International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 5 (2018) pp. 2753-2762
© Research India Publications. http://www.ripublication.com

Machine Learning for Big Data: A New Perspective

Suryabhan Pratap Singh
Research Scholar, Department of Computer Science and Engineering,
Madan Mohan Malaviya University of Technology, Gorakhpur-273010, Uttar Pradesh, India.
spsuryabhan@gmail.com

Umesh Chandra Jaiswal
Professor, Department of Computer Science and Engineering,
Madan Mohan Malaviya University of Technology, Gorakhpur-273010, Uttar Pradesh, India.
ucj_jaiswal@yahoo.com

Abstract

The big data are now the topic of rapidly growing research interest in all computer research domains. Learning techniques specifically convey crucial opportunities and transformative potential in several areas of thinking and innovative learning methods to address a number of issues. This paper reviews the recent advancement in machine learning for big data. Further, we discuss novel machine learning methods, then specifically emphasize on recently proposed learning techniques, for instance representation learning, transfer learning, deep learning, and active learning. Analysis and discussion of the existing machine learning tools, associated challenges and the solutions proposed are also presented and compared specifically with the perspective of big data.

Keywords - Big Data, Machine Learning, Data Mining, Learning Methods and Tools.

INTRODUCTION

Undoubtedly, Big data[1] is the motivating and rapidly changing research areas which attracted sound attention from academia, industry, and government. The potential of changing the real world force the researchers to come up with advanced learning methods to overcome the current alarming issues. Big Data is data having complexity, scalability, and diversity. It requires new techniques, architectures, analytics, and algorithms to manage hidden knowledge and extract value from it. Big data analytics[2]comprise the process of collecting, organizing and analysing big data. It examines large datasets having a variety of data types i.e., big data, to disclose hidden patterns, customer preferences, market trends, unknown correlations and other useful business information.

Machine learning is an area of research properly emphasises on the concept, presentation, theory properties, performance of learning algorithms as well as systems. Machine learning methods have been broadly recognized in several huge as well as a composite data-intensive area, for instance, astronomy, biology, medicine, etc. These methods afford conceivable remedies to pit the facts concealed in the data. Big data, the gathering of datasets is so huge and composite that it is tough to a pact with by means of out-dated learning techniques. Meanwhile, the conventional method of learning as of predictable datasets was not considered to and will not effort fine with huge sizes of data. Machine Learning algorithm is incorporated for the processing of high volume of data. Operative machine learning is tough because sufficient training data is not available so finding patterns is stiff.

LITERATURE REVIEW

Machine Learning is an interdisciplinary field related to more than one branch of knowledge. Machine learning has concealed more or less every scientific domain[3]. Artificial intelligence, optimal control, cognitive science, statistics, information theory, optimization theory, and many other disciplines of mathematics, engineering, and science are few fields in which machine learning has an extensive variety of applications[4][5]. Machine learning techniques have grown radically over past two decades. A learning problem can be defined as when executing some task, with the help of some type of training datasets for refining measures of performance. An innumerable range of machine learning methods has been developed to find a function or program that improves performance metric. It has been used for data mining, recognition systems, recommendation engines, informatics and so many varieties of problems[6]. Reinforcement learning, unsupervised learning, and supervised learning are subdomains of machine learning[7].

Supervised Learning approach involves training with inferring a function from labeled data which has inputs and preferred outcomes. It is the machine learning approach to extracting a function from labeled training dataset[8]. The training dataset comprises a set of training samples. Every sample comprises an input data and the desired outcome value[9]. Data-Processing Task: - Classification, Estimation, Regression.

Table 1. Supervised Learning Algorithms

Distinction norm	Learning Algorithm
Computational classifier	Support Vector Machine
Statistical classifier	Bayesian networks, Naïve Bayes, Hidden Markov Model
Connectionist classifier	Neural networks

Unsupervised Learning method uses to draw inferences from data set and does not require labeled training data. It is machine learning algorithm concluding a method to label hidden erection from "unlabeled" data[10]. Also, the environs only afford inputs without preferred targets, unlike the supervised learning. Data-Processing Tasks: Clustering/Predictions

Table 2. Unsupervised Learning Algorithms

Distinction norm	Learning Algorithm
Parametric	Gaussian mixture model, K-means
Non Parametric	X-means, Dirichlet process mixture model

Reinforcement Learning[11]is a method of Machine Learning that permits software and machines to automatically identify the ideal behaviour within a precise framework, in order to maximize its performance. It enables training from the response received through interfaces without environs. Data-Processing Tasks: Decision-Making.

Table 3. Reinforcement Learning Algorithms

Distinction norm	Learning Algorithm
Model_free	R_learning, Q_learning
Model_based	Sarsa learning, TD learning

A. Modern Learning Techniques

Some modern learning techniques are greatly desirable for resolving the big data problem.

