IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 23, 2017, accepted April 10, 2017, date of publication April 24, 2017, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2696365

Machine Learning With Big Data: Challenges and Approaches

ALEXANDRA L'HEUREUX[1], (Graduate Student Member, IEEE), KATARINA GROLINGER[1], (Member, IEEE), HANY F. ELYAMANY[1,2], (Member, IEEE), AND MIRIAM A. M. CAPRETZ[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Western University, London, ON N6A 3K7, Canada
[2]Computer Science Department, Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt

Corresponding author: Katarina Grolinger (kgroling@uwo.ca)

This work was partially supported by NSERC CRD at Western University under Grant CRD 477530-14.

ABSTRACT The Big Data revolution promises to transform how we live, work, and think by enabling process optimization, empowering insight discovery and improving decision making. The realization of this grand potential relies on the ability to extract value from such massive data through data analytics; machine learning is at its core because of its ability to learn from data and provide data driven insights, decisions, and predictions. However, traditional machine learning approaches were developed in a different era, and thus are based upon multiple assumptions, such as the data set fitting entirely into memory, what unfortunately no longer holds true in this new context. These broken assumptions, together with the Big Data characteristics, are creating obstacles for the traditional techniques. Consequently, this paper compiles, summarizes, and organizes machine learning challenges with Big Data. In contrast to other research that discusses challenges, this work highlights the cause–effect relationship by organizing challenges according to Big Data Vs or dimensions that instigated the issue: volume, velocity, variety, or veracity. Moreover, emerging machine learning approaches and techniques are discussed in terms of how they are capable of handling the various challenges with the ultimate objective of helping practitioners select appropriate solutions for their use cases. Finally, a matrix relating the challenges and approaches is presented. Through this process, this paper provides a perspective on the domain, identifies research gaps and opportunities, and provides a strong foundation and encouragement for further research in the field of machine learning with Big Data.

INDEX TERMS Big Data, Big Data Vs, data analysis, data analytics, deep learning, distributed computing, machine learning, neural networks.

Table of Acronyms

ACRONYM	DEFINITION
FPGA	Field-Programmable Gate Array
GPU	Graphic Processing Units
IoT	Internet of Things
i.i.d	Independent and Identically Distributed
LLE	Locally Linear Embedding
ML	Machine Learning
MLlib	Machine Learning Library
MLP	Multi-Layer Perceptron
MOA	Massive Online Analysis
PCA	Principal Component Analysis
RAMP	Reduce and Map Provenance
RDD	Resilient Distributed Datasets
SVM	Support Vector Machine
SVR	Support Vector Regression

I. INTRODUCTION

Today, the amount of data is exploding at an unprecedented rate as a result of developments in Web technologies, social media, and mobile and sensing devices. For example, Twitter processes over 70M tweets per day, thereby generating over 8TB daily [1]. ABI Research estimates that by 2020, there will be more than 30 billion connected devices [2]. These Big Data possess tremendous potential in terms of business value in a variety of fields such as health care, biology, transportation, online advertising, energy management, and financial services [3], [4]. However, traditional approaches are struggling when faced with these massive data.

The concept of Big Data is defined by Gartner [5] as high volume, high velocity, and/or high variety data that require new processing paradigms to enable insight discovery, improved decision making, and process optimization. According to this definition, Big Data are not characterized

by specific size metrics, but rather by the fact that traditional approaches are struggling to process them due to their size, velocity or variety. The potential of Big Data is highlighted by their definition; however, realization of this potential depends on improving traditional approaches or developing new ones capable of handling such data.

Because of their potential, Big Data have been referred to as a revolution that will transform how we live, work, and think [6]. The main purpose of this revolution is to make use of large amounts of data to enable knowledge discovery and better decision making [6]. The ability to extract value from Big Data depends on data analytics; Jagadish *et al.* [7] consider analytics to be the core of the Big Data revolution.

Data analytics involves various approaches, technologies, and tools such as those from text analytics, business intelligence, data visualization, and statistical analysis. This paper focusses on machine learning (ML) as a fundamental component of data analytics. The McKinsey Global Institute has stated that ML will be one of the main drivers of the Big Data revolution [8]. The reason for this is its ability to learn from data and provide data driven insights, decisions, and predictions [9]. It is based on statistics and, similarly to statistical analysis, can extract trends from data; however, it does not require the explicit use of statistical proofs. According to the nature of the available data, the two main categories of learning tasks are: *supervised learning* when both inputs and their desired outputs (labels) are known and the system learns to map inputs to outputs and *unsupervised learning* when desired outputs are not known and the system itself discovers the structure within the data. Classification and regression are examples of supervised learning: in classification the outputs take discrete values (class labels) while in regression the outputs are continuous. Examples of classification algorithms are k-nearest neighbour, logistic regression, and Support Vector Machine (SVM) while regression examples include Support Vector Regression (SVR), linear regression, and polynomial regression. Some algorithms such as neural networks can be used for both, classification and regression. Unsupervised learning includes clustering which groups objects based on established similarity criteria; k-means is an example of such algorithm. Predictive analytics relies on machine learning to develop models built using past data in an attempt to predict the future [10]; numerous algorithms including SVR, neural networks, and Naïve Bayes can be used for this purpose.

A common ML presumption is that algorithms can learn better with more data and consequently provide more accurate results [11]. However, massive datasets impose a variety of challenges because traditional algorithms were not designed to meet such requirements. For example, several ML algorithms were designed for smaller datasets, with the assumption that the entire dataset can fit in memory. Another assumption is that the entire dataset is available for processing at the time of training. Big Data break these assumptions, rendering traditional algorithms unusable or greatly impeding their performance.

A number of techniques have been developed to adapt machine learning algorithms to work with large datasets: examples are new processing paradigms such as MapReduce [12] and distributed processing frameworks such as Hadoop [13]. Branches of machine learning including deep and online learning have also been adapted in an effort to overcome the challenges of machine learning with Big Data.

This paper first compiles, summarizes, and organizes machine learning challenges with Big Data. In contrast to other research [7], [11], [14], [15], the focus is on linking the identified challenges with the Big Data V dimensions (volume, velocity, variety, and veracity) to highlight the cause-effect relationship. Next, emerging machine learning approaches are reviewed with the emphasis on how they address the identified challenges. Through this process, this study provides a perspective on the domain and identifies research gaps and opportunities in the area of machine learning with Big Data. Although security and privacy are important considerations from an application perspective, they do not impede the execution of machine learning and are therefore considered to be outside the scope of this paper.

The remainder of this paper is organized as follows: Section II reviews related work, and Section III presents machine learning challenges classified according to the Big Data dimensions. An overview of emerging machine learning approaches with discussion about challenges they address is provided in Section IV. Section V aggregates the findings and identifies future research directions. Finally, Section VI concludes the paper.

II. RELATED WORK

This paper highlights the challenges specific or highly relevant to machine learning in the context of Big Data, associates them with the V dimensions, and then provides an overview of how emerging approaches are responding to them. In the existing literature, some researchers have described general machine learning challenges with Big Data [4], [14], [16], [17] whereas others have discussed them in the context of specific methodologies [14], [18].

Najafabadi *et al.* [14] focused on deep learning, but noted the following general obstacles for machine learning with Big Data: unstructured data formats, fast moving (streaming) data, multi-source data input, noisy and poor-quality data, high dimensionality, scalability of algorithms, imbalanced distribution of input data, unlabelled data, and limited labeled data. Similarly, Sukumar [16] identified three main requirements: designing flexible and highly scalable architectures, understanding statistical data characteristics before applying algorithms; and finally, developing ability to work with larger datasets. Both studies, Najafabadi *et al.* [14] and Sukumar [16] reviewed aspects of machine learning with Big Data; however, they did not attempt to associate each identified challenge with its cause. Moreover, their discussions are on a very high level without presenting related solutions. In contrast, our work includes a thorough discussion of challenges, establishes their relations with Big Data

dimensions, and presents an overview of solutions that mitigate them.

Qiu *et al.* [17] presented a survey of machine learning for Big Data, but they focused on the field of signal processing. Their study identified five critical issues (large scale, different data types, high speed of data, uncertain and incomplete data, and data with low value density) and related them to Big Data dimensions. Our study includes a more comprehensive view of challenges, but similarly relates them to the V dimensions. Furthermore, Qiu *et al.* [17] also identified various learning techniques and discussed representative work in signal processing for Big Data. Although they do a great work of identifying existing problems and possible solutions, the lack of categorization and direct relationship between each approach and its challenges makes it difficult to make an informed decision in terms of which learning paradigm or solution would be best for a specific use case or scenario. Consequently, in our work emphasis is on establishing correlation between solutions and challenges.

Al-Jarrah *et al.* [4] reviewed machine learning for Big Data focussing on the efficiency of large-scale systems and new algorithmic approaches with reduced memory footprint. Although they mentioned various Big Data hurdles, they did not present a systematic view as is done in this work. Al-Jarrah *et al.* were interested in the analytical aspect, and methods for reducing computational complexity in distributed environments were not considered. This work, on the other hand, considers both the analytical aspect and computational complexity in distributed environments.

Existing studies have effectively discussed the obstacles encountered by specific techniques such as deep learning [14], [18]. However, these studies focussed on a narrow aspect of machine learning; a more comprehensive view of challenges and approaches in the Big Data context is needed.

Similar to our work, Gandomi and Haider categorized challenges in accordance with the Big Data Vs [19]. However, their characterization is general and not in terms of machine learning.

Surveys on platforms for Big Data analytics have also been presented [20], [21]. Singh and Reddy [20] considered vertical and horizontal scaling platforms. They discussed the advantages and disadvantages of different platforms in terms of attributes such as scalability, I/O performance, fault tolerance, real-time processing, and iterative task support. Similarly, de Almeida and Bernardino [21] reviewed open source platforms including Apache Mahout, massive online analysis (MOA), the R Project, Vowpal, Pegasos, and GraphLab. These studies reviewed and compared existing platforms, while the present study relates these platforms to the challenges they address. Moreover, in this work, Big Data platforms are just one category of reviewed solutions.

The challenges of data mining with Big Data have been explored in the literature [22], [23]. Fan and Bifet [22] focused on challenges for data mining with Big Data and, as opposed to this work, they do not classify those challenges nor provide possible solutions. The work of Wu *et al.* [23], categorized the challenges, but their categorization is according to three tiers: Tier I (Big Data mining platforms), Tier II (Semantics and application knowledge), and Tier III (Big Data mining algorithms). In contrast, the categorization in this paper is according to the V dimensions. Whereas Wu *et al.* considered data mining, this study deals with machine learning. Moreover, the present study relates Big Data solutions to the challenges that they address.

To understand the origin of machine learning challenges, the present work categorizes them using the Big Data definition. In addition, various machine learning approaches are reviewed, and how each approach is capable of addressing known challenges is discussed. This enables researchers to make better informed decision regarding which learning paradigm or solution to use based on the specific Big Data scenario. It also makes it possible to identify research gaps and opportunities in the domain of machine learning with Big Data. Consequently, this work serves as a comprehensive foundation and facilitator for future research.

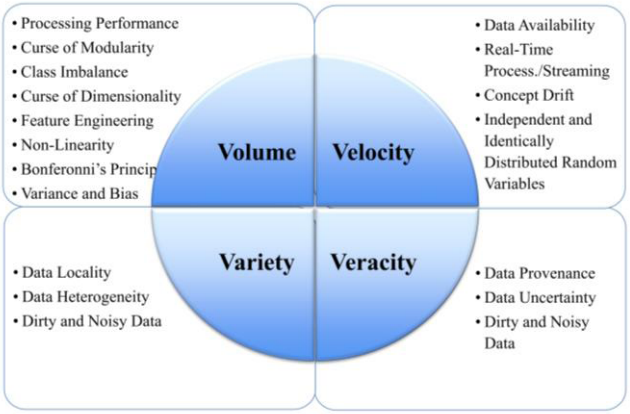

FIGURE 1. Big Data characteristics with associated challenges.

III. MACHINE LEARNING CHALLENGES ORIGINATING FROM BIG DATA DEFINITION

Big Data are often described by its dimensions, which are referred to as its Vs. Earlier definitions of Big Data focussed on three Vs [24] (volume, velocity, and variety); however, a more commonly accepted definition now relies upon the following four Vs [25]: volume, velocity, variety, and veracity. It is important to note that other Vs can also be found in the literature. For example, value is often added as a 5th V [22], [26]. However, value is defined as the desired outcome of Big Data processing [27] and not as defining characteristics of Big Data itself. For this reason, this paper considers only the four dimensions that characterize Big Data [28]. This provides an opportunity to relate challenges directly to the defining characteristics of Big Data, rendering the origin and cause of each explicitly. This section identifies machine learning challenges and associates each challenge with a specific dimension of Big Data. Fig. 1 illustrates the dimensions of Big Data along with their associated challenges as further discussed in the following sub-sections.

A. VOLUME

The first and the most talked about characteristic of Big Data is volume: it is the amount, size, and scale of the data. In the machine learning context, size can be defined either vertically by the number of records or samples in a dataset or horizontally by the number of features or attributes it contains. Furthermore, volume is relative to the type of data: a smaller number of very complex data points may be considered equivalent to a larger quantity of simple data [19]. This is perhaps the easiest dimension of Big Data to define, but at the same time, it is the cause of numerous challenges. The following sub-sections discuss machine learning challenges caused by volume.

