numerical methods. Sample measurements of the current state of the atmosphere are taken in order to approximate the future states by solving such equations. Solving these equations can be computationally expensive depending on the size and granularity of the modeled area. There are different numerical weather prediction models. The Weather Research and Forecasting (WRF) model [59, 60] is currently the world's most used model mainly due to its open-source nature as well as its higher resolution and accuracy. WRF was developed in the 1990s and it was openly released in the year 2000 [60].

Data-driven computer modeling systems, including BDA, can be used as an alternative to numerical weather prediction methods. One of the advantages of the data-driven approaches is obtaining a higher accuracy for short-term forecasts [61]. Several ML approaches have been applied to weather forecasting. Below we present approaches that employ ANN.

An ensemble of neural networks is proposed by Ahmadi et al. [62] for weather prediction. The authors' approach outperformed other similar approaches. One of the main disadvantages of this solution is that the ensemble creates a redundancy. Patil et al. [63] used neural networks to forecast sea surface temperature, whereas Rodríguez-Fernández et al. [64] applied neural networks to predict soil moisture. On the other hand, Sharaff and Roy [65] presented a comparative analysis of regression methods and the back propagation neural network for temperature forecasting. The authors concluded that the back propagation network achieves better accuracy than linear regression and regression trees.

One of the first attempts to employ deep ANNs to the domain area of weather forecasting was carried out by Liu [66], which presented a deep neural network-based feature representation for weather forecasting. The results showed that deep ANNs achieved a higher accuracy than traditional methods such as support vector regression (SVR). Also, a deep neural network was used for ultrashort-term wind speed prediction by Dalto et al. [67]. The authors' results show that deep neural networks outperformed shallow neural networks. In addition, Shi et al. [68] presented a deep learning approach with long short-term memory (LSTM) for precipitation nowcasting. The authors' approach uses a convolutional long short-term memory (LSTM) prediction of rain intensity over local areas. The accuracy of the fully connected LSTM approach is overtaken by the convolutional LSTM. Moreover, Hossain et al. [69] showed that their deep learning approach was able to obtain a higher accuracy than traditional ANNs for predicting temperature. Besides, Yonekura et al. [70] employed a deep learning neural network to predict short-term local temperature and rain. The deep learning approach obtained a higher accuracy than other ML methods.

Apart from ANNs, some other ML models have been used. Voyant et al. [71] presented a comparison of different traditional ML algorithms for radiation forecasting. The authors concluded that ANN and ARIMA are equivalent in terms of accuracy and that SVR, random forests, and regression trees obtained promising results. In addition, Rasel et al. [72] showed that SVR outperformed ANNs in rainfall prediction. However, ANNs obtained better results than SVR for temperature forecast. Mahmood et al. [73] employed a cumulative distribution function for the prediction of extreme weather changes. Moreover, Zhan et al. [74] carried out a correlation analysis of both meteorological and hydrological data whereby a correlation matrix is obtained. The authors then used an SVR model for horizontal comparison in order to obtain a higher accuracy. A random forest model was used for the same purpose; however, the SVR model obtained better results. Lastly, Maliyeckel et al. [75] proposed a hybrid ML model for rainfall prediction. The authors employed algorithms of the LightGBM framework together with an SVR model. The former is a gradient booting framework that uses tree-based learning algorithms. The authors reported that the hybrid model obtained better results than each of the individual models.

The algorithm that is currently more widely used for weather forecasting is an artificial neural network. Recently, deep learning networks have received special attention in the area of weather forecasting. It has been shown that deep learning networks achieve better accuracy than traditional ML methods. In particular, deep networks are able to model complex data with fewer elements than shallow networks. The reason is that the extra layers enable the composition of features from lower layers. One disadvantage of deep networks is that a larger computation time is required for training. Other algorithms that have been successfully employed are SVR, decision trees, and random forest.

Regarding distributed platforms, Hadoop and Spark are the most widely used systems for processing big data related to weather forecasting [76]. In addition, some of the most used languages to develop ML algorithms in the area of weather forecasting include Python and MATLAB [76].

There are a number of issues that need to be addressed regarding the use of BDA for weather forecasting. First of all, most of the works mentioned above do not use a distributed computing model such as MapReduce to manage large amounts of data. Rather, most of these works focus on developing and evaluating different ML algorithms for weather forecasting but miss to evaluate the scalability of their approaches. As a consequence, many proposals report good accuracy for short-term weather predictions. However, further research is needed to evaluate the accuracy of the ML models for larger-term forecasts where a larger amount of data is required. Another issue that requires further attention is that most works do not use a development process methodology for implementing BDA in the area of weather forecasting, giving place to ad hoc practices that can make this task far more complicated.