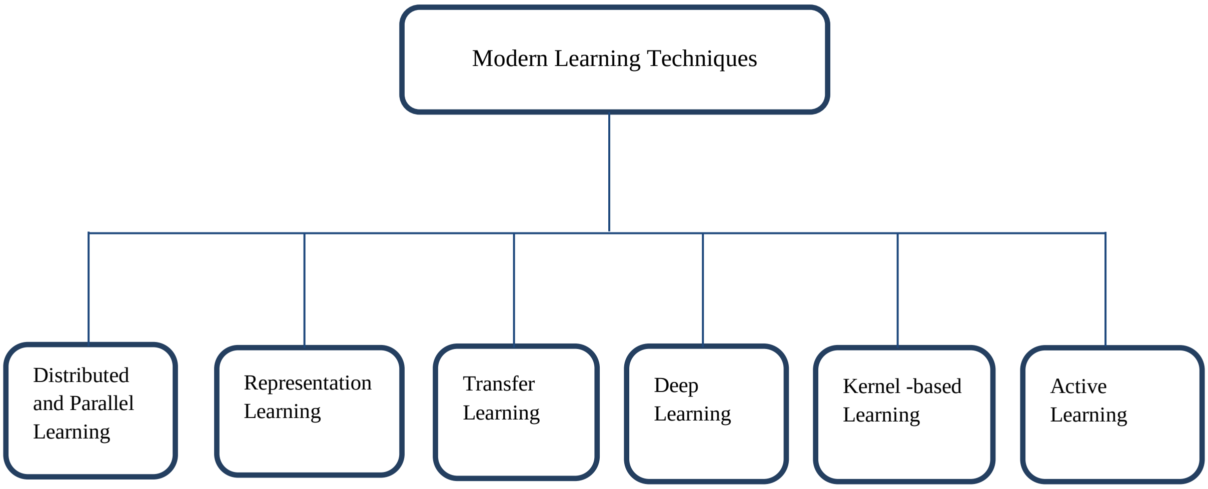

Figure 1. Modern learning techniques

1) *Distributed and Parallel Learning:* For preventing barriers such a large consecration is the incapability of training techniques for datasets to train within a sensible time. For this purpose, Distributed and Parallel learning[12] performs towards an auspicious research since assigning the learning procedure between a number of workspaces are an usual approach of scaling up training or learning techniques[13].Distributed and Parallel learning method to building such condition is expected at solving the real-world problem of how to learn from big data sets. A number of popular distributed and parallel machine learning techniques have been proposed, few of them are meta-learning[14], distributed boosting[15], decision rules[16], and stacked generalization[17]. These methods have now days become broadly reachable and accessible by the influence of multi-core processors, distributed and parallel computing, and cloud computing platforms[18]. The main idea is that the parallel and distributed learning algorithm should concentrate on the instances that are difficult to learn.

2) *Representation Learning:* Since the early 21st-century representation for word learning techniques uses neural network based algorithms for studies [19]–[22]. Representation learning learns the expressive, valuable and meaningful representations of the data. The performance of machine learning techniques is greatly reliant on the excellence of data representation to which they are applied[23]. It is at ease to excerpt valuable facts while building predictors or extra classifiers. It objective to reach that sensibly sized learned representation can detention the vast amount of conceivable effort outlines. This can significantly enable enhancements in equally statistical and computational competence. There are primarily few sub parts of representation learning: distance metric learning, feature extraction, and feature selection[24]. Real-world applications, such as intelligent vehicle systems, object recognition, natural language processing, signal processing and speech recognition[25]–[27]. A noble representation is one that makes a consequent learning action easier. The choice of representation usually depends on the choice of the consequent learning framework. Designing deep architectures is easier with the help of notation of representation.

3) *Transfer Learning:* Transfer learning is the capability of an erudition process to deed commonalities between dissimilar learning methods to share transfer knowledge and statistical strength across tasks. In transfer learning information obtained in one learning task is used to advance enactment of another related task. The facts detonation as of the diversity of sources, excessive heterogeneity of composed data abolishes the assumption. Although, a most important postulation in various out-dated machine learning techniques are those test and training data are taken from the similar feature space as well as have similar circulation. Transfer learning methods have been proposed to tackle this issue by permitting the distributions, tasks, and domains designate diverse, which can excerpt information from either one or more than one source tasks also smear information towards an objective task[28], [29]. The benefit of transfer learning technique is to solve new problems faster. It can perceptively apply information learned earlier. Transfer learning procedures require beings measured effectively in constructing useful priors, extensive manuscript classification, and cross-domain text classification. There are three sub parts of transfer learning; unsupervised, inductive, and Trans_ductive[29]. In terms of unsupervised transfer learning, the objective task is dissimilar from however associated with source framework. Currently, in inductive learning, source and objective frameworks are dissimilar, even no issue about the source and objective fields that are identical or not. In this setting, a slight quantity of labeled test dataset is used laterally with labeled training dataset. Trans_ductive transfer learning, the objective dominion is dissimilar from the source field, whiles the objective frameworks, as well as source frameworks, are similar. A significant quantity of un-labeled testing samples is taken along with the labeled training data. In many real-world applications, for example, constructing informative priors, large-scale document classification, cross-domain text classification uses transfer learning algorithms [30]–[32].