1) PROCESSING PERFORMANCE

One of the main challenges encountered in computations with Big Data comes from the simple principle that scale, or volume, adds computational complexity. Consequently, as the scale becomes large, even trivial operations can become costly. For example, the standard support vector machine (SVM) algorithm has a training time complexity of $O(m^3)$ and a space complexity of $O(m^2)$ [29], where m is the number of training samples. Therefore, an increase in the size m will drastically affect the time and memory needed to train the SVM algorithm and may even become computationally infeasible on very large datasets. Many other ML algorithms also exhibit high time complexity: for example, the time complexity of principal component analysis is $O(mn^2 + n^3)$, that of logistic regression $O(mn^2 + n^3)$, that of locally weighted linear regression $O(mn^2 + n^3)$, and that of Gaussian discriminative analysis $O(mn^2 + n^3)$ [30], where m is the number of samples and n the number of features. Hence, for all these algorithms, the time needed to perform the computations will increase exponentially with increasing data size and may even render the algorithms unusable for very large datasets.

Moreover, as data size increases, the performance of algorithms becomes more dependent upon the architecture used to store and move data. Parallel data structures, data partitioning and placement, and data reuse become more important with growth in data size [31]. Resilient distributed datasets (RDDs) [31] are an example of a new abstraction for in-memory computations on large clusters; RDDs are implemented in the Spark cluster computing framework [32]. Therefore, not only does data size affect performance, but it also leads to the need to re-think the typical architecture used to implement and develop algorithms.

2) CURSE OF MODULARITY

Many learning algorithms rely on the assumption that the data being processed can be held entirely in memory or in a single file on a disk [33]. Multiple classes of algorithms are designed on strategies and building blocks that depend on the validity of this assumption. However, when data size leads to the failure of this premise, entire families of algorithms are affected. This challenge is referred to as the curse of modularity [15].

One of the approaches brought forward as a solution for this curse is MapReduce, a scalable programming paradigm for processing large datasets by means of parallel execution on a large number of nodes. Some machine learning algorithms are inherently parallel and can be adapted to the MapReduce paradigm, whereas others are difficult to decompose in a way that can take advantage of large numbers of computing nodes. Grolinger *et al.* [11] have discussed challenges for MapReduce in Big Data. The three main categories of algorithms that encounter the curse of modularity when attempting to use the MapReduce paradigm include iterative graph, gradient descent, and expectation maximization algorithms. Their iterative nature together with their dependence on in-memory data create a disconnect with the parallel and distributed nature of MapReduce. This leads to difficulties in adapting these families of algorithms to MapReduce or to another distributed computation paradigm.

Consequently, although some algorithms such as k-means can be adapted to overcome the curse of modularity through parallelization and distributed computing, others are still bounded or even unusable with certain paradigms.

3) CLASS IMBALANCE

As datasets grow larger, the assumption that the data are uniformly distributed across all classes is often broken [34]. This leads to a challenge referred to as class imbalance: the performance of a machine learning algorithm can be negatively affected when datasets contain data from classes with various probabilities of occurrence. This problem is especially prominent when some classes are represented by a large number of samples and some by very few.

Class imbalance is not exclusive to Big Data and has been the subject of research for more than a decade [35]. Experiments performed by Japkowicz and Stephen [35] have shown that the severity of the imbalance problem depends on task complexity, the degree of class imbalance, and the overall size of the training set. They suggest that in large datasets, there is a good chance that classes are represented by a reasonable number of samples; however, to confirm this observation, evaluations of real-world Big Data sets are needed. On the other hand, the complexity of Big Data tasks is expected to be high, which could result in severe impacts from class imbalance.

It is to expect that this challenge would be more common, severe, and complex in the Big Data context because the extent of imbalance has immense potential to grow due to increased data size. The same authors, Japkowicz and Stephen [35], showed that decision trees, neural networks, and support vector machine algorithms are all very sensitive to class imbalance. Therefore, their unaltered execution in the Big Data context without addressing class imbalance may produce inadequate results. Similarly, Baughman *et al.* [36] considered extreme class imbalance in

gamification and demonstrated its negative effects on Watson machine learning.

Consequently, in the Big Data context, due to data size, the probability that class imbalance will occur is high. In addition, because of the complex problems embedded in such data, the potential effects of class imbalance on machine learning are severe.

4) CURSE OF DIMENSIONALITY

Another issue associated with the volume of Big Data is the curse of dimensionality [37] which refers to difficulties encountered when working in high dimensional space. Specifically, the dimensionality describes the number of features or attributes present in the dataset. The Hughes effect [38] states that for a training set of static size, the predictive ability and effectiveness of an algorithm decreases as the dimensionality increases. Therefore, as the number of features increases, the performance and accuracy of machine learning algorithms degrades. This can be explained by the breakdown of the similarity-based reasoning upon which many machine learning algorithms rely [37]. Unfortunately, the greater the amount of data available to describe a phenomenon, the greater becomes the potential for high dimensionality because there are more prospective features. Consequently, as the volume of Big Data increases, so does the likelihood of high dimensionality.

In addition, dimensionality affects processing performance: the time and space complexity of ML algorithms is closely related to data dimensionality [30]. The time complexity of many ML algorithms is polynomial in the number of dimensions. As already mentioned, the time complexity of the principal component analysis is $O(mn^2 + n^3)$ and that of logistic regression $O(mn^2 + n^3)$, where m is the number of samples and n is the number of dimensions.

5) FEATURE ENGINEERING

High dimensionality is closely related to another volume challenge: feature engineering. This is the process of creating features, typically using domain knowledge, to make machine learning perform better. Indeed, the selection of the most appropriate features is one of the most time consuming preprocessing tasks in machine learning [14]. As the dataset grows, both vertically and horizontally, it becomes more difficult to create new, highly relevant features. Consequently, in a manner similar to dimensionality, as the size of the dataset increases, so do the difficulties associated with feature engineering.

Feature engineering is related to feature selection: whereas *feature engineering* creates new features in an effort to improve learning outcomes, *feature selection* (dimensionality reduction) aims to select the most relevant features. Although feature selection reduces dimensionality and hence has the potential to reduce ML time, in high dimensions it is challenging due to spurious correlations and incidental endogeneity (correlation of an explanatory variable with the error term) [39].

Overall, both feature selection and engineering are still very relevant in the Big Data context, but, at the same time they become more complex.

6) NON-LINEARITY

Data size poses challenges to the application of common methodologies used to evaluate dataset characteristics and algorithm performance. Indeed, the validity of many metrics and techniques relies upon a set of assumptions, including the very common assumption of linearity [40]. For example, the correlation coefficient is often cited as a good indicator of the strength of the relationship between two or more variables. However, the value of the coefficient is only fully meaningful if a linear relationship exists between these variables. An experiment conducted by Kiang [41] showed that the performance of neural networks and logistic regression is very negatively affected by non-linearity. Although this problem is not exclusive to Big Data, non-linearity can be expected to be more prominent in large datasets.

The challenge of non-linearity in Big Data also stems from the difficulties associated with evaluating linearity. Linearity is often evaluated using graphical techniques such as scatterplots; however, in the case of Big Data, the large number of points often creates a large cloud, making it difficult to observe relationships [40] and assess linearity.

Therefore, both the difficulty of assessing linearity and the presence of non-linearity pose challenges to the execution of machine learning algorithms in the context of Big Data.

7) BONFERONNI'S PRINCIPLE

Bonferonni's principle [42] embodies the idea that if one is looking for a specific type of event within a certain amount of data, the likelihood of finding this event is high. However, more often than not, these occurrences are bogus, meaning that they have no cause and are therefore meaningless instances within a dataset. This statistical challenge is also often described as spurious correlation [19]. In statistics, the Bonferonni correction theorem provides a means of avoiding those bogus positive searches within a dataset. It suggests that if testing m hypotheses with a desired significance of α, each individual hypothesis should be tested at a significance level of α/m [43].

However, the incidences of such phenomena increase with data size, and as data become exponentially bigger, the chances of finding an event of interest, legitimate or not, is bound to increase. Recently, Calude and Longo [44] have discussed the impact and incidence of spurious correlations in Big Data. They have shown that given a large enough volume, most correlations tend to be spurious. Therefore, including a means of preventing those false positives is important to consider in the context of machine learning with Big Data.

8) VARIANCE AND BIAS

Machine learning relies upon the idea of generalization; through observations and manipulations of data, representations can be generalized to enable analysis and prediction.

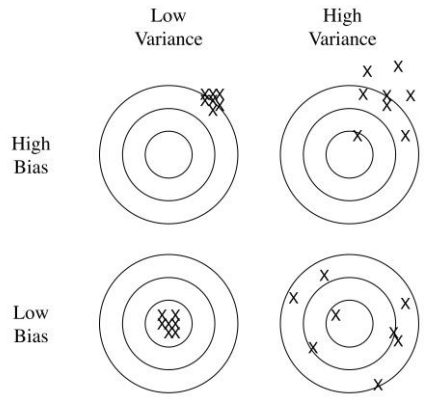

FIGURE 2. Variance and bias [37].

Generalization error can be broken down into two components: variance and bias [45]: Fig. 2 illustrates the relationship between them. *Variance* describes the consistency of a learner's ability to predict random things, whereas *bias* describes the ability of a learner to learn the wrong thing [37]. Ideally, both the variance and the bias error should be minimized to obtain an accurate output. However, as the volume of data increases, the learner may become too closely biased to the training set and may be unable to generalize adequately for new data. Therefore, when dealing with Big Data, caution should be taken as bias can be introduced, compromising the ability to generalize.

Regularization refers to techniques that aim to improve generalization and reduce overfitting; examples of regularization techniques include early stopping, Lasso, and Ridge [46]. Although these techniques improve generalization, they also introduce additional parameters that must be tuned to achieve good fit to unseen data. This is often done using approaches such as cross-validation, possibly with grid search; however, those require additional processing time, especially in the case of large datasets. Regularization techniques are well established in machine learning, but further investigation is needed with respect to their efficiency with Big Data.

B. VARIETY

The variety of Big Data describes not only the structural variation of a dataset and of the data types that it contains, but also the variety in what it represents, its semantic interpretation [7] and its sources. Although not as many as for other V dimensions, the challenges associated with this dimension have substantial impact.

1) DATA LOCALITY

The first challenge associated with variety is data locality [42]. Machine learning algorithms once again assume that the entire dataset is found in memory or in a single disk file [15]. However, in the case of Big Data, this may not be possible due to sheer size; not only do the data not fit into memory, but they are commonly distributed over large numbers of files residing in different physical locations. Traditional machine learning would first require data transfer to the computing location. With large datasets, transfer would result in processing latency and could cause massive network traffic.

Consequently, an approach of bringing computation to data as opposed to bringing data to computation has emerged. This is based on the premise that moving computation is cheaper, in terms of time and bandwidth, than moving data. This approach is especially prominent with Big Data. The MapReduce paradigm also uses it: map tasks are executed on the nodes where data reside, with each map task processing its local data. Moreover, a large number of NoSQL data stores adapt this model; as distributed storage solutions, they store data over a large number of nodes and then use the MapReduce paradigm to bring computation to data [47]. However, as already mentioned, MapReduce-based approaches encounter difficulties when working with highly iterative algorithms.

With small datasets, physical location is a non-issue; however, with Big Data, data locality is a paramount challenge that must be addressed in any successful Big Data system.

2) DATA HETEROGENEITY

Big Data analytics often involve integrating diverse data from several sources. These data may be diverse in terms of data type, format, data model, and semantics. Two main heterogeneity categories can be recognized: syntactic and semantic heterogeneity.

Syntactic heterogeneity refers to diversity in data types, file formats, data encoding, data model, and similar. To carry out analytics with integrated datasets, these syntactic variations must be reconciled [7]. Machine learning often requires a data pre-processing and cleaning step to configure data to fit within a specific model. However, with data coming from different sources, these data are likely formatted differently. Furthermore, the data to be processed may be of completely different types; for example, images may need to be processed along with categorical and numerical data. This causes difficulties for machine learning algorithms because they are not designed to recognize various types of representations at one time and to create efficient unified generalizations.

Semantic heterogeneity refers to differences in meanings and interpretations. As with syntactic, semantic heterogeneity increases in the case of Big Data when a number of datasets developed by different parties are integrated [48]. Again, machine learning approaches were not developed to handle semantically diverse data, and therefore heterogeneity must be resolved before applying such approaches.

In statistics, heterogeneity also refers to differences in statistical properties among the different parts of an overall dataset. Although present in small datasets, this challenge is enlarged in Big Data because datasets typically involve parts coming from different sources. This statistical heterogeneity breaks the common machine learning assumption that statistical properties are similar across a complete dataset.

Both syntactic and semantic heterogeneity as well as statistical heterogeneity have been active research topics for a

long time, but with the emergence of Big Data, they have attracted renewed attention [48]. The business value of data analytics typically involves correlating diverse datasets, and integration is crucial for carrying out machine learning over such datasets.