In Table 2, we selected a sample of works that cover different aspects of the weather forecasting domain. More concretely, we selected works aiming to forecast different aspects of weather such as temperature, rainfall, thunderstorms, wind speed, and severe convective weather.

We present next a more detailed description of the applications shown in Table 2. Hewage et al. [61] used two variants of recurrent neural networks (RNN) called long short-term memory (LSTM) and temporal convolutional networks (TCNs) for weather forecasting. The authors developed a multi-input multi-output (MIMO) model and a multi-input single-output (MISO) model. The former is fed with ten surface parameters (i.e., surface temperature, surface pressure, X component of wind, Y component of wind, humidity, convective rain, non-convective rain, snow water equivalent, soil temperature, and soil moisture) and predicts the same parameters; thus, only one model is needed to predict all the parameters. The latter is fed with ten surface parameters and predicts a single parameter; hence, ten models are required for predicting all the parameters. The authors employed 675,924 records to develop the models. Also, the Keras tool (a Python library) was employed for developing and evaluating the models. The LSTM and TCN models outperformed classic ML approaches such as standard regression, SVR, and random forest. The proposed models also produced better prediction results than WRF in the case of short-term forecasting. However, WRF produces better forecasting results in the case of long-term forecasting.

Table 2 Some applications of machine learning algorithms on big data in the weather forecasting domain

Ref.	Desired goal	Platforms and/or tools	Machine learning algorithm	Datasets	Issues
[61]	Use deep learning for weather forecasting. Proposed model outperforms traditional methods such as regression, SVM, and random forest. Also, better results than WRF for short-term forecasting	Python	Deep learning network with long short-term memory	675,924 records	The approach has not been tested for long-term forecasting
[77]	Deep learning approach to predict severe convective weather	Python	A convolutional network model	4,582,577 thunderstorm samples; 3,609,185 heavy rain samples; 1,468,158 hail samples	Deep CNN training time is much longer than simpler algorithms. There are still inadequacies in the proposed algorithm as it issues too many false alarms of hail
[78]	Deep learning approach to predict severe convective weather such as heavy rain and thunderstorms	Not mentioned	Deep convolutional neural network	Temperature prediction data from 2009 to 2015; wind prediction data from 2000 to 2010	The approach has only been tested for short-term forecasting
[79]	A regression tree model for very short-term wind speed prediction	R	Regression tree	3061 hourly samples of wind speed	Only supports very short-term predictions
[80]	An SVR model to forecast rainfall of landslides	Spark	SVR	The data was taken from September 15, 2016, to February 28, 2017	The study considered only a small dataset involving 4008 records

The work of Zhou et al. [77] proposes a deep learning approach for severe convective weather involving heavy rain, hail, and thunderstorms. The authors employed 5 years of severe weather observation involving 4,582,577 thunderstorm samples, 3,609,185 heavy rain samples, and 1,468,158 hail samples. The results of this work show that the six-layer convolutional neural network obtained better results than SVR, random forest, and other traditional ML approaches. The proposed deep learning model is currently used in the National Meteorological Center of China to provide guidance on the operational forecast of severe convective weather events in China. Unfortunately, although the authors employ big data to develop their models, their work does not report on the computing approach taken to deal with big data.

Mehrkanoon [78] proposed a convolutional neural network to predict temperature and wind speed, both involving short-term forecasts. The author shows that the two-layer and three-layer networks outperform shallow networks. The datasets employed for developing the models include data from 2009 to 2015 for temperature prediction, whereas data from 2000 to 2010 was used for wind speed prediction. The authors used large amounts of data to develop their models; nevertheless, their work does not mention what tools and platforms were employed.

Troncoso et al. [79] evaluated the accuracy of different types of regression tree models employed in very short-term forecasts of wind speed. The authors also show that regression trees are able to outperform—for this specific problem—other ML approaches such as SVR and neural networks. The package CORElearn (a library of R) was used to generate the models. The authors used a sample of 3061 samples of hourly wind speed measures taken by eight towers.

Lee et al. [80] proposed an SVR model to forecast rainfall of landslides on the Apache Spark platform. This platform was configured in the standalone mode in which the worker node employed the SVR model. The model was developed with data taken from September 15, 2016, to February 28, 2017.

We can see that these works have aimed at forecasting different variables of the weather. For example, Troncoso et al. focus on forecasting wind speed, whereas Lee et al. target forecasting rainfall of landslides. Other works focus on severe convective weather, such as the work by Zhou et al. and the work by Mehrkanoon. Other efforts have taken a more holistic approach in which multiple variables are predicted, such as the case of the work by Hewage et al. On the other hand, the most popular tools employed are Python, R, and Apache Spark. Crucially, most of the reviewed works do not pay attention to the issue of efficiently processing big data; rather, the authors focus on showing which ML algorithm is more accurate. In fact, apart from the works of Zhou et al. and Hewage et al., most of the reviewed approaches do not include large amounts of data in their experiments. Therefore, further work is still required to investigate the accuracy of the proposed ML methods in the case of large datasets and long-term forecasting.