4) *Deep Learning:* In distinction to utmost out-dated learning methods; those are deliberated using shallow-structured learning manners. Deep learning attempts to model various levels of abstraction within data. There are various tools to train and deploy deep networks, and Intel is actively working with the deep learning community to optimize many of the frameworks to significantly improve computational performance on Intel® architecture. Objectives of deep learning to learn the parameter of the transformations that minimize a cost function. Deep learning primarily takes either or both unsupervised and supervised approaches in deep manners towards robotically absorb top-down or hierarchical representations[33]. It delivers extrapolative computational analysis remedies for big datasets. Mostly with the advances in graphics processors and the improved processing power[34]. Deep learning techniques transmute their inputs concluded extra layers than shallow-learning techniques. At every level, the signal is transmuted by a processing item, similar to an artificial-neuron, whose parameters are learned with training. Two core streams of deep learning techniques are Convolutional-Neural-Network[35], [36] and Deep-Belief Network[33], [37]. Applications of deep learning are natural language processing, information retrieval, speech recognition, and computer vision[33], [38]–[40].

5) Kernel-based Learning: In machine learning, kernel-based learning has become been popular from last few years. It has a stronger mathematical gradient than other machine learning techniques[41]. Kernel-based machine learning is an alternative machine learning where in place of considering oodles of features, we consider one kernel function to extract similarity between objects or images. We use kernel function combining with the labels and images to learning method and get target output as a classifier. The outstanding benefit of kernel-based learning techniques are their graceful characteristics of subliminally representing a trial group from actual space into a possibly vast dimensional feature space[42]. With the used kernel, function method calculation of internal products can be direct. The kernel-based learning technique corresponds to a dot product in a feature space, but it can formulate everything in terms of kernel evaluations. Without knowing the mapping explicitly the kernel-based trick makes available a well-designed mathematical worth to paradigm influential non-linear variants of extreme renowned statistical linear methods. The maximum extensively used kernel-based functions contain Polynomial kernels and Gaussian kernels. Many challenging applications developed in Kernel-based learning[43], for instance adaptive multi regression[44], convexly constrained parameter/function estimation[45], online classification[46], and beam forming problems[47].

6) Active Learning: Active learning is an extraordinary situation of semi-supervised machine learning in which a learning technique is accomplished to interactively probe the learner or some other data and information source to acquire the preferred outcomes at new data points. There can be arises such a circumstances that data might be rich but the label is rare or exclusive to obtain in many real-world applications. Regularly, learning from huge quantities of unlabelled data is time taking and tough process. Active learning techniques have an effort to solve this concern by choosing a subgroup of utmost precarious cases for labeling[48]. The active learner has an objective to accomplish exactness using as rare labeled cases as conceivable, thus diminishing the cost of tracking down labeled data[49]. It can acquire reasonable classification performance with scarcer labeled trials through probe approaches than those of predictable active learning[50]. We have three key active learning consequences, flood-based selective sampling, comprising membership request processing, and pool-based sampling[49]. In the field of machine learning these have been studied broadly as well as applied to various data processing issues like biological DNA identification[51], [52]as well as image classification. In statistics collected works active learning is occasionally also called optimal experimental design.

B. Machine Learning Tools for Big Data

1) *Hadoop Ecosystem*: Hadoop is an open source software framework for storing data and running applications on clusters of commodity machine having distributed file system (DFS)[53]. It has grown into an enormous web of projects related to each step of a big data workflow. Hadoop includes data collection, storage, processing, and much more. The expanse of frameworks that have been developed to either one replace or complement these novel elements has made the recent description of Hadoop vague. To abundantly realize Hadoop, we need to consider both the framework itself and the environment that supports it[54]. The Hadoop framework itself presently involves some modules[55].

Hadoop Common: It refers to the collection of Java libraries and utilities required by other Hadoop segments[56]. It is a crucial module of the Apache Hadoop Framework, along with the Hadoop Distributed File System (HDFS), Hadoop Map Reduce and Hadoop YARN. Hadoop Core is another name of Hadoop Common.

Hadoop distributed file system (HDFS): A file outline deliberate to hoard huge quantities of data through several knobs of commodity machine or hardware[57]. It affords high throughput access to application data. This scheme possesses integral fault tolerance capabilities. It typically maintains three or more copies of each data block in case of disk failure. HDFS has design goals same as Google File System[58]. Both target at data concentrated computing requests where enormous data files are communal and get enhanced in errand of extraordinary continuous bandwidths in spite of less potential or latency, for enhanced maintain batch-processing style workloads.

Hadoop YARN is a cluster managing technique, which is used for cluster resource management and job scheduling[59]. It is most important features in the second-generation Hadoop 2.0 version of the Apache Software Foundation's open source distributed processing framework. YARN permits a more generalized Hadoop that makes Map Reduce just one type of YARN application request. It can be left out completely in favor for various processing engine. It has instinctive collective libraries that comprise Java applications for compression error detection, I/O utilities, and codes.