3) DIRTY AND NOISY DATA

According to Ratner [40], data possess their own set of distinct features that can be used for characterization:

- *Condition* defines the readiness of the data for analysis.
- *Location* refers to where the data physically reside.
- *Population* describes the entities and their sets of common attributes that together form the dataset.

Big Data are typically described as ill-conditioned due to the amount of time and resources necessary to get them ready for analysis. They also come from various locations and unknown populations. The combination of these properties leads to Big Data often being described as dirty.

Fan *et al.* [39] referred to such data as noisy data; they contain various types of measurement errors, outliers, and missing values. They discussed noise accumulation, which is especially severe with the high dimensionality typical in Big Data. It is important to note that Fan *et al.* considered noisy data one of the three main challenges of Big Data analysis.

Swan [49] suggested that data analysis should include a step to extract signal from noise directly following the steps of data collection and integration. She also recognized that Big Data may be too noisy to produce meaningful results.

The studies described above demonstrate the importance of dealing with noise in the context of generic Big Data analysis. Likewise, noise needs to be considered in machine learning with Big Data.

C. VELOCITY

The velocity dimension of Big Data refers not only to the speed at which data are generated, but also the rate at which they must be analyzed. With the omnipresence of smartphones and real-time sensors and the impending need to interact quickly with our environment through the development of technologies such as smart homes, the velocity of Big Data has become an important factor to consider.

1) DATA AVAILABILITY

Historically, many machine learning approaches have depended on data availability, meaning that before learning began, the entire dataset was assumed to be present. However, in the context of streaming data, where new data are constantly arriving, such a requirement cannot be fulfilled. Moreover, even data arriving at non-real-time intervals may pose a challenge.

In machine learning, a model typically learns from the training set and then performs the learned task, for example classification or prediction, on new data. In this scenario, the model does not automatically learn from newly arriving data, but instead carries out the already learned task on new data. To accommodate the knowledge embedded in new data, these models must be retrained. Without retraining, they may become outdated and cease to reflect the current state of the system.

Therefore, to adapt to new information, algorithms must support incremental learning [50], sometimes referred to as sequential learning, which is defined as an algorithm's ability to adapt its learning based on the arrival of new data without the need to retrain on the complete dataset. This approach does not assume that the entire training set is available before learning begins, but processes new data as they arrive. Although incremental learning is a relatively old concept, it is still an active research area due to the difficulty of adapting some algorithms to continuously arriving data [51].

2) REAL-TIME PROCESSING/STREAMING

Similar to the already discussed data availability challenge, traditional machine learning approaches are not designed to handle constant streams of data [19], which leads to another velocity dimension challenge - the need for real-time processing. This is subtly different from the data availability challenge: whereas data availability refers to the need to update the ML model as new data arrive, real-time processing refers to the need for real-time or near-real-time processing of fast-arriving data. The business value of real-time processing systems lies in their ability to provide instantaneous reaction; developers of algorithmic trading, fraud detection, and surveillance systems have been especially interested in such solutions [11].

The importance of real-time processing in today's era of sensors, mobile devices, and IoT has resulted in the emergence of a number of streaming systems; examples include Twitter's Storm [52] and Yahoo's S4 [53]. Although those systems have seen great success in real-time processing, they do not include sophisticated or diverse ML, but users can add ML features using external languages or tools.

The need exists to merge these streaming solutions with machine learning algorithms to provide instantaneous results; however, the complexity of such algorithms and the sparse availability of online learning solutions make this a difficult task.

3) CONCEPT DRIFT

Big Data are non-stationary; new data are arriving continuously. Consequently, acquiring the entire dataset before processing it is not possible, meaning that it cannot be determined whether the current data follow the same distribution as future data. This leads to another interesting challenge in machine learning with Big Data: concept drift [15]. *Concept drift* can be formally defined as changes in the conditional distribution of the target output given the input, while the distribution of the input itself may remain unchanged [54]. Specifically, this leads to a problem that occurs when machine learning models are built using older data that no longer accurately reflect the distribution of new data [55]. For example, energy consumption and demand prediction models can be built using data from electricity meters [56], but when

buildings are retrofitted to improve their energy efficiency, the present model does not accurately represent the new energy characteristics. Sliding window is a possible way of dealing with concept drift: the model is built using only the samples from the training window which is moved to include only the most recent samples. Windowing approach assumes that the most recent data is more relevant which may not always be true [54].

There exist various types of concept drift: incremental, gradual, sudden, and recurring [57], each bringing its own set of issues. However, the challenges typically lie in quickly detecting when concept drift is occurring and effectively handling the model transition during these changes. Like several already mentioned concepts, concept drift is not a new issue; mentions of it date back to 1986 [54]. However, the advent and nature of Big Data have increased frequency of its occurrence and have rendered some previous methodologies unusable. For example, Lavaire *et al.* [58] conducted an experiment on the influence of high dimensional Big Data on existing concept drift mitigation techniques. Their conclusions were that algorithm performance was highly degraded by the changes in the data. Therefore, finding new means to handle concept drift in the context of Big Data is an important task for the future of machine learning.

4) INDEPENDENT AND IDENTICALLY DISTRIBUTED RANDOM VARIABLES

Another common assumption in machine learning is that random variables are independent and identically distributed (i.i.d.) [59]; it simplifies underlying methods and improves convergence. In other words, i.i.d. assumes that each random variable has the same probability distribution as the others and that all are mutually independent. In reality, this may or may not be true. Moreover, some algorithms also depend on other distributions; for example the Markov sequence assumes that probability distribution of the next state depends only on the current state [60].

Nonetheless, Big Data by their very nature may prevent reliance on i.i.d. assumption based on the following [15]:

- i.i.d. requires data to be in random order while many datasets have a pre-existing non-random order. A typical solution would be to randomize the data before applying the algorithms. However, when dealing with Big Data, this becomes a challenge of its own and is often impractical.
- By their very nature, Big Data are fast and continuous. It is therefore not realistic to randomize a dataset that is still incomplete, nor is it possible to wait for all the data to arrive.

Dundar *et al.* [61] have shown that many typical machine learning algorithms such as back-propagation neural networks and support vector machines depend upon this assumption and could benefit greatly from a way of accounting for it. The high likelihood of a broken i.i.d. assumption with Big Data makes this challenge an important one to address.

D. VERACITY

The veracity of Big Data refers not only to the reliability of the data forming a dataset, but also, as IBM has described, to the inherent unreliability of data sources [19]. The provenance and quality of Big Data together define the veracity component [62], but also pose a number of challenges as discussed in the following sub-sections.

1) DATA PROVENANCE

Data provenance is the process of tracing and recording the origin of data and their movements between locations [63]. Recorded information, the provenance data, can be used to identify the source of processing error since it identifies all steps, transactions, and processes undergone by invalid data, thus providing contextual information to machine learning. It is therefore important to capture and retain this metadata [7].

However, as pointed out by Wang *et al.* [62], in the context of Big Data, the provenance dataset itself becomes too large, therefore, while these data provide excellent context to machine learning, the volume of these metadata creates its own set of challenges. Moreover, not only is this dataset too large, but the computational cost of carrying this overhead becomes overwhelming [62]. Although, certain methods have been brought forward to capture data provenance for specific data processing paradigms, such as the Reduce and Map Provenance (RAMP) developed for MapReduce as an extension for Hadoop [64], the added burden of provenance generally adds to the already high complexity and computational cost of machine learning with Big Data. Consequently, as provenance data provide a way to establish the veracity of Big Data, means of balancing its computational overhead and cost with the veracity value are needed.

2) DATA UNCERTAINTY

Data are now being gathered about various aspects of our lives in different ways; however, the means and methods used to gather data can introduce uncertainty and therefore impact the veracity of a dataset.

For example, sentiment data are being collected through social media [65], but although these data are highly important because they contain precious insights into subjective information, the data themselves are imprecise. The certainty and accuracy of this type of data is not objective because it relies only upon human judgment [20]. The lack of objectivity, or of absolute truth, within the data makes it difficult for a machine learning algorithm to learn from it.

Another recent method of capturing data is crowdsourcing; it solicits services or ideas from a large group of people. The data obtained from crowdsourcing, more particularly those gathered through participatory sensing, contain an even higher degree of uncertainty than sentiment data [7].

Moreover, inherent uncertainties exist in various types of data, such as weather or economic data for example, and even the most sophisticated data pre-processing methods

TABLE 1. Machine learning approaches and the challenges they address.

APPROACHES			CHALLENGES																	
			VOLUME								VARIETY			VELOCITY				VERACITY		
			Processing Performance	Curse of Modularity	Class Imbalance	Curse of Dimensionality	Feature Engineering	Non-linearity	Bonferonni's Principle	Variance and Bias	Data locality	Data Heterogeneity	Dirty and noisy Data	Data availability	Real-time Processing/Streaming	Concept drift	I.i.d	Data Provevance	Data Uncertainty	Dirty and Noisy Data
MANIPULATIONS	Data Manipulations	Dimensionality Reduction	✓			✓														
		Instance Selection	✓	✓																
		Data Cleaning											✓							✓
	Processing Manipulations	Vertical Scaling	✓															*		
		Hori-zontal Scaling: Batch-oriented	✓	✓		*					✓							*		
		Hori-zontal Scaling: Stream-oriented	✓	✓										✓	✓			*		
	Algorithm Manipulations	Algorithm Modifications	✓	*		*					✓				✓					
		Algorithm Mod. with new Paradigm	✓	*		*					✓				✓					
LEARNING PARADIGMS	Deep Learning						✓	✓				✓	*						*	*
	Online Learning		✓	✓	*						✓		*	✓	✓	*	✓			*
	Local Learning		✓	✓	✓					✓	✓									
	Transfer Learning				✓							✓	*						*	*
	Lifelong Learning		✓		✓							✓	*	✓	✓	*			*	*
	Ensemble Learning		✓	✓												✓				

cannot expunge this intrinsic unpredictability [66]. Once again, machine learning algorithms are not designed to handle this kind of imprecise data, thus resulting in another set of unique challenges for machine learning with Big Data.

3) DIRTY AND NOISY DATA

Furthermore, in addition to being imprecise, data can also be noisy [67]. For example, the labels or contextual information associated with the data may be inaccurate, or readings could be spurious. From the machine learning perspective this is different from imprecise data; having an unclear picture is different from having the wrong picture, although it may yield similar results. Noise and dirtiness come from various sources and are related to variety; many of the causes related to variety have been discussed in Section III.B. However, the noise challenge associated with crowdsourcing has yet to be discussed.

Crowdsourcing leads to uncertainty, especially when used for participatory sensing, but it can also lead to noisy data because it makes use of human judgment to assign labels to data. Moreover, the incorrect label can be either purposely or accidently assigned. The number of incorrect or noisy labels not only influences data veracity, but can also affect the performance of machine learning by potentially providing them with improperly labelled data.

Dirty and noisy data are not unique to Big Data, but the means by which they can be handled may not be easily adaptable to large datasets.

IV. APPROACHES

In response to the presented challenges, various approaches have been developed. Although designing entirely new algorithms would appear to be a possible solution [68], researchers have mostly preferred other methods. Many approaches have been suggested and surveys have been published on specific categories of solutions; examples include surveys on platforms for Big Data analytics [20], [21] and review of data mining with Big Data [23]. This paper reviews and organizes various proposed machine learning approaches and discusses how they address the identified challenges. The big picture of approach-challenge correlations is presented in Table 1; it includes a list of approaches along with the

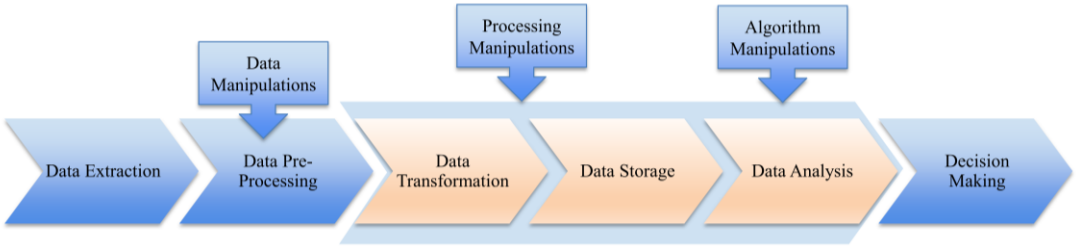

FIGURE 3. Data Analytics Pipeline.

challenges that each best addresses. Symbol '' indicate high degree of remedy while '∗' represents partial resolution.

As it can be seen from the table, there are two main categories of solutions. The first category relies on data, processing, and algorithm manipulations to handle Big Data. The second category involves the creation and adaptation of different machine learning paradigms and the modification of existing algorithms for these paradigms.

In addition to these two categories, it is important to note several machine learning as a service offerings: Microsoft Azure Machine Learning, now part of Cortana Intelligence Suite [69]; Google Cloud Machine Learning Platform [70]; Amazon Machine Learning [71]; and IBM Watson Analytics [72]. Because these services are backed up by powerful cloud providers, they offer not only scalability but also integration with other cloud platform services. However, at the moment, they support a limited number of algorithms compared to the R language [73], MATLAB [74], or Weka [75]. Moreover, computation happens on cloud resources, which requires data transfer to remote nodes. With Big Data, this results in high network traffic and may even become infeasible due to time or bandwidth requirements. Because these ML services are proprietary, information about their underlying technologies is very limited; therefore, this paper does not discuss them further.