4.3 Social Networks and the Internet

Social networking and the Internet handle a large amount of passive data. These data involve user information, historical data, comments, interaction, blogs, etc. from websites and social media networks. Some examples of websites and social media networks are: Twitter Inc.'s microblogging site twitter.com, Google Inc.'s video platform youtube.com, Meta Corp.'s instagram.com, facebook.com, the associated Meta WhatsApp messaging service, and devices apps. In other words, these websites and social media networks catch information flows from web-based life or applications that predict end-user behavior patterns. The main goal of ML is to enable data-driven decision-making. This decision must be accurately based on analyzed data. However, these data should have privacy, security, accuracy, and confidentiality for this domain, as the increase of everyday data generated by humans is 2.5 quintillion bytes [37].

Some properties and data types, from social networking and the Internet, are time, GPS coordinates, user ID, texts, videos, velocity, address, posts, SMS, and IP social media, among others [89] that need to be analyzed by ML algorithms in this domain. A wide variety of unstructured data is produced mainly from email conversations and social networking sites as graphics and text [19]. Data evolve rapidly in a highly connected society, which is generated by data sources such as social media, mobile devices, and the Internet of Things (IoT) [81].

There are some successive phases to manage organization data processes such as data generation, data acquisition, data preprocessing, data storage, data analysis, data visualization, and data exposition, which are defined in [81]. In data generation, the data is generated from different sources (e.g., IoT, social media, operational and commercial data); therefore, data acquisition has three subphases: data identification, data collection, and data transfer. In data analysis, ML models are applied to predict future events and drive proactive decisions. The most common ML algorithms are clustering, graph analysis, decision trees, classification, and regression and association analysis in ML analysis [81].

Table 3 shows some applications of ML algorithms on big data obtained from the social networks and the Internet domain.

Some of the current ML algorithms for social networking and the Internet are found in the state-of-the-art literature. For instance, Nti et al. [4] studied the applications of the decision tree, neural network, and support vector machine algorithms, and the platforms that they used were Hadoop, MapReduce, and Spark, with SQL (structured query language) as language. The authors' aim was to make data-driven decisions to accomplish the desired goals. On the other hand, Kaur and Lal [19] used k-means and hierarchical clustering algorithms, along with the SparkR platform and the R language; their main aim was to improve clustering and reduce CPU utilization through ML. In another work proposed by Latif and Afzal [82], logistic regression, simple logistic multilayer, perceptron J48, and naive Bayes PART implemented with WEKA and Java were used; they concluded that efficient models could predict a movie's popularity for social networking. Lakshmanaprabu et al. [83] used

Table 3 Some applications of machine learning algorithms on big data in the social networks and the Internet domain

Ref.	Desired goal	Platform and/or tools	Machine learning algorithm	Datasets	Issues
[4]	Proposal of a taxonomy with a keyword search and using the appropriate tool or platform for the right task	Hadoop, MapReduce, Spark	Decision trees, neural networks, and support vector machines	Data from 1512 published articles	The high number of free BDA tools, platforms, and data mining tools makes it challenging to select the appropriate one for the right task
[19]	Improvement of clustering and reduction of CPU utilization through ML	SparkR	*K*-means, hierarchical clustering	622 datasets	There is a need to use these algorithms for heterogeneous data such as image, video, streams, etc.
[82]	Construction of efficient models that can predict the popularity of a movie	WEKA	Logistic regression, simple logistic multilayer perceptron J48, and naive Bayes PART	2000 data points	It is necessary to select specific attributes related to a movie
[83]	Reduction of noise and unwanted data from the database to improve the efficiency of their algorithm	Hadoop, MapReduce	Linear kernel, SVM	34,042 data instances	Large amounts of data are required to analyze social networking systems involving Internet of Things
[84]	Reduction of redundant and irrelevant datasets	Not mentioned	Random forest, K-means, and support vector machines	Approximately 33,000 attack accounts, NSL-KDD	It is necessary to monitor the network and analyze the incoming traffic dynamically

a linear kernel support vector, Hadoop MapReduce, and Java to reduce noise and unwanted data from a database to improve the efficiency of their algorithm. Finally, Patgiri et al. [84] used random forest, support vector machine (SVM), and NSL-KDD to reduce redundant and irrelevant datasets.