Hadoop Map Reduce: Map Reduce is YARN based system for parallel processing for a huge amount of datasets[60]. A Map Reduce system contains two phases. First map phase, it takes raw data and converts it into a set of data, where individual elements are broken down into tuples know as key/value pairs. Second, reduce phase which combines those data tuples into a smaller set of tuples and processes data in parallel.

2) *Map Reduce*: Based on the LISP map and reduce primitives, Map Reduce is a software framework for applications. It processes a huge quantity of data in parallel on the large cluster (millions of nodes) of commodity hardware[54]. This entire process must be done in a reliable as well as fault tolerant way. An associated implementation of Map Reduce having parallel programming model is introduced by Google[61]. Map Reduce is used in massive data analysis such as data mining, data analytics. It is continually being explored on efficiency, performance and various parameters[62]. Map Reduce programming actually does not deal with communication between nodes, and distribution of tasks. It refers to writing a Map task function and Reduce task function[63]. These functions are used by Hadoop program. Several Map task functions can be executed simultaneously. It takes key value pair as a source data and generates output as a list of intermediate values with its key. This is a part of the procedure that divides tasks. The output of the Map-task function is taken by Reduce-task function as an input data. It performs a certain process on them and combines values to produce the preferred outcome in an output file.

Following steps are involved in Map Reduce:

i. The Map Reduce function first splits input data into X_m tuples of usually 2^{14} kilobytes to 2^{16} kilobytes (KB) per tuple. After that, it initiates several replicas of the function on a knot of systems[64].

ii. One of the replicas of the function is special- the master replica and remaining is working nodes. Master assigns a task to all working nodes. There are X_m Map-task functions and X_r Reduce-task functions to allocate. The master selects and allocates a task to all idle nodes.

iii. A node who is allocated a map task function read out the contents of the consistent input tuple split. It analyzes key/value pairs tuple available input data and permits every pair to the user-defined Map function. The intermediary key/value pairs tuple twisted by the Map function are buffered in memory.

iv. Sporadically, the buffered sets are written to native disk separated into X_r areas by separating function. The positions of these buffered sets on native disk are returned to the master. Master is liable for progress these positions and forwarding to reduce nodes.

v. Whenever a reduce node is reformed by the master about these positions. Reduce node uses separate procedure requests to read buffered data from the native disk of map nodes. When all intermediate data has been read by reduce node then it will be sorted by the intermediary keys. There are normally several dissimilar key map to similar reduce task so sorting is required.

vi. Again the sorted intermediate value is passed to reduce function. Further, Reduced function's output is attached to the final output file.

vii. When all map task and reduce task have been completed the Amp-Reduce call in the user function returns to the user code.

3) *APACHE HIVE*: HIVE [58] is a data warehouse organizing tool that resides on top of Hadoop for processing data query, analysis, and summarization and specifically used to process structured data[65]. Hive provides a SQL-type interface to query data stowed in different file systems and databases that assimilate with Hadoop. Hive provides an essential SQL abstraction to assimilate SQL-type Queries without implementing queries in the low-level Java API. It also provides a SQL-type language called Hive Query Language (HQL). It is designed for Online Transaction Processing (OLAP). It is familiar, scalable, fast, and extensible.

4) *APACHE PIG*: Generally used with Hadoop, Pig[65] is an abstraction over Apache Spark[53], Tez or Map Reduce. Further, APACHE PIG provides a high-level platform to create programs run on Apache Hadoop[58]. To write data analysis function for this platform, a high-level language known as Pig Latin is provided by Pig. Manipulation operations on data in Hadoop can be implemented using Pig Latin. It also allows writing a data flow program by which data can be transformed. Pig Latin provides several operators to perform an operation such as sort, filter, join, and many other operations. Pig Latin language provides various operators to the user for developing their own function for reading, writing, and processing data. Users can extend Pig Latin by using Ruby, Python, Java, or other scripting languages. 'Pig' command and 'java' are two ways used to run Pig.

Map Reduce Mode- Map Reduce mode is the default mode for accessing Hadoop cluster.

Local Mode- All files are installed and run using file system and a local host having access to the single machine in Local Mode.

5) HBase: It is distributed database which is developed to board organized data in tables that could have billions of row and millions of columns.

6) HCatalog: HCatalog is a storage management layer for Hadoop which stores data in table format. It supports diverse modules existing in Hadoop like Hive, Map Reduce, and Pig which helps easily read and write data from the knot. HCatalog provides visibility for data archiving and cleaning tools. It also supports different types of file formats such as RC File, ORC, CSV, JSON file formats.