The following sub-sections introduce techniques and methodologies being developed and used to handle the challenges associated with machine learning with Big Data. First, manipulation techniques used in conjunction with existing algorithms are presented. Second, various machine learning paradigms that are especially well suited to handle Big Data challenges are discussed.

A. MANIPULATIONS FOR BIG DATA

Data analytics using machine learning relies on an established suite of events, also known as the *data analytics pipeline*. The approaches presented in this section discuss possible manipulations in various steps of the existing pipeline. The purpose of these modifications is to respond to the challenges of machine learning with Big Data. Fig. 3 shows a representation of the pipeline based on the work of Labrinidis and Jagadish [76], along with the three types of manipulations to be discussed in this section: data manipulations, processing manipulations, and algorithm manipulations. These three categories, along with their corresponding sub-categories and sample solutions, are presented in Fig. 4. The examples included are only representatives and in no way provide a comprehensive list of solutions.

1) DATA MANIPULATIONS

One of the first manipulations to be attempted in an effort to adapt Big Data for machine learning is to try to modify the data in order to mimic non-Big Data. This modification takes place in the data pre-processing stage of the pipeline, as illustrated in Fig. 3.

Two of the most important data-related aspects affecting machine learning performance are high dimensionality (wide datasets) and large number of samples (high datasets). Therefore, two intuitive data manipulations for learning with Big Data are dimensionality reduction and instance selection as shown in Fig. 4. The term *data reduction* sometimes refers to both these manipulations, but occasionally specifically denotes instance selection. Additionally, data clearing is another important aspect of data manipulations.

Dimensionality reduction aims to map high dimensionality space onto lower-dimensionality one without significant loss of information. A variety of means exists to reduce dimensions in the context of Big Data. One popular, but very old technique (it originates from 1901) is *principal component analysis* (PCA). PCA belongs to the family of linear mapping techniques: orthogonal transformations are applied to transform a set of possibly correlated variables into a set of linearly uncorrelated variables, called principal components. The first principal component accounts for the largest proportion of the variability in the data, the second one has the next highest variance and is orthogonal to the first, and so on. Thus, choosing only the first p principal components can reduce dimensionality.

Examples of non-linear dimensionality reduction techniques, sometimes referred to as *manifold learning*, include kernel PCA, Laplacian Eigenmaps, Isomap, locally linear embedding (LLE), and Hessian LLE [77]. Random projection is another well-developed dimensionality reduction technique [78]. The idea behind it is to make use of random unit matrices to project the original dataset onto a lower-dimensional space. This technique has been used for a variety of data types such as text and images. However, it is limited to locally available static data. Therefore, although interesting,

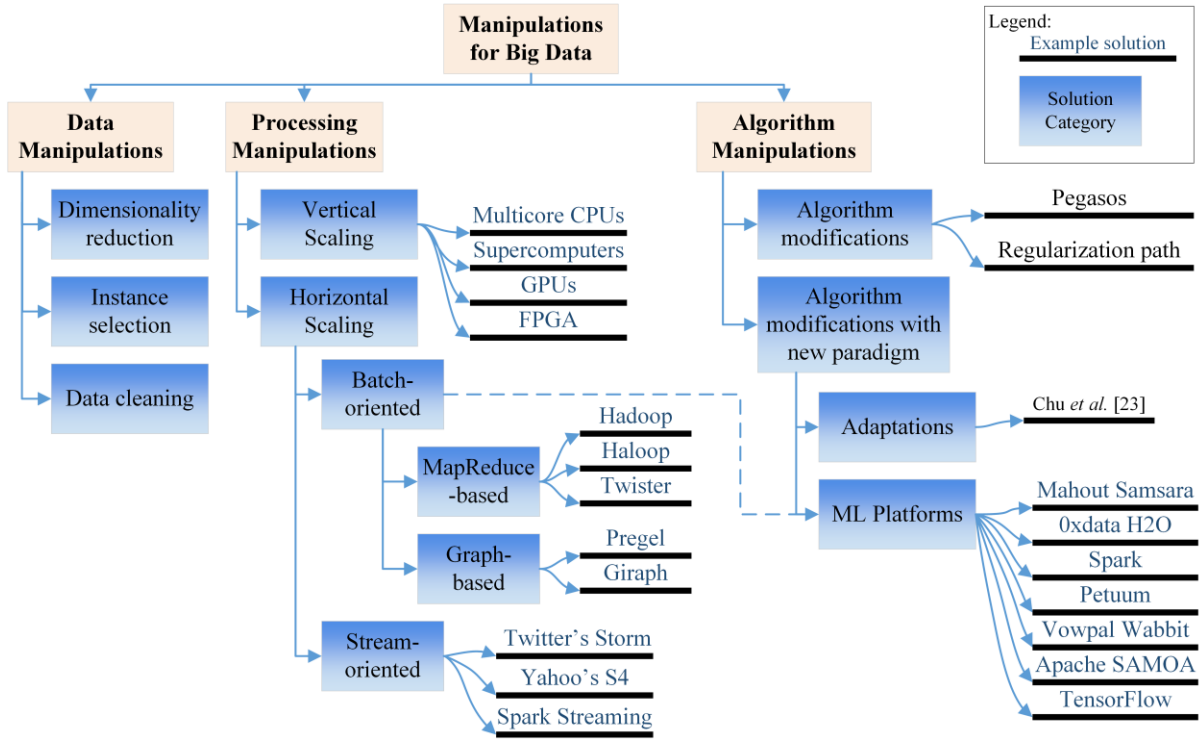

FIGURE 4. Manipulations for Big Data.

random projection addresses only the issues related with high dimensionality and is unable to mitigate other challenges such as data locality and availability.

Autoencoders have also been used to reduce dimensions; they learn an encoding of a dataset [14]. Their architecture is similar to a multi-layer perceptron (MLP): one input, one output, and one or more hidden layers. The difference from the MLP is that an autoencoder always has the same number of input and output nodes. Whereas the MLP learns the mapping between the input and target variables, an autoencoder learns to reconstruct its inputs. The hidden layers are responsible for encoding, they map the input feature space X to a lower-dimension space F, creating a compressed representation of X. The output layer on the other hand, serves as the decoder and reconstructs the input X from the compressed representation F.

Dimensionality reduction primarily addresses the curse of dimensionality and the processing performance challenges.

Instance selection refers to techniques for selecting a data subset that resembles and represents the whole dataset. Whereas dimensionality reduction deals with wide datasets, data reduction, more specifically instance selection, aims to reduce a dataset's height. The subset is consequently used to make inferences about the whole dataset. Instance selection approaches are diverse and include random selection, genetic algorithm-based selection, progressive sampling, using domain knowledge, and cluster sampling [79].

Although instance selection reduces dataset size thus improves processing performance and eases the curse of modularity, a number of questions arise:

- How big should the sample be? The sample size should balance accuracy and computing time.
- What sampling approach should be used? The choice of approach has a major impact on how well the subset represents the whole.
- How good will the model be? Instance selection introduces sampling error due to the differences between the sample and the whole dataset.

These issues, although well researched in learning with small datasets, are enlarged in the Big Data context because data size makes it more difficult to evaluate different properties or models.

Moreover, as already mentioned, in the Big Data context, challenges of class imbalance, noise, variance, and bias are more common and more difficult. In turn, this makes it more challenging to select a subset that will adequately represent the whole set. For example, with a large class imbalance, the selection approach must ensure that instances from all classes are selected. On the other hand, an appropriate instance selection can remedy class imbalance.

Data cleaning is another type of data manipulations; it refers to pre-processing such as noise and outlier removal. Thus, it tackles the challenges of dirty and noisy data. In this area, there is no significant development with respect to Big Data. Noise removal has been an especially active research topic in the audio, image, and video domains. Example techniques include smoothing filters and wavelet transforms [80]. However, such pre-processing is not practical for real-time

processing or when the data distribution may change over time. Autoencoders, in addition to dimensionality reduction, can perform denoising when they recover a signal from partially corrupted input data. Therefore, the challenges of noisy and dirty data can also be addressed by this method.

2) PROCESSING MANIPULATIONS

To improve machine learning performance with Big Data, processing manipulations focus on modifying how data are processed and stored. Here, the term *storage* refers not only to physical storage on a permanent medium, but also to how data are represented in memory. As illustrated in Fig. 3, processing manipulations can happen during three phases of the data analytics pipeline: data transformation, data storage, and data analysis.

In these stages, independent of the category of manipulations, processes can be embedded to capture data provenance and therefore remedy provenance challenge; an example is the Reduce and Map Provenance (RAMP) developed for MapReduce as an extension for Hadoop [64]. However, such process carries a significant computational overhead.

The main stream of solutions from this category includes those based on parallelization. Processing techniques can take advantage of the inherent parallel nature of certain algorithms. Many learning algorithms, such as brute-force search and genetic algorithms, are trivially parallel, and therefore parallelization can provide massive performance improvements. Consequently, researchers have developed techniques and tools to parallelize machine learning. Two categories of parallel systems can be distinguished: vertical and horizontal scaling paradigms.

The ***vertical scaling*** (scaling up) paradigm includes multicore CPUs, supercomputers (blades), hardware acceleration including graphic processing units (GPUs) and field-programmable gate arrays (FPGAs). In the Big Data context, it is often discarded because it is limited to resources available on a single node; however, for machine learning, it is important to highlight the GPU approach. GPUs are specially designed for manipulating images for output displays. The large number of cores (in thousands) compared to CPUs and the development of GPU interfaces such as Nvidia's CUDA have resulted in increased use of GPUs for general purpose processing. Because GPUs were originally designed for graphic display, image processing, matrix operations, and vector operations are especially suited for such systems. However, other ML algorithms can also be implemented on a GPU if they can be parallelized to a sufficiently high degree. Today, a large number of GPU accelerated ML algorithms are available [81]. Although it provides excellent performance through highly parallel processing, the size of data that GPU machine learning can process is limited by memory because typically processing happens on a single node.

FPGAs have been rarely mentioned in the context of Big Data. They are hardware components especially built for a specific purpose or application. Although this limits their applicability to Big Data, it is important to note the excellent FPGA performance achieved when scanning large amounts of network data [82].

The ***horizontal scaling*** (scaling out) paradigm refers to distributed systems where processing is dispersed over networked nodes. As with vertical scaling, processing is parallelized, but it also involves distributed nodes and hence network communication. Due to its capability to scale over large numbers of commodity nodes, distributed systems have been the focus of Big Data research.

This paradigm has been developed in two streams: batch- and stream-oriented systems.

Batch-oriented systems process a large amount of data at once, have access to most of the data, and typically are more concerned with throughput than with latency. MapReduce-based solutions such as Hadoop [83] and NIMBLE [84] belong to this category. Because MapReduce encounters difficulties when dealing with iterative algorithms, new solutions have been proposed: HaLoop [85] and Twister [86] extend Hadoop to provide better support for iterative processing. MapReduce-based solutions address the curse of modularity as they typically do not require the complete dataset to be held in memory. Moreover, data locality is also resolved as those solutions support work with data residing on different physical location. Such solutions facilitate work with high dimensional data, but they do not resolve the breakdown of the similarity-based reasoning, thus they provide partial resolution for the curse of dimensionality.

Similarly, to support this type of processing more effectively, to handle highly interconnected data, and accommodate graph algorithms, graph-based solutions have emerged. Pregel [87], the algorithm behind Google's PageRank, and Giraph [88], used by Facebook to analyze social connections, are examples from this category. They are based on the bulk synchronous parallel paradigm, and they retain states in memory, which facilitates iterative processing. Solutions from this category deal with the challenge of processing performance.

Stream-oriented systems operate on one data element or a small set of recent data in real-time or near real-time. Typically, computations performed in such a system are not as complex as those performed by batch systems. Examples from this category are Apache Storm [52], Yahoo's S4 [53], and Spark Streaming [32]. Although inspired by MapReduce, these platforms present a significant departure from Hadoop MapReduce; they have moved from in-memory data dependence to streams. Both Storm and S4 express computations using a graph topology, and their runtime engines handle parallelization, message passing, and fault tolerance. In contrast to Storm and S4, which perform one-by-one processing, Spark Streaming divides data into micro-batches and carries out computation on the micro-batches.

Stream-oriented systems enable mitigation of processing performance and real-time processing issues. They also mitigate the curse of modularity as they do not require the dataset to fit into memory and remedy the data availability challenge as they work with continuously arriving data. However,

streaming systems are only suitable for very simple machine learning. Gorawski *et al.* [89] and Cugola and Margara [90] surveyed data stream processing tools and complex event processing. While they do not mention machine learning specifically, their discussion on the ability of each tool or approach to meet specific requirements provides insights about proper tool and approach selection.