Considering other works, in [85] the authors used classification, regression, dimensionality reduction, clustering, and density estimation to classify the good, the bad, and the ugly use for cybersecurity and cyber physical systems. On the other hand, the authors in [86] analyzed big data for social transportation. The authors concluded that social data contain abundant information and evolve with time. The authors in [87] developed a model for fake news detection using SVM and NB. Other approaches such as [88] focused on traffic management. The authors used online

learning, which was handled by an online adaptive clustering algorithm and incremental learning. Incremental learning is based on the incremental knowledge acquisition and self-learning (IKASL) algorithm, decremental learning, and concept drift detection. Finally, the authors in [19] used *k*-means and hierarchical clustering algorithms for analyzing social networking using SparkR. The authors' models were fed with social media involving YouTube datasets.

The algorithm more commonly used for social networking is SVM, which is internally deployed for tracking and classifying key metrics—e.g., likes, loyalty, and value information. Other algorithms that have been employed for processing social media data are naive Bayes, decision trees, and the clustering algorithm *k*-means. Therefore, the selection of the most appropriate algorithm for the social network and Internet domain depends on the goal of the application; for instance, SVM is a supervised algorithm, whereas *k*-means is an unsupervised algorithm, and both have been used for this domain. Furthermore, from the previous examples, it can be seen that authors frequently used a combination of two or more ML algorithms to achieve a higher performance.

Some of the challenges in this application domain are the large number of free BDA tools, platforms, and data mining tools that are available, making it difficult to select the appropriate one for the right task. Another challenge is the large diversity of heterogeneous data formats that exist for social media, such as image, video, and text, among others.

5 Conclusions

Given the huge amounts of data that are currently produced in practically all domains of human knowledge and social interaction and that this quantity of data will most likely continue to increase exponentially in the foreseeable future, the use of automated computational tools to extract meaningful information from these data is no longer an option but a necessity. Big data analytics in conjunction with machine learning algorithms are poised to fill this need, as machine learning techniques have been developed precisely to computationally automate the extraction of knowledge from data.

Concerning the domain areas mentioned in the previous section, and the three kinds of big data analytics (descriptive, predictive, and prescriptive), healthcare and weather forecasting benefit especially from predictive analytics. That is, in general it is more important, for instance, to predict the appearance of new disease outbreaks, or to predict bad weather conditions, than it is to explain previous occurrences of events.

As for the machine learning techniques currently in use in big data analytics, practically all kinds of algorithms can be applied depending on the specific goals desired. However, deep learning algorithms have proven to still have room for improvements and for being applied in other application domains in big data analytics systems.

Regarding the open-software platforms currently used, both Apache Hadoop and Apache Spark continue to be the preferred choice for big data analytics systems, with a preference for Spark when speed and real-time processing are needed, and a predilection for Hadoop when processing data in batches and a speedy response is not an issue.

In the literature review that we made, we found that many researchers used big data analytics at a small scale only to demonstrate the feasibility of a machine learning approach to solve a problem, but without the application to actual big data for solving real-world problems. Furthermore, many reports concentrated on achieving high accuracy on their proposals for integrating machine learning with big data analytics systems, but without concern for building computationally efficient systems. We also found that in the reviewed literature, the use of distributed systems using Hadoop or Spark is not as widespread as it should be in big data analytics systems. Thus, we consider that research and applications of big data analytics in conjunction with machine learning will continue to grow in the years to come, both in academia and industry.

References

1. Reinsel, D., Gantz J., Rydning, J.: The Digitalization of The World: From Edge to Core (2018), https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf
2. Rahman, M.S., Reza, H.: A systematic review towards Big Data analytics in social media. Big Data Min. Anal. **5**, 228–244 (2022). https://doi.org/10.26599/BDMA.2022.9020009
3. Fisher, D., DeLine, R., Czerwinski, M., Drucker, S.: Interactions with Big Data analytics. Interactions. **19**, 50–59 (2012). https://doi.org/10.1145/2168931.2168943
4. Nti, I.K., Quarcoo, J.A., Aning, J., Fosu, G.K.: A mini-review of machine learning in big data analytics: applications, challenges, and prospects. Big Data Min. Anal. **5**, 81–97 (2022). https://doi.org/10.26599/BDMA.2021.9020028
5. Wixom, B., Ariyachandra, T., Douglas, D., Goul, K., Gupta, B., Iyer, L., Kulkarni, U., Mooney, B.J.G., Phillips-Wren, G., Turetken, O.: The current state of business intelligence in academia: the arrival of big data. Commun. Assoc. Inf. Syst. **34**, 1–13 (2014). https://doi.org/10.17705/1cais.03401
6. Laney, D.: 3D data management: Controlling data volume velocity and variety, https://studylib.net/doc/8647594/3d-data-management%2D%2Dcontrolling-data-volume%2D%2Dvelocity%2D%2Dan... (2001)
7. Qiu, J., Wu, Q., Ding, G., Xu, Y., Feng, S.: A survey of machine learning for big data processing. EURASIP J. Adv. Signal Proc. **2016**, 1–16 (2016)
8. EMC (ed.): Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. Wiley Publishing (2015)
9. Grover, P., Kar, A.K.: Big Data analytics: a review on theoretical contributions and tools used in literature. Global J. Flex. Syst. Manag. **18**, 203–229 (2017). https://doi.org/10.1007/s40171-017-0159-3
10. Mikalef, P., Pappas, I.O., Krogstie, J., Giannakos, M.: Big data analytics capabilities: a systematic literature review and research agenda. Inf. Syst. E-Bus. Manag. **16**, 547–578 (2018). https://doi.org/10.1007/s10257-017-0362-y
11. Zhou, L., Pan, S., Wang, J., Vasilakos, A.V.: Machine learning on big data: opportunities and challenges. Neurocomputing. **237**, 350–361 (2017). https://doi.org/10.1016/j.neucom.2017.01.026