C. Critical Problems of Machine Learning for Big Data

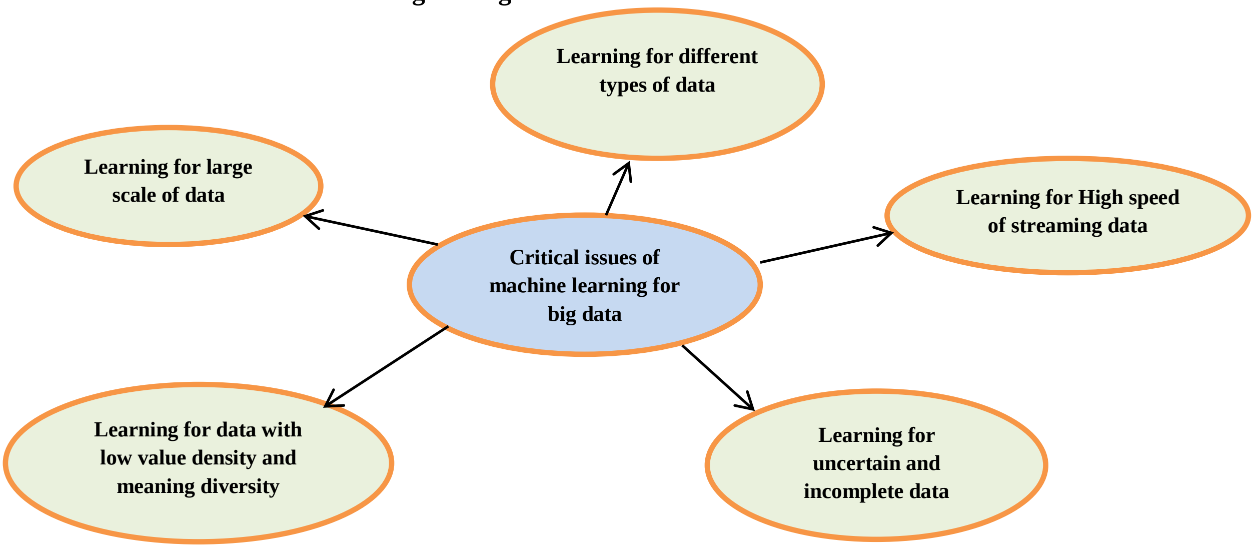
Learning for different types of data
Learning for large scale of data
Learning for High speed of streaming data
Critical issues of machine learning for big data
Learning for data with low value density and meaning diversity
Learning for uncertain and incomplete data

Figure 2. Critical problems of machine learning for big data

1) Learning for large Scale of Data

Critical problems-This is clear that the data size or volume is a major quality of big data. Those benevolences major issue or defy for machine learning. The Large extent of facts flooding in from our phones, computers, trains, buses, planes, parking meters, and social sites. Taking only digital or social media data as an example, every day, approximately 24 petabytes (petabyte = 1024 × 1024 × 1024 × 1024 × 1024 bytes) of data Google alone required being processed[66]. Facebook procedures and processes up to one million photographs every moment. Prior estimations recommend that Facebook stowed 260 billion photos using storage space of more than 20 petabytes. In addition, if we keep attention the other data sources, the data scale will be very large. Beneath recent development tendencies, data collected by large organizations and analysed then it will definitely reach Petabyte to Exabyte (Exabyte = 1024 petabyte) or even more scale quickly.

Possible solution-We are swimming in the ocean of detail of the information and data, which is excessively big on the way to design and develop a machine learning technique by means of storage and a central processor. As an alternative, parallel computing with distributed outlines are favoured. Alternating direction methods of multipliers (ADMM's)[67], [68], helping by means of an encouraging computing outline for distributed, emerging ascendable, online-convex optimization techniques are suitable for distributed and parallel inclusive large-scale data processing. ADMM's core competence is the capability to decouple or divide many variables in optimization issues, so that a small scale global optimization problem may be solved, thereby reducing small sub-problems and sub-sub-problems. ADMMs are concurrent for arched optimization; nevertheless, there is a lack of concurrent and hypothetical enactment guarantee for non-convex optimization. Though, the vast experimental confirmation in the fiction provisions the experimental merging and noble enactment of ADMM[69]. For the large scale dataset of ADMM, machine learning problems have been discussed in a variety of applications[69].

Cloud computing supported learning technique is an alternative effective evolution which is considered to deal with the challenge of the volume of large data for the data systems. It has been already validated appreciable flexibility by cloud computing[70], [71] which allows hopes to achieve the necessary scalability for machine-learning techniques. It will improve storage capacity and compute over cloud computing infrastructure. In this perspective, an outline for machine learning in distributed Graph_Lab and cloud has been implemented [72].

In addition to distributing hypothetical context for machine-learning to reduce the challenges associated with the large volume, certain real-world parallel programming techniques have also been implemented on learning techniques to deal with big data sets. Map Reduce[73], [74] is a powerful programming framework which has great fault tolerance capability important for dealing with large data sets. It also enables the distribution and automatic paralleling of computational applications on huge groups of service machines. Map Reduce is an outline for dealing out processing parallelizable issues across huge data-sets by means of the cluster or grid handling can arise on data stored either one in a database or in a file system. Map Reduce can proceeds advantage and benefit in the vicinity of data processing it nearby the place[75].It is stowed in mandate to minimize communication overhead. The main idea of the Map Reduce is to first split large data into small fragments and then assign these fragments in distributed and parallel method to generate intermediary consequences. With gathering entirely intermediary consequences, the conclusive outcome is generated. A collective resource of programming-machine-learning techniques on multi-core has been examined through the benefit of the Map Reduce.