Research in ML with Big Data has focussed mainly on the horizontal scaling paradigm: MapReduce-based solutions, graph-based solutions, and streaming. As discussed earlier, each category addresses specific problems and encounters difficulties with others. All solutions from the processing manipulation category primarily focus on improving performance (throughput or latency) and do not remedy a number of other challenges as illustrated in Table 1. The combination of processing manipulations with algorithms and new learning paradigms provides research opportunities to undertake the remaining Big Data challenges.

3) ALGORITHM MANIPULATIONS

Algorithm manipulations include approaches that modify existing algorithms, with or without applying new paradigms. Since the very beginning of machine learning, researchers have been trying to improve existing algorithms and to reduce their time and/or space complexity. With Big Data, these efforts have intensified because it has become more important to handle large datasets.

As illustrated in Fig. 4, this study distinguishes two categories: *algorithm modifications* and *algorithm modifications with new paradigms* (approaches that involve modifying existing algorithms, applying process manipulations, and/or new paradigms).

Algorithm modifications have focussed on modifying algorithms to improve their performance. For example, the following approaches have been developed for specific machine learning algorithms to address volume challenges:

- Pegasos [91] provides an optimized version of the support vector machine (SVM) algorithm for large-scale text processing. Its runtime does not depend directly on training set size, and hence Pegasos is especially suitable for large datasets.
- Regularization paths [92] for linear models supports linear regression, two-class logistic regression, and multinomial regression problems. This approach enables processing of large datasets and efficiently handles sparse features.

Solutions from this category deal with processing performance and real-time processing. As they are typically distributed computing solutions, they also address the data locality challenge. Moreover, the curses of dimensionality and modularity are partially remedied as those solutions support work with high dimensional data and may provide smaller memory footprint.

Algorithm modifications with new paradigms category involves modifying ML algorithms to work better with new process manipulations and/or new paradigms. An example from this category would be to modify an algorithm through parallelization and to adapt the algorithm for a new parallel processing paradigm such as MapReduce. Chu *et al.* [30] adapted several algorithms to multicore MapReduce, including naïve Bayes, Gaussian discriminative analysis, k-means, neural networks, support vector machines, and others. As Chu *et al.* [30] have shown, by using an approach such as MapReduce, some algorithms can be modified to improve performance. However, other algorithms, especially iterative ones, cannot be easily parallelized, and consequently, due to the curse of modularity, whole families of algorithms may not be usable for this paradigm [11].

ML platforms are another type of solutions from this category; they combine algorithm adaptation with new paradigms. Solutions from this category started with disk-based systems such as Apache Mahout [93] with underlying Hadoop. Because disk access is slow, ML platforms evolved to memory-based solutions: examples include Apache Spark [32] and 0xdata H2O [94]. Spark is a distributed computing framework based on distributed datasets and in-memory processing. In addition to the Spark Core, which provides a distributed computing foundation, Spark also includes libraries built on top of the core, including Spark SQL, Spark Streaming, MLlib (machine learning library), and GraphX. The MLlib library offers a large number of machine learning algorithms, but it is still limited compared to the R language or MATLAB.

0xdata H2O is another distributed machine learning platform. Like Spark, it is an in-memory distributed system and can therefore support massively scalable data analytics. Whereas Spark focuses on providing a platform for in-memory analytics, the emphasis of H2O is specifically on scalable machine learning. H2O can be installed on top of Spark (Sparkling Water) to combine H2O's machine learning capabilities with the powerful Spark distributed platform. With the Mahout Samsara release, Mahout has also been transformed into a memory-based system.

Another example of a distributed machine learning platform is Petuum [95]. While Spark MLlib or H2O rely on general purpose distributed platforms (MLlib on Spark and H2O on Spark or Hadoop), Petuum is a complete platform developed specifically for machine learning. Vowpal Wabbit [96] is yet another interesting solution from this category; it uses an online learning approach. This means that data do not have to be pre-loaded into memory, and hence the memory footprint is not dependent on the number of samples.

Lastly Apache SAMOA [97] and Google's TensorFlow [98] both provide machine learning libraries for distributed processing environments. Unlike SAMOA which abstracts the distributed streaming nature of the processing from the user, TensorFlow makes use of data flow graphs as a flexible architecture to enable users to deploy computations across devices. Although these solutions are quite promising, they each are bound to a subset of challenges: SAMOA is

bound to stream processing and TensorFlow is designed for Deep Learning.

This work classifies ML platforms as an *algorithm modifications* solution; however, they also can be considered *processing modifications*, as illustrated by the dashed line in Fig. 4.

All algorithm modification solutions (with or without new paradigms) focus on providing the capability to process large datasets, improving performance, or providing real-time processing capabilities. They also remedy data locality as distributed computation can be performed with data residing on different physical locations. Algorithm modification with new paradigms, specifically ML platforms, mitigate the curse of modularity as they use memory of all nodes in the cluster and the course of dimensionality as they facilitate work with high dimensional data. However, they do not specifically address a number of other challenges as illustrated in Table 1. Although unable to provide a complete solution to all the challenges discovered, algorithm manipulations offer a means for researchers to deploy, modify, and adapt existing algorithms for Big Data in order to address some of the unaddressed concerns.

B. MACHINE LEARNING PARADIGMS FOR BIG DATA

A variety of learning paradigms exists in the field of machine learning; however, not all types are relevant to all areas of research. For example, Deng and Li [99] presented a number of paradigms that were applicable to speech recognition. Congruently, the work presented here includes machine learning paradigms relevant in the Big Data context, along with how they address the identified challenges.

1) DEEP LEARNING

Deep learning is an approach from the representation learning family of machine learning. Representation learning is also often referred to as feature learning [100]. This type of algorithm gets its name from the fact that it uses data representations rather than explicit data features to perform tasks. It transforms data into abstract representations that enable the features to be learnt. In a deep learning architecture, these representations are subsequently used to accomplish the machine learning tasks. Henceforth, because the features are learned directly from the data, there is no need for feature engineering. In the context of Big Data, the ability to avoid feature engineering is regarded as a great advantage due to the challenges associated with this process.

Deep learning uses a hierarchical learning process similar to that of neural networks to extract data representations from data. It makes use of several hidden layers, and as the data pass through each layer, non-linear transformations are applied. These representations constitute high level complex abstractions of the data [14]. Each layer attempts to separate out the factors of variation within the data. Because the output of the last layer is simply a transformation of the original input, it can be used as an input to other machine learning algorithms as well. Deep learning algorithms can capture

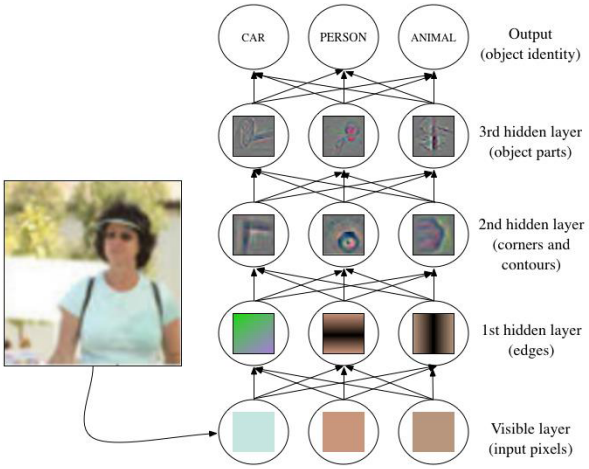

FIGURE 5. Deep Learning [101].

various levels of abstractions, thus this type of learning is an ideal solution to the problem of image classification and recognition. Fig. 5 provides an abstract view of the deep learning process [101]. Each layer learns a specific feature: edges, corners and contours, and object parts.

The deep learning architecture is versatile and can be built using a multitude of components: autoencoders and restricted Boltzmann machine are typical building blocks [14]. Autoencoders are unsupervised algorithms that can be used for many purposes such as anomaly detection, but in the context of Big Data, they typically serve as a precursor step with neural networks [102]. They work through backpropagation by attempting to set their target output as their input, thereby auto-encoding themselves. Boltzmann machines [103] are similar except that they use a stochastic rather than deterministic process. Deep belief networks [104] are another example of deep learning algorithms.

Furthermore, the reliance upon an abstract representation also makes these algorithms more flexible and adaptable to data variety. Because the data are abstracted, the diverse data types and sources do not have a strong influence on the algorithm results, making deep learning a great candidate for dealing with data heterogeneity.

Interestingly, deep learning can be used for both supervised and unsupervised learning [18]. This is possible due to the very nature of the technique; it excels at extracting global relationships and patterns from data because of its reliance upon creating high level abstractions. In the context of Big Data, this is a great advantage as it renders the algorithms less sensitive to veracity challenges such as dirty, noisy, and uncertain data [18]. Moreover, its multiple layers of non-linear transformations addresses the challenge associated with data non-linearity. Deep compression has been proposed as a way of speeding up processing without the loss of accuracy [105].

According to the described characteristics, deep learning seems to be well suited to address many of the previously identified challenges such as feature engineering, data

heterogeneity, non-linearity, noisy and dirty data, and data uncertainty. However, those algorithms are not fundamentally built to learn incrementally [106] and are therefore susceptible to the data velocity issue. Although they are especially well adapted to handle large datasets with complex problems, they do not do so in a computationally efficient way. For high dimensional data [14] or large numbers of samples such algorithms may even become infeasible, making deep learning susceptible to the curse of dimensionality.

2) ONLINE LEARNING

Because it responds well to large-scale processing by nature, online learning is another machine learning paradigm that has been explored to bridge efficiency gaps created by Big Data. Online learning can be seen as an alternative to batch learning, the paradigm typically used in conventional machine learning. As its name implies, batch learning processes data in batches and requires the entire dataset to be available when the model is created [42]. Furthermore, once generated, the model can no longer be modified. This makes it difficult to deal with the dimensions of Big Data for the following reasons:

- Volume: having to process a very large amount of data at one time is not computationally efficient or always feasible.
- Variety: the need to have the entire dataset available at the beginning of the processing limits the use of data from various sources.
- Velocity: the requirement to have access to the entire dataset at the time of processing does not enable real-time analysis or use of data from various sources.
- Veracity: because the model cannot be altered, it is highly susceptible to performance impediments caused by poor data veracity.

Conversely, online learning uses data streams for training, and models can learn one instance at a time [18]. This "learn-as-you-go" paradigm alleviates the computational load and processing performance because the data do not have to be entirely held in memory. This enables processing of very large volumes of data, remedies the curse of modularity, facilitates real-time processing, and provides the ability to learn from non-i.i.d. data [18]. Moreover, as it does not require all data to be present at once or located at the same place, this paradigm remedies data availability and locality.

Furthermore, the descriptor "online" also reflects the fact that this paradigm continuously maintains its model; the model can be modified whenever the algorithm sees fit. Its adaptive nature makes it possible to handle a certain amount of dirty and noisy data, class imbalances, and concept drift. Indeed, Mirza *et al.* [107] proposed an ensemble of subset online sequential extreme learning machines to achieve a solution for concept drift detection and class imbalance, while Kanoun and van der Shaar [108] presented an online learning solution for remedying the challenge of concept drift.

It is apparent that the online learning architecture responds well to the challenges associated with Big Data velocity; its incremental learning nature alleviates challenges of data availability, real-time processing, i.i.d, and concept drift. For example, this paradigm could be used to handle stock data prediction due to the ever-changing and rapidly evolving nature of the stock market. However, the issues associated with dimensionality, feature engineering, and variety remain unresolved. Moreover, not all machine learning algorithms can be easily adapted to the online learning paradigm.

3) LOCAL LEARNING

First proposed by Bottou and Vapnik in 1992 [109], local learning is a strategy that offers an alternative to typical global learning. Conventionally, ML algorithms make use of global learning through strategies such as generative learning [110]. This approach assumes that based upon the data's underlying distribution, a model can be used to re-generate the input data. It basically attempts to summarize the entire dataset, whereas local learning is concerned only with subsets of interest. Therefore, local learning can be viewed as a semi-parametric approximation of a global model. The stronger but less restrictive assumptions of this hybrid parametric model yield low variance and bias [4].

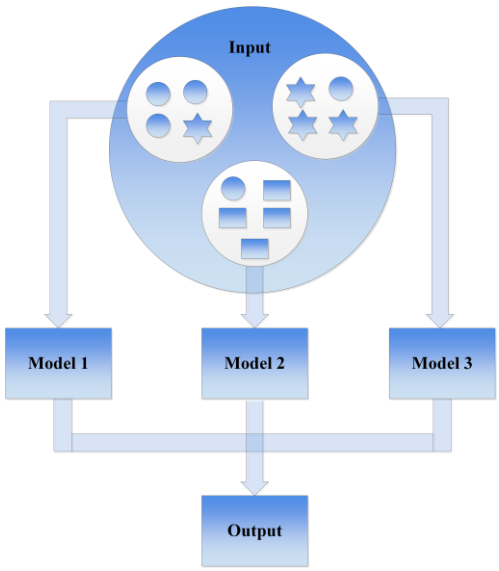

FIGURE 6. Local Learning.

Fig. 6 provides an abstract view of the local learning process. The idea behind it is to separate the input space into clusters and then build a separate model for each cluster. This reduces overall cost and complexity. Indeed, it is much more efficient to find a solution for k problems of size m/k than for a single problem of size m. Consequently, such approach could enable processing of datasets that were considered too large for global paradigms. Another way of implementing local learning is to modify the learning algorithms so that only neighbouring samples influence the output variable.