12. Russell, S., Norvig, P.: Artificial Intelligence: a Modern Approach. Prentice Hall (2010)
13. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press (2000)
14. Sun, Z.Q., Fox, G.C.: Study on parallel SVM based on MapReduce. In: International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 495–561, Las Vegas, NV, USA (2012)
15. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Taylor & Francis (1984)
16. Dai, W., Ji, W.-Z.: A MapReduce implementation of C4.5 Decision Tree algorithm. Int. J. Database Theory Appl. **7**, 49–60 (2014)
17. Purdilă, V., Pentiuc, Ş.-G.: MR-Tree-A scalable MapReduce algorithm for building decision trees. J. Appl. Comput. Sci. Math. **8**, 16–19 (2014)
18. Mahdavinejad, M.S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., Sheth, A.P.: Machine learning for internet of things data analysis: a survey. Digit. Commun. Netw. **4**, 161–175 (2018). https://doi.org/10.1016/j.dcan.2017.10.002
19. Kaur, N., Lal, N.: Clustering of social networking data using SparkR in Big Data. In: Mayank, S., Gupta, P.K., T.V, F.J, Ö.T (eds.) Advances in Computing and Data Sciences, pp. 217–226. Springer Singapore, Singapore (2018)
20. Arora, P., Deepali, Varshney, S.: Analysis of K-means and K-Medoids algorithm for Big Data. In: International Conference on Information Security & Privacy (ICISP2015), pp. 507–512 (2016)
21. Prabhu, C.S.R., Chivukula, A.S., Mogadala, A., Ghosh, R., Livingston, L.M.J.: Big Data Analytics: Systems, Algorithms, Applications. Springer, Singapore (2019)
22. Ray, S.: A quick review of Machine Learning algorithms. In: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 35–39 (2019)
23. Yuan, C., Yang, H.: Research on K-value selection method of K-means clustering algorithm. J (Basel). **2**, 226–235 (2019). https://doi.org/10.3390/j2020016
24. Narayanan, B.N., Djaneye-Boundjou, O., Kebede, T.M.: Performance analysis of machine learning and pattern recognition algorithms for Malware classification. In: 2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS), pp. 338–342 (2016)
25. Narayanan, B.N., Hardie, R.C., Kebede, T.M.: Performance analysis of a computer-aided detection system for lung nodules in CT at different slice thicknesses. J. Med. Imag. **5**, 14504 (2018). https://doi.org/10.1117/1.JMI.5.1.014504
26. Narayanan, B.N., Hardie, R.C., Kebede, T.M., Sprague, M.J.: Optimized feature selection-based clustering approach for computer-aided detection of lung nodules in different modalities. Pattern Anal. Appl. **22**, 559–571 (2019). https://doi.org/10.1007/s10044-017-0653-4
27. Al-Yaseen, W.L., Othman, Z.A., Nazri, M.Z.A.: Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Syst. Appl. **67**, 296–303 (2017). https://doi.org/10.1016/j.eswa.2016.09.041
28. Ge, Y., Tang, K.: Distributed dynamic cluster algorithm for wireless sensor networks. In: 6th International Conference on Wireless, Mobile and Multi-Media (ICWMMN 2015), pp. 23–27 (2015)
29. Ran, X., Zhou, X., Lei, M., Tepsan, W., Deng, W.: A novel K-means clustering algorithm with a noise algorithm for capturing urban hotspots. Appl. Sci. (Switzerland). **11** (2021). https://doi.org/10.3390/app112311202
30. Bendechache, M., Kechadi, M.-T.: Distributed clustering algorithm for spatial data mining. In: 2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM), pp. 60–65 (2015)
31. Chiroma, H., Abdullahi, U.A., Abdulhamid, S.M., Abdulsalam Alarood, A., Gabralla, L.A., Rana, N., Shuib, L., Targio Hashem, I.A., Gbenga, D.E., Abubakar, A.I., Zeki, A.M., Herawan, T.: Progress on artificial neural networks for Big Data analytics: a survey. IEEE Access. **7**, 70535–70551 (2019). https://doi.org/10.1109/ACCESS.2018.2880694