2) Learning for Different Types of Data

Critical problem- Huge variation in data is another quality of big data that makes it interesting as well as challenging. Structured, semi-structured, and unstructured data generally come from several sources are combine and generate non-linear, heterogeneous, and high dimensional data with various representation forms. Learning with this type of data set, the degree of complexity is not even conceivable before reaching here.

Possible solution- Data integration technique use to combine data residing at various sources for heterogeneous data[76], [77]. This is a vital technique that provides a united view of this type of data to the user. It is an effective method to acquire noble data representations to every single data input. It can also be used to assimilate the erudite features at various levels. Deep learning and representation learning methods are useful for this type of problems. Non-linear and high dimensional are another challenges associated with a huge variety of data[78]. Human gene distributions, stellar spectra, and global climate pattern are few examples of this type of challenges[79].Dimensionality reduction is an efficient way to deal with high-dimensional data. It finds evocative low-dimensional degree structures obscured in its high-dimensional interpretations[80], [81].Collective methods are used to employ extraction as well as feature selection to diminish the data dimensions. Transfer learning is a noble method for this type of problems.

3) Learning for Fervid of Flooding Data Speed

Critical problem- Velocity or speed truly matters for big data. It is one more emerging challenge for machine learning. The system has to complete a job in real-world applications within the definite time otherwise result become less valuable or even worthless. The potential worth of data rely on data novelty. Agent-based autonomous exchange systems, stock market prediction, and earthquake prediction are few examples. Data requires being handled in real-time in time-sensitive cases.

Possible solution- Online learning approaches are the best resolution for training from such fervid speed data. Online learning[82]–[85] is a well-established learning model. It has a methodology to learn a single instance at a time. On other hand batch or offline learning, approaches need to gather complete information about training dataset. Existing machines do not process the whole data set in its storage; thus, this sequential learning method glowing for big data. In comparison with some traditional learning technique such as extreme-learning machine[86] is an innovative machine learning technique for speedup learning methodology. Extreme learning machine has stout benefits in handling with a fervid velocity of data. It provides improved generalization performance, faster learning speed, and with least human interference[87]. Data distribution is varying over time and frequently non-stationary[34]. It is another challenging problem with the high velocity that requires training methods for recognizing data as a flood. The potential superiority of data flooding techniques and processing theory[88] have a goal to evaluate data rapidly to derive its outcomes as the effective solution for this problem.

4) Learning for Incomplete and Uncertain Data

Critical problem- In recent days, machine learning techniques usually nourished with comparatively precise data from quite limited as well as renowned sources. The learning outcomes have a tendency to be certain also. Thus, veracity has not ever been a stern problem for anxiety. The accuracy and faith of the input data rapidly become a problem for the stark size of data accessible today. The data inputs are coming from various different ancestries and data quality is not completely verifiable. Therefore, another critical problem with big data is veracity.

Possible solution- Ambiguous data are an exclusive kind of data related to probability or some random distributions. Data collections and readings are non-deterministic for ambiguous data. Data ambiguity is common in many applications. The dominant challenge is that the attribute or dataset feature for ambiguous data is apprehended by a unique value and also shown as sample distributions[89]. Summary statistics i.e. variances and means to abstract sample distributions are a simple manner to deal with data ambiguity. To construct a decision tree utilize the complete facts supported by probability distributions is another technique which is called distribution-based approach[90].

5) Learning for Data with Meaning Diversity and Low Value Density

Critical problem- There are a variety of training methods used to evaluate big data sets. The ultimate goal is to excerpt appreciated information from huge volumes of data in the form of salable benefits as well as deep perception. The value is also characterized as a leading big data feature[91], [92]. A small worth density is not straight forward to originate considerable value from massive volumes of data.

Possible solution-Data mining technologies, as well as knowledge discovery in databases, are useful to handle this type of challenges[89], [93], [94]. These types of techniques confer possible explanations to search the essential facts hidden in an enormous data[93]. Specifically, utilizing frequent patterns, classification, and clustering techniques to mine value from enormous data with the perspective of frames. The variety of data connotation related with evaluation of big data is also a challenging problem. In which the fiscal value of various data differs expressively, even the similar data have dissimilar value if acknowledge from various frameworks as well as perspectives. Some novel cognition-assisted training methods should be also developed to make existing training techniques further intelligent as well as flexible. It is predictable that the period of perceptive computing will come soon.

DISCUSSION AND CONCLUSION

Outdated machine learning methods are not scalable enough to handle the data having a large volume, high speed, varying types, low value density, and incompleteness and uncertainty. These alarming challenges force to develop new data processing to satisfy the rapidly changing needs of the end system/organization. Learning techniques specifically discussed in this paper convey a list of techniques, algorithms, methods that is an opportunity for the researchers to undergo. The big data are now the topic of rapidly growing interest in all computer science research domains. Analysis and discussion of the existing machine learning tools, associated challenges and the solutions proposed are also presented and compared specifically from the big data point of view. Advancement in machine learning for big data, novel machine learning methods, recently proposed learning techniques such as representation-learning, transfer-learning, deep-learning, active-learning etc. are discussed.