A typical example where local learning would be beneficial is, for instance, predicting the energy consumption of

several customers. Building a model for similar customers could be favourable to building one unique model for all customers or to building one model per customer.

Recently, Do and Poulet [111] developed a parallel ensemble learning algorithm of random local support vector machines that was able to perform much better than the typical SVM algorithm in addressing volume related issues, thereby demonstrating how local learning can help alleviate some of the issues associated with Big Data. Moreover, local learning has outperformed global learning in terms of accuracy and computation time in several forecasting studies [112], [113].

Dividing the problem into manageable data chunks reduces the size of data that need to be handled and potentially loaded into memory at once; therefore, this paradigm alleviates the curse of modularity. In addition, because of the locality of each cluster, models are not significantly affected by the challenges associated with class imbalances and data locality. Recent work has shown that local learning often yields better results than global learning when dealing with imbalanced datasets [4].

Therefore, the challenge of the curse of modularity, class imbalance, variance and bias, and data locality can be alleviated by a local approach. However, matters of dimensionality and velocity, such as concept drift among others, have yet to be addressed. Overall, in the Big Data context, the local approach remains largely unexplored; studying how this paradigm could better handle velocity and veracity challenges appears to be particularly open.

4) TRANSFER LEARNING

Transfer learning is an approach for improving learning in a particular domain, referred to as the *target domain,* by training the model with other datasets from multiple domains, denoted as *source domains*, with similar attributes or features, such as the problem and constraints. This type of learning is used when the data size within the target domain is insufficient or the learning task is different [114]. Fig. 7 shows an abstract view of transfer learning.

The distinguishing characteristic of transfer learning from other traditional ML approaches is fact that the training set does not necessarily come from the same domain as the testing set. Moreover, it can train on data from several domains individually or combined together using regular or adapted machine learning algorithms; domains do not have to have the same data distribution or the same feature space [115]. Different elements can be transferred from the source domains into the target domain: instances, feature representations, model parameters, and relational knowledge [116]. An example use case is energy consumption prediction for a new building (the target domain) using datasets collected from other similar buildings (the source domains) that probably have similar consumption patterns, but are different in size and efficiency.

Consequently, transfer learning is a possible candidate for resolving some of the challenges related to the volume, variety, and veracity dimensions of Big Data environments.

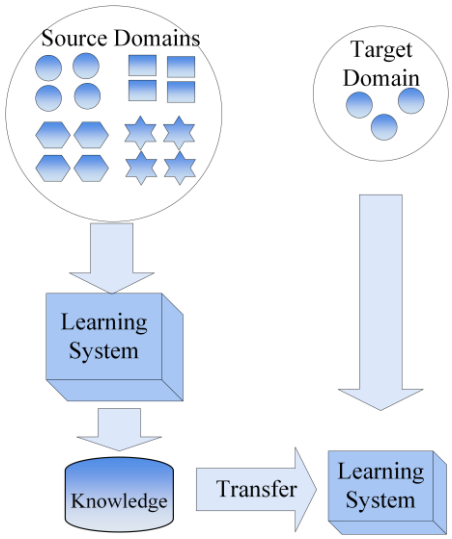

FIGURE 7. Transfer Learning.

From the volume category, it can remedy class imbalance as the instances can be transferred from other diverse domains to better balance classes in the target domain. Because of the ability to learn from different domains and transfer knowledge between different datasets, transfer learning is a promising solution for the data heterogeneity challenge (the variety category). Moreover, instance transfer from different domains can contribute to reducing challenges of dirty and noisy data as well as data uncertainty (the variety and veracity categories).

A few studies have discussed transfer learning for Big Data. Yang *et al.* [115] introduced an automatic transfer learning algorithm for Big Data. They improved a supervised learning approach, the Laplacian eigenmaps algorithm, by enabling automatic knowledge transfer from the source domains to the target domain. The system was designed for analyzing short text data using knowledge from the long text obtained from the Web. Zhang [48] described several examples of transfer learning with Big Data based on the concept of data fusion among homogenous and heterogeneous datasets; data fusion refers to combining data from several sources. For these reasons, transfer learning is one of the paradigms that address data size and heterogeneity.

5) LIFELONG LEARNING

Lifelong learning mimics human learning; learning is continuous; knowledge is retained and used to solve different problems. It is directed to maximize overall learning, to be able to solve a new task by training either on one single domain or on heterogeneous domains collectively [117]. The learning outcomes from the training process are collected and combined together in a space called the *topic model* or *knowledge model*. In the case of training on heterogeneous domains, transfer learning might be used in the combining step to create such a topic model. The existing knowledge in this topic

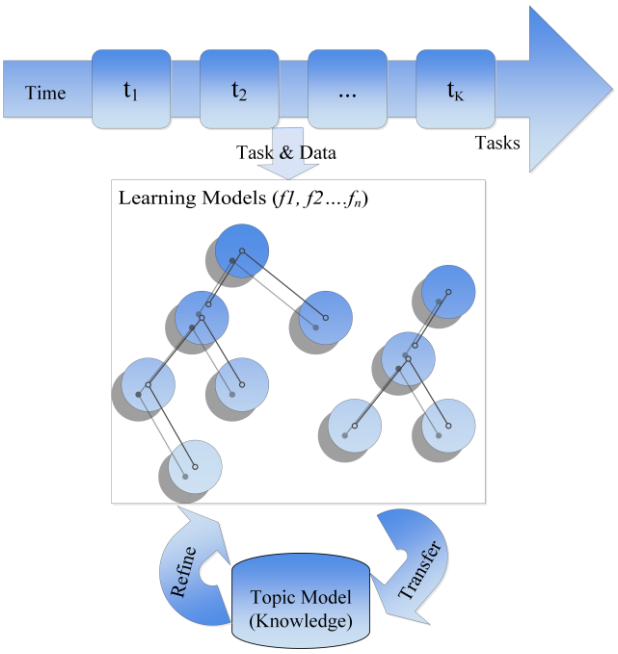

FIGURE 8. Transfer Learning.

model is used to perform a new task regardless of where the knowledge comes from.

Lifelong learning is related to online learning and transfer learning. Like online learning, lifelong learning is a continuous process; however, whereas online learning considers only a single domain, lifelong learning includes a multitude of domains. Like transfer learning, lifelong learning is capable of transferring knowledge among domains. But, unlike transfer learning, lifelong learning is a continuous process over time because the topic space is refined each time a new learning outcome arrives. For example, consider the process of inferring individuals' sport interests when various data are available such as their physical location, social media interactions, weather data, and Web browsing history. The prediction of individual interests needs to change periodically over time when new data become available because the area of interest may shift over time. Fig. 8 depicts a high level view of lifelong learning.

Traditional machine learning algorithms have a single target domain and cannot transfer learning from one task to another (with the exception of transfer learning). Consequently, Khan *et al.* [118] consider such approaches unsuitable for Big Data because of the many domains covered by such data and the constant appearance of new ones. They consider lifelong learning to be a good candidate solution for Big Data because the model is continuously refined and the learned task is applicable to different domains.

Similar to online learning, lifelong learning is a promising solution for processing performance, real-time processing, data availability, and concept drift. This is because it does not rebuild the model every time a new piece of data arrives, but only updates the existing knowledge based on the new incoming data. New knowledge is of course added to the existing knowledge base for all future tasks. Since lifelong learning incorporates transfer learning as its integral part, it addresses the same issues as transfer learning: the data heterogeneity, class imbalance, dirty and nosy data, and data uncertainty. However, presently there has been little development in this area because achieving this vision is very challenging.

Lifelong learning has been applied in various domains. Chen and Liu [119] suggested an unsupervised learning algorithm, the Gibbs Sampler, for mining the topic model and for generating a better knowledge space in the context of e-commerce reviews. In their work, the Big Data are divided into small sets, each set is trained individually, and the learning results are compared to yield a better topic model. Suthaharan [120] discussed issues related to combining machine learning, including lifelong learning, with Big Data technologies such as Hadoop and Hive. They focussed on network intrusion detection where data are growing quickly and arriving fast from different sources.

Khan *et al.* [118] surveyed lifelong learning models, including probabilistic topic models and knowledge-based topic models, for natural language processing with large data volumes. Their work discussed how these models can be used with Big Data to improve learning performance and accuracy.

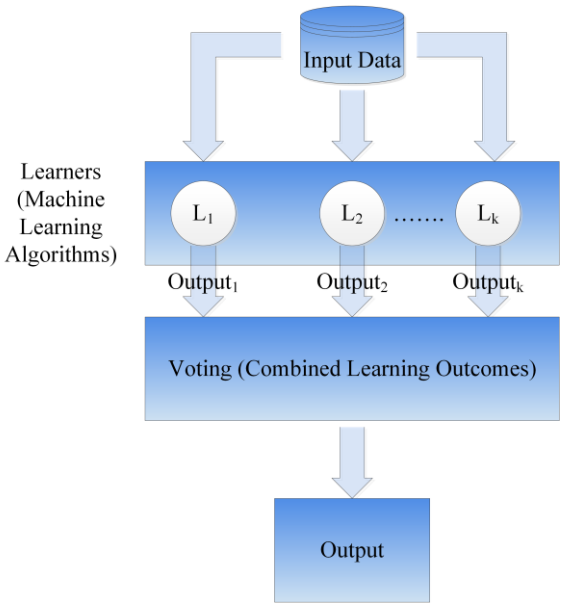

FIGURE 9. Ensemble Learning.

6) ENSEMBLE LEARNING

Ensemble learning combines multiple learners to obtain better learning outcomes (e.g., prediction, classification) than those obtained from any constituent learner [121]. Fig. 9 presents an abstract view of the ensemble learning process. Typically, the overall outcome is determined by a voting process among the weighted outcomes of individual learners [121]. These individual learners can be similar or from completely different categories, including those belonging to supervised and unsupervised ML. The weighting

mechanism assigns a value to each learning output point and combines them. The voting process could be implemented in a straightforward way by directly aggregating the values of the learning points or through the use of the statistical techniques to obtain a combined value of the learning outputs that may lead to better learning performance [122]. Waske and Benediktsson [123] have applied SVM for individual learners and also used an SVM in the voting process.

There are two main ways to apply ensemble learning: the first one trains different learners, each one on the complete dataset, whereas the second one splits the dataset and trains each learner (same or different) only on a subset. The second approach has potential in the Big Data context because it can speed up and improve the learning process. Basically, improvement is achieved by splitting up large volumes of data into small disjoint datasets. One or more machine learning algorithm(s) are then trained on a different disjoint dataset. The outcomes from all learners are combined using some kind of voting scheme to improve the accuracy of the overall solution. For example, to detect anomalous behavior, ensemble learning could be applied by breaking a large dataset into small ones, training learners on small sets, and combining results to identify anomalies with high accuracy, but with few false alarms.

Several studies have discussed applying ensemble learning to Big Data. For example, Tang *et al.* [124] used it to decrease computation time and simultaneously improve learning accuracy. They split the large dataset into smaller datasets using a probabilistic approximation technique called the Reservoir sampling method.

Research studies have shown that ensemble learning performs well with different datasets [125], [126]. Zang *et al.* [125] presented a comparative study of ensemble and incremental learning. They observed that incremental algorithms were faster, but ensemble models performed better in the presence of concept drift. Alabdulrahman [126] experimented with four different datasets; he investigated correlations among ensemble sizes, base classifiers, and voting methods.

Works of Tang *et al.* [124] and Gruber *et al.* [127] used ensemble learning mainly to minimize computing time by dividing a big dataset into small training sets. Unlike the work of Tang *et al.* [124], Gruber *et al.* [127] suggested invoking several different machine learning algorithms on the small sets. Then the output values of all algorithms were weighted, combined, and evaluated using inverse probability weights (i.e., the voting phase). Finally, the authors showed that ensemble learning could also be used to choose learning algorithms by removing those that did not have an impact on the final outcome. This approach consequently decreased computing time.

Ensemble learning plays a key role in emphasizing correctness of learning outcomes as well as speeding up the learning process. With respect to correctness, using several different learners or training with different subsets has the potential to minimize the error rates. On the other hand, similar to local learning, splitting the dataset and training on subsets improves processing performance as it is more efficient to find a solution for k problems of size m/kthan for a single problem of size m. This data splitting also reduces data chunks that need to be loaded into memory at once making ensemble learning capable of addressing the curse of modularity. Moreover, ensemble learning is capable of addressing concept drift [125]. How ensemble learning could be modified to best handle the issues related to variety and velocity remains to be explored.

V. DISCUSSION

Ratner [40] describes machine learning as a field that makes use of algorithms to solve problems with an underlying statistical component, such as regression, classification, and clustering, by means of an assumption-free non-parametric approach. For years, researchers have used machine learning to solve such problems without worrying about whether the situations they were facing met the requirements and the classical statistical assumptions upon which certain methodologies rely [40]. Arguably, the lack of concern with regard to these assumptions has enabled scientists to advance more quickly in the field of machine learning than in the field of statistics [40]. However, with the advent of Big Data, many of the assumptions upon which the algorithms rely have now been broken, thereby impeding the performance of analytical tasks. In response to those pitfalls, together with the need to process large datasets fast, a number of new machine learning approaches and paradigms have been developed. However, it remains consistently difficult to find the best tools and techniques to tackle specific challenges.