32. Shen, D., Wu, G., Suk, H.-I.: Deep learning in medical image analysis. Annu. Rev. Biomed. Eng. **19**, 221–248 (2017). https://doi.org/10.1146/annurev-bioeng-071516-044442
33. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM. **51**, 107–113 (2008). https://doi.org/10.1145/1327452.1327492
34. Harerimana, G., Jang, B., Kim, J.W., Park, H.K.: Health Big Data analytics: a technology survey. IEEE Access. **6**, 65661–65678 (2018). https://doi.org/10.1109/ACCESS.2018.2878254
35. Apache Software Foundation: Apache Hadoop, https://hadoop.apache.org/
36. Ketu, S., Mishra, P.K., Agarwal, S.: Performance analysis of distributed computing frameworks for Big Data analytics: Hadoop vs Spark. Computación y Sistemas. **24**, 669–686 (2020). https://doi.org/10.13053/CyS-24-2-3401
37. Mohd, A.B., Banu, A., Yakub, M.: Evolution of big data and tools for big data analytics. J. Interdiscipl. Cycle Res. **12**, 309–316 (2020)
38. Gupta, P., Sharma, A., Jindal, R.: Scalable machine-learning algorithms for big data analytics: a comprehensive review. WIREs Data Min. Knowl. Discov. **6**, 194–214 (2016). https://doi.org/10.1002/widm.1194
39. Raza, M.U., XuJian, Z.: A comprehensive overview of BIG DATA technologies: a survey. In: Proceedings of the 5th International Conference on Big Data and Computing, pp. 23–31. Association for Computing Machinery, New York, NY, USA (2020)
40. Venkatram, K., Geetha, M.A.: Review on Big Data & analytics – concepts, philosophy, process and applications. Cybern. Inf. Technol. **17**, 3–27 (2017). https://doi.org/10.1515/cait-2017-0013
41. Ikegwu, A.C., Nweke, H.F., Anikwe, C.V., Alo, U.R., Okonkwo, O.R.: Big data analytics for data-driven industry: a review of data sources, tools, challenges, solutions, and research directions. Cluster Comput. (2022). https://doi.org/10.1007/s10586-022-03568-5
42. Faridoon, A., Imran, M.: Big data storage tools using NoSQL databases and their applications in various domains: a systematic review. Comput. Inf. **40**, 489–521 (2021). https://doi.org/10.31577/cai_2021_3_489
43. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., DATA, M.: Practical machine learning tools and techniques. In: Data Mining (2005)
44. R Core Team: R.: A Language and Environment for Statistical Computing, https://www.R--project.org/ (2022)
45. Kambatla, K., Kollias, G., Kumar, V., Grama, A.: Trends in big data analytics. J. Parallel Distrib. Comput. **74**, 2561–2573 (2014). https://doi.org/10.1016/j.jpdc.2014.01.003
46. Galetsi, P., Katsaliaki, K.: A review of the literature on big data analytics in healthcare. J. Oper. Res. Soc. **71**, 1511–1529 (2020). https://doi.org/10.1080/01605682.2019.1630328
47. Cirillo, D., Valencia, A.: Big data analytics for personalized medicine. Curr. Opin. Biotechnol. **58**, 161–167 (2019). https://doi.org/10.1016/j.copbio.2019.03.004
48. Akundi, S.H., Soujanya, R., Madhuri, P.M.: Big Data analytics in healthcare using Machine Learning algorithms: a comparative study. Int. J. Online Biomed. Eng. (iJOE). **16**, 19–32 (2020). https://doi.org/10.3991/ijoe.v16i13.18609
49. Agarwal, R., Dhar, V.: Editorial—Big Data, data science, and analytics: the opportunity and challenge for IS research. Inf. Syst. Res. **25**, 443–448 (2014). https://doi.org/10.1287/isre.2014.0546
50. Sunil Kumar, M.S.: Big Data analytics for healthcare industry: impact, applications, and tools. Big Data Min. Anal. **2**, 48 (2019). https://doi.org/10.26599/BDMA.2018.9020031
51. Ristevski, B., Chen, M.: Big Data analytics in medicine and healthcare. J. Integr. Bioinform. **15** (2018). https://doi.org/10.1515/jib-2017-0030
52. Gostin, L.O., Halabi, S.F., Wilson, K.: Health data and privacy in the digital era. JAMA. **320**, 233–234 (2018). https://doi.org/10.1001/jama.2018.8374
53. Gulshan, V., Peng, L., Coram, M., Stumpe, M.C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P.C., Mega, J.L., Webster, D.R.: Development and validation of a Deep Learning algorithm for detection of