In future, an extensive literature survey is required specifically in the sub domains of the Machine Learning for Big Data. The need to address such issues is the result of the complexity involved with this area incorporating various other fields such as Database Systems, Artificial Intelligence, Soft Computing, Algorithm Design etc.

REFERENCES

[1] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, “Big data analytics on Apache Spark,” Int. J. Data Sci. Anal., vol. 1, no. 3–4, pp. 145–164, 2016.

[2] P. Russom, “Big data analytics,” TDWI best Pract. report, fourth Quart., vol. 19, p. 40, 2011.

[3] C. Rudin and K. L. Wagstaff, “Machine learning for science and society,” Mach. Learn., vol. 95, no. 1, pp. 1–9, 2014.

[4] Russell, S., Norvig, P., & Intelligence, A. (1995). A modern approach. Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, 25, 27.

[5] Mitchell, T. M. (2006). The discipline of machine learning (Vol. 3). Carnegie Mellon University, School of Computer Science, Machine Learning Department.

[6] C. M. Bishop, Pattern Recognition and Machine Learning, vol. 53, no. 9. 2013.

[7] Adam, B., & Smith, I. F. (2008). Active tensegrity: A control framework for an adaptive civil-engineering structure. Computers & Structures, 86(23), 2215-2223.

[8] S. Shalev-shwartz, Y. Singer, and A. Y. Ng, “Online and Batch Learning of Pseudo-Metrics,” Proc. 21st Int. Conf. Mach. Learn., pp. 1–8, 2004.

[9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). Overview of supervised learning. In The elements of

statistical learning(pp. 9-41). Springer New York.

[10] B. Fritzke, “Growing cell structures—A self-organizing network for unsupervised and supervised learning,” Neural Networks, vol. 7, no. 9, pp. 1441–1460, 1994.

[11] R. Giryes and M. Elad, “Reinforcement Learning: A Survey,” Eur. Signal Process. Conf., pp. 1475–1479, 2011.

[12] R. Bekkerman, M. Bilenko, and J. Langford,Scaling Up Machine Learning: Parallel and Distributed Approaches. 2012.

[13] D. Peteiro-Barral and B. Guijarro-Berdiñas, “A survey of methods for distributed machine learning,” Prog. Artif. Intell., vol. 2, no. 1, pp. 1–11, 2013.

[14] Leyva, E., Gonzalez, A., & Perez, R. (2015). A set of complexity measures designed for applying meta-learning to instance selection. *IEEE Transactions on Knowledge and Data Engineering*, *27*(2), 354-367.

[15] M. Sarnovsky and M. Vronc, “Distributed boosting algorithm for classification of text documents,” SAMI 2014 - IEEE 12th Int. Symp. Appl. Mach. Intell. Informatics, Proc., pp. 217–220, 2014.

[16] H. Chen, T. Li, C. Luo, S. J. Horng, and G. Wang, “A rough set-based method for updating decision rules on attribute values’ coarsening and refining,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 12, pp. 2886–2899, 2014.

[17] J. Chen, C. Wang, and R. Wang, “Using stacked generalization to combine SVMs in magnitude and shape feature spaces for classification of hyperspectral data,” IEEE Trans. Geosci. Remote Sens., vol. 47, no. 7, pp. 2193–2205, 2009.

[18] S. R. Upadhyaya, “Parallel approaches to machine learning - A comprehensive survey,” J. Parallel Distrib. Comput., vol. 73, no. 3, pp. 284–292, 2013.

[19] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A Neural Probabilistic Language Model,” J. Mach. Learn. Res., vol. 3, pp. 1137–1155, 2003.

[20] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” Proc. 25th Int. Conf. Mach. Learn., pp. 160–167, 2008.

[21] Turian, J., Ratinov, L., & Bengio, Y. (2010, July). Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 384-394). Association for Computational Linguistics.

[22] A. Mnih and G. E. Hinton, “A Scalable Hierarchical Distributed Language Model.,” Adv. Neural Inf. Process. Syst., pp. 1–8, 2008.

[23] 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, 2013.

[24] W. Tu and S. Sun, “Cross-domain representation-learning framework with combination of class-separate and domain-merge objectives,” Proc. 1st Int. Work. Cross Domain Knowl. Discov. Web Soc. Netw. Min. - CDKD ’12, pp. 18–25, 2012.

[25] K. Dwivedi, K. Biswaranjan, and A. Sethi, “Drowsy driver detection using representation learning,” Souvenir 2014 IEEE Int. Adv. Comput. Conf. IACC 2014, pp. 995–999, 2014.

[26] Ryynanen, M. P., & Klapuri, A. (2005, October). Polyphonic music transcription using note event modeling. In Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on (pp. 319-322). IEEE

[27] Bordes, A., Glorot, X., Weston, J., & Bengio, Y. (2012, March). Joint learning of words and meaning representations for open-text semantic parsing. In *Artificial Intelligence and Statistics* (pp. 127-135).