This paper has identified and presented challenges in machine learning with Big Data, reviewed emerging machine learning approaches, and discussed how each approach is capable of addressing the identified challenges. The overview of the relation between challenges and approaches has been presented in Table 1. A single application or a use case may encounter several challenges and may need to combine several approaches to handle those challenges. The following use cases demonstrate the use of the presented work:

Case study 1: Energy Prediction. Sensor-based forecasting using historical sensor readings to infer future energy consumption has gained popularity due to proliferation of sensors and smart meters. Grolinger *et al.* [103] considered energy forecasting with large sensor data. They encountered two main challenges from the volume category: processing performance (long time to build the model) and curse of modularity (complete data set fitting into memory). To resolve those challenges, they applied local learning, thus complying with Table 1 which indicates that local learning addresses challenges of processing performance and curse of modularity. As indicated in Table 1, those challenges could potentially be solved with other approaches such as horizontal scaling. In their study, local learning not only significantly improved the performance, but also increased prediction accuracy.

Case study 2: Recommender system. Collaborative filtering mechanisms are capable of suggesting relevant online content to users based on their browsing history. Millions of digital transactions are collected daily and they need to be continuously analyzed to improve recommendations and increase user engagement. Like case study 1, to address processing performance and curse of modularity challenges, Bachmann [128] applied local learning. Additionally, local learning enabled embedding contextual awareness into collaborative filtering.

Case study 3: Machine translation. Machine translation systems are computationally expensive and, in the case of large data sets, may even be prohibitive [129]. To handle this processing performance challenge, Google uses deep learning, specifically TensorFlow which is one of the ML platforms as illustrated in Figure 4. TensorFlow can be deployed on one or more CPUs or GPUs making this a horizontally and vertically scalable approach. Thus, Google machine translation uses a combination of several approaches (deep learning, horizontal and vertical scaling) to address the performance challenge.

Case study 4: Activity recognition. The constant advancements of the IoT has led to the development of numerous sensor equipped wearable devices. Saeedi *et al.* [130] proposed autonomous reconfiguration of wearable systems in order to handle impact of configuration changes on activity recognition. The main challenge they faced lied in the heterogeneity of the data; wearable sensor data contains a variety of signal heterogeneity such as subject, device and sampling frequency related heterogeneity. In accordance with Table 1, transfer learning was used to tackle this Big Data challenge. Deep learning and lifelong learning could have also been considered. Nevertheless, the authors reported a performance increase between 3-13% due to their solution.

The correlation between approaches and Big Data challenges presented in Table 1. makes it possible to identify the main opportunities and directions for future research in machine learning with Big Data. Because a single approach may address more than one challenge and several challenges may be addressed with a single approach, the future directions are not categorized according to V dimensions, but they represent broad research opportunities:

- Data fusion will become even more important as researchers and industry try to combine data from a number of different sources with different formats and semantics to provide new insights.
- Process and algorithm manipulations are expected to continue to attract significant research interest because they make it possible to use traditional algorithms adapted to work with Big Data.
- Paradigms that enable updating of existing models upon arrival of new data without the need to retrain the complete model are very promising because they can accommodate bigger datasets than batch learning. Paradigms from this category are online learning and lifelong learning.
- Stream processing is presently limited to relatively simple problems. However, with new developments, it is anticipated that it will be able to handle more complex computations.
- Online learning may possibly merge with stream processing. If online learning can update its model in real time or near real time, it can be integrated into stream processing.
- Integration of various approaches such as deep learning and online learning presents itself as an interesting and promising research area worthy of further consideration. A combination of various approaches would ensure better coverage of the issues related to machine learning with Big Data.

From Table 1 it can be noted that there is no correlation between Bonferonni's principle and any of the reviewed approaches. Although its impact in the Big Data context is large [44], we are not aware of a specific Big Data solution addressing this problem. Preventing those false positives is important, but it appears it attracted very limited attention from the Big Data community.

VI. CONCLUSIONS

This paper has provided a systematic review of the challenges associated with machine learning in the context of Big Data and categorized them according to the V dimensions of Big Data. Moreover, it has presented an overview of ML approaches and discussed how these techniques overcome the various challenges identified.

The use of the Big Data definition to categorize the challenges of machine learning enables the creation of cause-effect connections for each of the issues. Furthermore, the creation of explicit relations between approaches and challenges enables a more thorough understanding of ML with Big Data. This fulfills the first objective of this work; to create a foundation for a deeper understanding of machine learning with Big Data.

Another objective of this study was to provide researchers with a strong foundation for making easier and better-informed choices with regard to machine learning with Big Data. This objective was achieved by developing a comprehensive matrix that lays out the relationships between the various challenges and machine learning approaches, thereby highlighting the best choices given a set of conditions. This paper enables the creation of connections among the various issues and solutions in this field of study, which was not easily possible on the basis of the existing literature.

From the development or adaptation of new machine learning paradigms to tackle unresolved challenges, to the combination of existing solutions to achieve further performance improvements, this paper has identified research opportunities. This work has therefore accomplished its last objective by providing the academic community with potential directions for future work and will hopefully serve as groundwork for great improvements in the field of machine learning with Big Data.

REFERENCES

[1] R. Krikorian. (2010). *Twitter by the Numbers, Twitter*. [Online]. Available: http://www.slideshare.net/raffikrikorian/twitter-by-the-numbers?ref=http://techcrunch.com/2010/09/17/twitter-seeing-6-billion-api-calls-per-day-70k-per-second/

[2] ABI. (2013). *Billion Devices Will Wirelessly Connect to the Internet of Everything in 2020*, ABI Research. [Online]. Available: https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne/

[3] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: Promise and potential," *Health Inf. Sci. Syst.*, vol. 2, no. 1, pp. 1–10, 2014.

[4] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient machine learning for big data: A review," *Big Data Res.*, vol. 2, no. 3, pp. 87–93, Sep. 2015.

[5] M. A. Beyer and D. Laney, "The importance of 'big data': A definition," Gartner Research, Stamford, CT, USA, Tech. Rep. G00235055, 2012.

[6] V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Boston, MA: Houghton Mifflin Harcourt, 2013.

[7] H. V. Jagadish *et al.*, "Big data and its technical challenges," *Commun. ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[8] M. James, C. Michael, B. Brad, and B. Jacques, *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. New York, NY: McKinsey Global Institute, 2011.

[9] M. Rouse. (2011). *Machine Learning Definition*. [Online]. Available: http://whatis.techtarget.com/definition/machine-learning

[10] M. Rouse. (2009). *Predictive Analytics Definition*. [Online]. Available: http://searchcrm.techtarget.com/definition/predictive-analytics

[11] K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison, and M. A. M. Capretz, "Challenges for MapReduce in big data," in *Proc. IEEE World Congr. Services (SERVICES)*, Jun. 2014, pp. 182–189.

[12] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. 6th Symp. Oper. Syst. Design Implement.*, 2004, pp. 137–149.

[13] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Proc. IEEE 26th Symp. Mass Storage Syst. Technol. (MSST)*, May 2010, pp. 1–10.

[14] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep Learning Applications and Challenges in Big Data Analytics," *J. Big Data*, vol. 2, no. 1, p. 1, Feb. 2015.

[15] C. Parker, "Unexpected challenges in large scale machine learning," in *Proc. 1st Int. Workshop Big Data, Streams Heterogeneous Source Mining Algorithms, Syst., Programm. Models Appl. (BigMine)*, 2012, pp. 1–6.

[16] S. R. Sukumar, "Machine learning in the big data era: Are we there yet?" in *Proc. 20th ACM SIGKDD Conf. Knowl. Discovery Data Mining, Workshop Data Sci. Social Good (KDD)*, 2014, pp. 1–5.

[17] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP J. Adv. Signal Process.*, vol. 67, pp. 1–16, Dec. 2016.

[18] X.-W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.

[19] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *Int. J. Inf. Manage.*, vol. 35, no. 2, pp. 137–144, Apr. 2015.

[20] D. Singh and C. K. Reddy, "A survey on platforms for big data analytics," *J. Big Data*, vol. 2, no. 1, pp. 1–20, 2015.

[21] P. D. C. de Almeida and J. Bernardino, "Big data open source platforms," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2015, pp. 268–275.

[22] W. Fan and A. Bifet, "Mining big data: Current status, and forecast to the future," *SIGKDD Explorations Newslett.*, vol. 14, no. 2, pp. 1–5, Dec. 2012.

[23] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[24] R. Narasimhan and T. Bhuvaneshwari, "Big data—A brief study," *Int. J. Sci. Eng. Res.*, vol. 5, no. 9, pp. 350–353, 2014.

[25] F. J. Ohlhorst, *Big Data Analytics: Turning Big Data into Big Money*, vol. 15. Hoboken, NJ: Wiley, 2012.

[26] Y. Demchenko, P. Grosso, C. De Laat, and P. Membrey, "Addressing big data issues in scientific data infrastructure," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2013, pp. 48–55.

[27] M. Ali-ud-din Khan, M. F. Uddin, N. Gupta, and N. Gupta, "Seven V's of big data understanding big data to extract value," in *Proc. Zone Conf. Amer. Soc. Eng. Edu.*, Apr. 2014, pp. 1–5.

[28] R. Kune, P. K. Konugurthi, A. Agarwal, R. R. Chillarige, and R. Buyya, "The anatomy of big data computing," *Softw., Pract. Exper.*, vol. 46, no. 1, pp. 79–105, 2016.

[29] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *J. Mach. Learn. Res.*, vol. 6, pp. 363–392, Apr. 2005.

[30] C.-T. Chu *et al.*, "Map-reduce for machine learning on multicore," in *Proc. 20th Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2006, pp. 281–288.

[31] M. Zaharia *et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proc. 9th USENIX Conf. Netw. Syst. Design Implement. (NSDI)*, 2012, p. 2.

[32] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, 2010, p. 10.

[33] K. A. Kumar, J. Gluck, A. Deshpande, and J. Lin, "Hone: 'Scaling down' Hadoop on shared-memory systems," *Proc. VLDB Endowment*, vol. 6, no. 12, pp. 1354–1357, 2013.

[34] M. Ghanavati, R. K. Wong, F. Chen, Y. Wang, and C.-S. Perng, "An effective integrated method for learning big imbalanced data," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2014, pp. 691–698.

[35] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Oct. 2002.

[36] A. K. Baughman, W. Chuang, K. R. Dixon, Z. Benz, and J. Basilico, "DeepQA jeopardy! gamification: A machine-learning perspective," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 1, pp. 55–66, Mar. 2014.

[37] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[38] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.

[39] J. Fan, F. Han, and H. Liu, "Challenges of big data analysis," *Nat. Sci. Rev.*, vol. 1, no. 2, pp. 293–314, 2014.

[40] B. Ratner, *Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis of Big Data*. Boca Raton, FL: CRC Press, 2011.

[41] M. Y. Kiang, "A comparative assessment of classification methods," *Decision Support Syst.*, vol. 35, no. 4, pp. 441–454, Jul. 2003.

[42] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, vol. 13. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[43] O. J. Dunn, "Multiple comparisons among means," *J. Amer. Statist. Assoc.*, vol. 56, no. 293, pp. 52–64, 1961.

[44] C. S. Calude and G. Longo, "The deluge of spurious correlations in big data," *Found. Sci.*, pp. 1–18, Mar. 2016.

[45] P. Domingos, "A unified bias-variance decomposition and its applications," in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 231–238.

[46] J. Franklin, "The elements of statistical learning: Data mining, inference and prediction," *Math. Intell.*, vol. 27, no. 2, pp. 83–85, 2005.

[47] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores," *J. Cloud Comput., Adv., Syst. Appl.*, vol. 2, no. 1, p. 22, 2013.

[48] Y. Zheng, "Methodologies for cross-domain data fusion: An overview," *IEEE Trans. Big Data*, vol. 1, no. 1, pp. 16–34, Mar. 2015.

[49] M. Swan, "The quantified self: Fundamental disruption in big data science and biological discovery," *Big Data*, vol. 1, no. 2, pp. 85–99, 2013.

[50] X. Geng and K. Smith-Miles, "Incremental learning," in *Encyclopedia Biometrics*. New York, NY, USA: Springer, 2009, pp. 731–735.

[51] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental learning for ν-support vector regression," *Neural Netw.*, vol. 67, pp. 50–140, Jul. 2015.

[52] N. Marz. (2014). *Apache Storm*. [Online]. Available: http://storm.apache.org

[53] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, "S4: Distributed stream computing platform," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2010, pp. 170–177.

[54] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, 2014.

[55] A. Tsymbal, "The problem of concept drift: Definitions and related work," Dept. Comput. Sci., Trinity College Dublin, Dublin, Ireland, Tech. Rep. TCD-CS-2004-15, 2004, vol. 106.