diabetic retinopathy in retinal fundus photographs. JAMA. **316**, 2402–2410 (2016). https://doi.org/10.1001/jama.2016.17216
54. Yuvaraj, N., SriPreethaa, K.R.: Diabetes prediction in healthcare systems using machine learning algorithms on Hadoop cluster. Cluster Comput. **22**, 1–9 (2019). https://doi.org/10.1007/s10586-017-1532-x
55. Chen, M., Hao, Y., Hwang, K., Wang, L., Wang, L.: Disease prediction by machine learning over big data from healthcare communities. IEEE Access. **5**, 8869–8879 (2017). https://doi.org/10.1109/ACCESS.2017.2694446
56. Dugan, T.M., Mukhopadhyay, S., Carroll, A., Downs, S.: Machine learning techniques for prediction of early childhood obesity. Appl. Clin. Inform. **06**, 506–520 (2015)
57. Alotaibi, S., Mehmood, R., Katib, I., Rana, O., Albeshri, A.: Sehaa: a Big Data analytics tool for healthcare symptoms and diseases detection using Twitter, Apache Spark, and machine learning. Appl. Sci. **10** (2020). https://doi.org/10.3390/app10041398
58. Richardson, L.F., Lynch, P.: Weather Prediction by Numerical Process. Cambridge University Press (2007)
59. NCAR/UCAR.: WRF model users site, http://www2.mmm.ucar.edu/wrf/users/
60. Powers, J.G., Klemp, J.B., Skamarock, W.C., Davis, C.A., Dudhia, J., Gill, D.O., Coen, J.L., Gochis, D.J., Ahmadov, R., Peckham, S.E., Grell, G.A., Michalakes, J., Trahan, S., Benjamin, S.G., Alexander, C.R., Dimego, G.J., Wang, W., Schwartz, C.S., Romine, G.S., Liu, Z., Snyder, C., Chen, F., Barlage, M.J., Yu, W., Duda, M.G.: The weather research and forecasting model: overview, system efforts, and future directions. Bull. Am. Meteorol. Soc. **98**, 1717–1737 (2017). https://doi.org/10.1175/BAMS-D-15-00308.1
61. Hewage, P., Trovati, M., Pereira, E., Behera, A.: Deep learning-based effective fine-grained weather forecasting model. Pattern Anal. Appl. **24**, 343–366 (2021). https://doi.org/10.1007/s10044-020-00898-1
62. Ahmadi, A., Zargaran, Z., Mohebi, A., Taghavi, F.: Hybrid model for weather forecasting using ensemble of neural networks and mutual information. In: 2014 IEEE Geoscience and Remote Sensing Symposium, pp. 3774–3777 (2014)
63. Patil, K., Deo, M.C.: Basin-scale prediction of sea surface temperature with artificial neural networks. In: 2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO), p. 1–5 (2018)
64. Rodriguez-Fernandez, N.-J., de Rosnay, P., Albergel, C., Aires, F.: SMOS Neural Network Soil Moisture Data Assimilation. (2017)
65. Sharaff, A., Roy, S.R.: Comparative analysis of temperature prediction using regression methods and back propagation neural network. In: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 739–742 (2018)
66. Liu, J.N.K., Hu, Y.-X., You, J.J., Chan, P.W.: Deep neural network based feature representation for weather forecasting. In: The 2014 World Congress in Computer Science, Computer Engineering, and Applied Computing (2014)
67. Dalto, M., Matuško, J., Vašak, M.: Deep neural networks for ultra-short-term wind forecasting. In: 2015 IEEE International Conference on Industrial Technology (ICIT), pp. 1657–1663 (2015)
68. Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., Woo, W.: Convolutional LSTM network: a machine learning approach for precipitation Nowcasting. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, pp. 802–810. MIT Press, Cambridge, MA (2015)
69. Hossain, M., Rekabdar, B., Louis, S.J., Dascalu, S.: Forecasting the weather of Nevada: a deep learning approach. In: 2015 International Joint Conference on Neural Networks (IJCNN), p. 1–6 (2015)
70. Yonekura, K., Hattori, H., Suzuki, T.: Short-term local weather forecast using dense weather station by deep neural network. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 1683–1690 (2018)
71. Voyant, C., Notton, G., Kalogirou, S., Nivet, M.-L., Paoli, C., Motte, F., Fouilloy, A.: Machine learning methods for solar radiation forecasting: a review. Renew. Energy. **105**, 569–582 (2017). https://doi.org/10.1016/j.renene.2016.12