[56] K. Grolinger, A. L'Heureux, M. A. M. Capretz, and L. Seewald, "Energy forecasting for event venues: Big data and prediction accuracy," *Energy Buildings*, vol. 112, pp. 222–233, Jan. 2016.

[57] P. B. Dongre and L. G. Malik, "A review on real time data stream classification and adapting to various concept drift scenarios," in *Proc. IEEE Int. Adv. Comput. Conf. (IACC)*, Feb. 2014, pp. 533–537.
[58] J. D. D. Lavaire, A. Singh, M. Yousef, S. Singh, and X. Yue, "Dimensional scalability of supervised and unsupervised concept drift detection: An empirical study," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2015, pp. 2212–2218.
[59] A. Clauset, "A brief primer on probability distributions," Santa Fe Inst., Santa Fe, NM, USA, Tech. Rep. CSCI 7000-001, 2011.
[60] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.
[61] M. Dundar, B. Krishnapuram, J. Bi, and R. B. Rao, "Learning classifiers when the training data is not IID," in *Proc. 20th Int. joint Conf. Artif. Intell. (IJCAI)*, 2007, pp. 756–761.
[62] J. Wang, D. Crawl, S. Purawat, M. Nguyen, and I. Altintas, "Big data provenance: Challenges, state of the art and opportunities," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2015, pp. 2509–2516.
[63] P. Buneman, S. Khanna, and W.-C. Tan, "Data provenance: Some basic issues," in *FST TCS 2000: Foundations of Software Technology and Theoretical Computer Science*. Berlin, Germany: Springer, 2000, pp. 87–93.
[64] H. Park, R. Ikeda, and J. Widom, "RAMP: A system for capturing and tracing provenance in MapReduce workflows," *Proc. VLDB Endowment*, vol. 4, no. 12, pp. 1351–1354, 2011.
[65] N. Cao, L. Lu, Y.-R. Lin, F. Wang, and Z. Wen, "SocialHelix: Visual analysis of sentiment divergence in social media," *J. Vis.*, vol. 18, no. 2, pp. 221–235, May 2015.
[66] M. Schroeck, R. Shockley, J. Smart, D. Romero-Morales, and P. Tufano, "Analytics: The real-world use of big data," IBM Global Business Services Saïd Business School Univ. Oxford, Tech. Rep. GBE03519-USEN-00, 2012, pp. 1–20.
[67] R. Lovelace, M. Birkin, P. Cross, and M. Clarke, "From big noise to big data: Toward the verification of large data sets for understanding regional retail flows," *Geogr. Anal.*, vol. 48, no. 1, pp. 59–81, Jan. 2016.
[68] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA: Morgan Kaufmann, 2016.
[69] R. Barga, V. Fontama, and W. H. Tok, "Cortana analytics," in *Predictive Analytics With Microsoft Azure Machine Learning*. Berkeley, CA, USA: Apress, 2015, pp. 279–283.
[70] Google. (2016). *Google Cloud Machine Learning*. [Online]. Available: https://cloud.google.com/products/machine-learning/
[71] Amazon Web Services. (2016). *Amazon Machine Learning*. [Online]. Available: https://aws.amazon.com/machine-learning/
[72] IBM. (2014). *IBM Watson Ecosystem Program*. [Online]. Available: http://www-03.ibm.com/innovation/us/watson/
[73] R Core Team, *R: A Language and Environment for Statistical Computing*, Vienna, Austria, 2015, vol. 1.
[74] *MATLAB*, The MathWorks Inc., Natick, MA, USA, 2016.
[75] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
[76] A. Labrinidis and H. V. Jagadish, "Challenges and opportunities with big data," *Proc. VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, 2012.
[77] C. L. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inf. Sci.*, vol. 275, pp. 314–347, Aug. 2014.
[78] E. Bingham and H. Mannila, "Random projection in dimensionality reduction," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2001, pp. 245–250.
[79] H. Liu and H. Motoda, *Instance Selection and Construction for Data Mining*, vol. 608. New York, NY, USA: Springer, 2013.
[80] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.
[81] NVIDIA. (2016). *GPU Applications: Transforming Computational Research and Engineering*. [Online]. Available: http://www.nvidia.ca/object/machine-learning.html
[82] G. S. Jedhe, A. Ramamoorthy, and K. Varghese, "A scalable high throughput firewall in FPGA," in *Proc. 16th IEEE Symp. Field-Programm. Custom Comput. Mach. (FCCM)*, Apr. 2008, pp. 43–52.
[83] A. Gesmundo and N. Tomeh, "HadoopPerceptron: A toolkit for distributed perceptron training and prediction with MapReduce," in *Proc. Demonstrations 13th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2012, pp. 97–101.
[84] A. Ghoting, P. Kambadur, E. Pednault, and R. Kannan, "NIMBLE: A toolkit for the implementation of parallel data mining and machine learning algorithms on Mapreduce," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 334–342.
[85] Y. Bu, B. Howe, and M. D. Ernst, "HaLoop: Efficient iterative data processing on large clusters," *Proc. VLDB Endowment*, vol. 3, nos. 1–2, pp. 285–296, 2010.
[86] J. Ekanayake *et al.*, "Twister," in *Proc. 19th ACM Int. Symp. High Perform. Distrib. Comput. (HPDC)*, 2010, pp. 810–818.
[87] G. Malewicz *et al.*, "Pregel: A system for large-scale graph processing," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2010, pp. 135–146.
[88] Apache. (2016). *Apache Giraph*. [Online]. Available: http://giraph.apache.org
[89] M. Gorawski, A. Gorawska, and K. Pasterak, "A survey of data stream processing tools," in *Information Sciences and Systems*. Cham, Switzerland: Springer, 2014, pp. 295–303.
[90] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," *ACM Comput. Surv.*, vol. 44, no. 3, p. 15, 2012.
[91] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for SVM," *Math. Programm.*, vol. 127, no. 1, pp. 3–30, Oct. 2010.
[92] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.
[93] Apache. (2016). *Apache Mahout*. [Online]. Available: http://mahout.apache.org
[94] 0xdata. (2016). *H2O*. [Online]. Available: http://www.h2o.ai
[95] E. P. Xing *et al.*, "Petuum: A new platform for distributed machine learning on big data," *IEEE Trans. Big Data*, vol. 1, no. 2, pp. 49–67, Jun. 2015.
[96] J. Langford, L. Li, and A. Strehl. (2007). *Vowpal Wabbit Online Learning Project*. [Online]. Available: http://hunch.net/~vw/
[97] G. De Francisci Morales and A. Bifet, "SAMOA: Scalable advanced massive online analysis," *J. Mach. Learn. Res.*, vol. 16, pp. 149–153, 2015.
[98] M. Abadi *et al.* (Mar. 2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1603.04467
[99] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 5, pp. 1060–1089, May 2013.
[100] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[101] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[102] Q. V. Le. (2015). A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. [Online]. Available: https://cs.stanford.edu/~quoc/tutorial2.pdf
[103] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2009, pp. 448–455.
[104] G. Hinton, "Deep belief nets," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA, USA: Springer, 2010, pp. 267–269.
[105] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," *CoRR*, vol. abs/1510.0, 2015.
[106] J. Read, F. Perez-Cruz, and A. Bifet, "Deep learning in partially-labeled data streams," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, 2015, pp. 954–959.
[107] B. Mirza, Z. Lin, and N. Liu, "Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift," *Neurocomputing*, vol. 149, pp. 316–329, Feb. 2015.
[108] K. Kanoun and M. van der Schaar, "Big-data streaming applications scheduling with online learning and concept drift detection," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, 2015, pp. 1547–1550.
[109] L. Bottou and V. Vapnik, "Local Learning Algorithms," *Neural Comput.*, vol. 4, no. 6, pp. 888–900, 1992.

[110] K. Huang, H. Yang, I. King, and M. R. Lyu, "Local learning vs. global learning: An introduction to maxi-min margin machine," in *Support Vector Machines: Theory and Applications*. Berlin, Germany: Springer, 2005, pp. 113–131.
[111] T.-N. Do and F. Poulet, "Random local SVMs for classifying large datasets," in *Future Data and Security Engineering SE-1*, vol. 9446. Cham, Switzerland: Springer, 2015, pp. 3–15.
[112] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 4, pp. 438–447, Jul. 2010.
[113] K. Grolinger, M. A. M. Capretz, and L. Seewald, "Energy consumption prediction with big data: Balancing prediction accuracy and computational resources," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Jun. 2016, pp. 157–164.
[114] L. Torrey and J. Shavlik, *Handbook of Research on Machine Learning Applications and Trends*. Hershey, PA: IGI Global, 2010.
[115] L. Yang, Y. Chu, J. Zhang, L. Xia, Z. Wang, and K.-L. Tan, "Transfer learning over big data," in *Proc. 10th Int. Conf. Digit. Inf. Manage. (ICDIM)*, Oct. 2015, pp. 63–68.
[116] S. Thrun and L. Pratt, *Learning to Learn*. Norwell, MA: Kluwer, 1998.
[117] D. L. Silver, Q. Yang, and L. Li, "Lifelong machine learning systems: Beyond learning algorithms," in *Proc. AAAI Spring Symp.*, 2013, pp. 49–55.
[118] M. T. Khan, M. Durrani, S. Khalid, and F. Aziz, "Lifelong aspect extraction from big data: Knowledge engineering," *Complex Adapt. Syst. Model.*, vol. 4, no. 1, pp. 1–15, 2016.
[119] Z. Chen and B. Liu, "Topic modeling using topics from many domains, lifelong learning and big data," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 703–711.
[120] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 4, pp. 70–73, 2014.
[121] T. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, vol. 1857. London, U.K.: Springer-Verlag, 2000, pp. 1–15.
[122] M. Sewell, "Ensemble learning," Dept. Comput. Sci., UCL, London, U.K., Tech. Rep. RN/11/02, 2011, p. 12.
[123] B. Waske and J. A. Benediktsson, "Fusion of support vector machines for classification of multisensor data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3858–3866, Dec. 2007.
[124] Y. Tang, Z. Xu, and Y. Zhuang, "Bayesian network structure learning from big data: A reservoir sampling based ensemble method," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, Dallas, TX, USA, Apr. 2016, pp. 209–222.
[125] W. Zang, P. Zhang, C. Zhou, and L. Guo, "Comparative study between incremental and ensemble learning on data streams: Case study," *J. Big Data*, vol. 1, no. 1, p. 5, 2014.
[126] R. Alabdulrahman, "A comparative study of ensemble active learning," M.S. thesis, Sch. Elect. Eng. Comput. Sci., Univ. Ottawa, Ottawa, ON, Canada, 2014.
[127] S. Gruber, R. W. Logan, I. Jarrín, S. Monge, and M. A. Hernán, "Ensemble learning of inverse probability weights for marginal structural modeling in large observational datasets," *Statist. Med.*, vol. 34, no. 1, pp. 106–117, Jan. 2015.
[128] D. E. Bachmann, "Contextual model-based collaborative filtering for recommender systems," M.S. thesis, Dept. Elect. Comput. Eng, Univ. Western Ontario, London, ON, Canada, 2017.
[129] Y. Wu *et al.* (Sep. 2016). "Google's neural machine translation system: Bridging the gap between human and machine translation." [Online]. Available: https://arxiv.org/abs/1609.08144
[130] R. Saeedi, H. Ghasemzadeh, and A. H. Gebremedhin, "Transfer learning algorithms for autonomous reconfiguration of wearable systems," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 563–569.

ALEXANDRA L'HEUREUX received the B.E.Sc. and M.E.Sc. degrees in software engineering from Western University, Canada, in 2013 and 2015, respectively, where she is currently pursuing the Ph.D. degree. Her current research interests include big data, machine learning, data analytics, and concept drift.

KATARINA GROLINGER received the B.Sc. and M.Sc. degrees in mechanical engineering from the University of Zagreb, Croatia, and the M.Eng. and Ph.D. degrees in software engineering from Western University, Canada. She is currently an Assistant Professor with Western University, where she is also a Post-Doctoral Fellow. She is also a Certified Oracle Database Administrator with over ten years of industry experience in database administration and software development. Her research interests include NoSQL data stores, big data management, Internet of Things, data analytics, and cloud computing.

HANY F. ELYAMANY (M'06) received the B.Sc. degree in computational sciences from Suez Canal University, Ismailia, Egypt; the M.Sc. degree in computer science from Ain Shams University, Egypt; and the Ph.D. degree in software engineering from Western University, Canada. He has been involved in the software engineering area in academia and industry for 20 years. He is currently a Post-Doctoral Fellow with Western University. He is also an Associate Professor with the Computer Science Department, Suez Canal University. His research interests include service-oriented architecture, data mining, cloud computing, and security.

MIRIAM A. M. CAPRETZ received the B.Sc. and M.E.Sc. degrees from UNICAMP, Brazil, and the Ph.D. degree from the University of Durham, U.K. She has been involved in the software engineering area for over 35 years. She was with the University of Aizu, Japan. She is currently a Professor with the Department of Electrical and Computer Engineering, Western University, Canada. Her current research interests include cloud computing, big data, service oriented architecture, and privacy and security. She has been involved with the organization of workshops and symposia and has been serving on program committees in international conferences.