72. Rasel, R.I., Sultana, N., Meesad, P.: An application of data mining and machine learning for weather forecasting. In: Meesad, P., Sodsee, S., Unger, H. (eds.) Recent Advances in Information and Communication Technology 2017, pp. 169–178. Springer International Publishing, Cham (2018)
73. Mahmood, M.R., Patra, R.K., Raja, R., Sinha, G.R.: A novel approach for weather prediction using forecasting analysis and data mining techniques. In: Saini, H.S., Singh, R.K., Kumar, G., Rather, G.M., Santhi, K. (eds.) Innovations in Electronics and Communication Engineering, pp. 479–489. Springer Singapore, Singapore (2019)
74. Zhan, Y., Zhang, H., Liu, Y.: Forecast of meteorological and hydrological features based on SVR model. In: 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), pp. 579–583 (2021)
75. Maliyeckel, M.B., Sai, B.C., Naveen, J.: A comparative study of LGBM-SVR hybrid machine learning model for rainfall prediction. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), p. 1–7 (2021)
76. Fathi, M., Haghi Kashani, M., Jameii, S.M., Mahdipour, E.: Big Data analytics in weather forecasting: a systematic review. Arch. Comput. Methods Eng. **29**, 1247–1275 (2022). https://doi.org/10.1007/s11831-021-09616-4
77. Zhou, K., Zheng, Y., Li, B., Dong, W., Zhang, X.: Forecasting different types of convective weather: a deep learning approach. J. Meteorolog. Res. **33**, 797–809 (2019). https://doi.org/10.1007/s13351-019-8162-6
78. Mehrkanoon, S.: Deep shared representation learning for weather elements forecasting. Knowledge-Based Syst. **179**, 120–128 (2019). https://doi.org/10.1016/j.knosys.2019.05.009
79. Troncoso, A., Salcedo-Sanz, S., Casanova-Mateo, C., Riquelme, J.C., Prieto, L.: Local models-based regression trees for very short-term wind speed prediction. Renew. Energy. **81**, 589–598 (2015). https://doi.org/10.1016/j.renene.2015.03.071
80. Lee, Z.-J., Lee, C.-Y., Yuan, X.-J., Chu, K.-C.: Rainfall forecasting of landslides using support vector regression. In: 2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII), pp. 1–3 (2020)
81. Faroukhi, A.Z., Alaoui, I., Gahi, Y., Amine, A.: An adaptable big data value chain framework for end-to-end big data monetization. Big Data Cogn. Comput. **4**, 1–27 (2020). https://doi.org/10.3390/bdcc4040034
82. Latif, M.H., Afzal, H.: Prediction of movies popularity using machine learning techniques. Int. J. Comput. Sci. Netw Secur. **16**, 127–131 (2016)
83. Lakshmanaprabu, S.K., Shankar, K., Khanna, A., Gupta, D., Rodrigues, J.J.P.C., Pinheiro, P.R., de Albuquerque, V.H.C.: Effective features to classify big data using social internet of things. IEEE Access. **6**, 24196–24204 (2018)
84. Patgiri, R., Varshney, U., Akutota, T., Kunde, R.: An investigation on intrusion detection system using machine learning. In: Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018, p. 1684–1691. Institute of Electrical and Electronics Engineers Inc. (2019)
85. Liang, F., Hatcher, W.G., Liao, W., Gao, W., Yu, W.: Machine learning for security and the Internet of Things: the good, the bad, and the ugly. IEEE Access. **7**, 158126–158147 (2019). https://doi.org/10.1109/ACCESS.2019.2948912
86. Zheng, X., Chen, W., Wang, P., Shen, D., Chen, S., Wang, X., Zhang, Q., Yang, L.: Big Data for social transportation. IEEE Trans. Intell. Transp. Syst. **17**, 620–630 (2016). https://doi.org/10.1109/TITS.2015.2480157
87. Jain, A., Shakya, A., Khatter, H., Gupta, A.K.: A smart system for fake news detection using machine learning. In: 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), p. 1–4 (2019)
88. Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., de Silva, D., Alahakoon, D., Pothuhera, D.: Online incremental machine learning platform for Big Data-driven smart traffic management. IEEE Trans. Intell. Transp. Syst. **20**, 4679–4690 (2019). https://doi.org/10.1109/TITS.2019.2924883