[28] E. W. Xiang, B. Cao, D. H. Hu, and Q. Yang, “Knowledge for Transfer Learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 6, pp. 770–783, 2010.

[29] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2010.

[30] X. Ling and G. Xue, “Spectral Domain-Transfer Learning,” pp. 488–496.

[31] R. Raina, A. Y. Ng, and D. Koller, “Constructing informative priors using transfer learning,” Proc. 23rd Int. Conf. Mach. Learn. - ICML ’06, pp. 713–720, 2006.

[32] J. Zhang, “Deep transfer learning via restricted Boltzmann machine for document classification,” Proc. - 10th Int. Conf. Mach. Learn. Appl. ICMLA 2011, vol. 1, pp. 323–326, 2011.

[33] D. Yu and L. Deng, “Deep Learning and Its Applications to Signal and Information Processing [Exploratory DSP],” Signal Process. Mag. IEEE, vol. 28, no. 1, pp. 145–154, 2011.

[34] Xue-Wen Chen and Xiaotong Lin, “Big Data Deep Learning: Challenges and Perspectives,” IEEE Access, vol. 2, pp. 514–525, 2014.

[35] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural Language Processing (Almost) from Scratch,” J. Mach. Learn. Res., vol. 12, pp. 2493–2537, 2011.

[36] P. Le Callet, C. Viard-Gaudin, and D. Barba, “A convolutional neural network approach for objective video quality assessment,” IEEE Trans. Neural Networks, vol. 17, no. 5, pp. 1316–1327, 2006.

[37] Bastien, F., Bengio, Y., Bergeron, A., Boulanger-Lewandowski, N., Breuel, T., Chherawala, Y., & Glorot, X. (2010). Deep self-taught learning for handwritten character recognition. *arXiv preprint arXiv:1009.3589*.

[38] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” IEEE Trans. Audio, Speech Lang. Process., vol. 20, no. 1, pp. 30–42, 2012.

[39] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82-97.

[40] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition," Neural Comput., vol. 22, no. 12, pp. 3207–3220, 2010.

[41] Ding, G., Wu, Q., Yao, Y. D., Wang, J., & Chen, Y. (2013). Kernel-based learning for statistical signal processing in cognitive radio networks: Theoretical foundations, example applications, and future directions. IEEE Signal Processing Magazine, 30(4), 126-136.

[42] C. Li, M. Georgiopoulos, and G. C. Anagnostopoulos, "A unifying framework for typical multitask multiple kernel learning problems," IEEE Trans. Neural Networks Learn. Syst., vol. 25, no. 7, pp. 1287–1297, 2014.

[43] Montavon, G., Braun, M. L., Krueger, T., & Muller, K. R. (2013). Analyzing local structure in kernel-based learning: Explanation, complexity, and reliability assessment. IEEE Signal Processing Magazine, 30(4), 62-74.

[44] K. Slavakis, P. Bouboulis, and S. Theodoridis, "Adaptive multi regression in reproducing kernel hilbert spaces: The multiaccess MIMO channel case," IEEE Trans. Neural Networks Learn. Syst., vol. 23, no. 2, pp. 260–276, 2012.

[45] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," IEEE Signal Process. Mag., vol. 28, no. 1, pp. 97–123, 2011.

[46] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," IEEE Trans. Signal Process., vol. 56, no. 7 I, pp. 2781–2796, 2008.

[47] K. Slavakis, S. Theodoridis, and I. Yamada, "Adaptive constrained learning in reproducing kernel hilbert spaces: The robust beamforming case," IEEE Trans. Signal Process., vol. 57, no. 12, pp. 4744–4764, 2009.

[48] Y. Fu, B. Li, X. Zhu, and C. Zhang, "Active learning without knowing individual instance labels: A pairwise label homogeneity query approach," IEEE Trans. Knowl. Data Eng., vol. 26, no. 4, pp. 808–822, 2014.

[49] B. Settles, "Active Learning Literature Survey," Mach. Learn., vol. 15, no. 2, pp. 201–221, 2010.

[50] M. M. Crawford, D. Tuia, and H. L. Yang, "Active learning: Any value for classification of remotely sensed data?," Proc. IEEE, vol. 101, no. 3, pp. 593–608, 2013.

[51] M. M. Haque, L. B. Holder, M. K. Skinner, and D. J. Cook, "Generalized query-based active learning to identify differentially methylated regions in DNA," IEEE/ACM Trans. Comput. Biol. Bioinforma., vol. 10, no. 3, pp. 632–644, 2013.

[52] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari, "A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification," IEEE J. Sel. Top. Signal Process., vol. 5, no. 3, pp. 606–617, 2011.

[53] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, "A survey of open source tools for machine learning with big data in the Hadoop ecosystem," J. Big Data, vol. 2, no. 1, p. 24, 2015.

[54] Kaluzka, J. (2016). Data locality in Hadoop (Master's thesis, Universitat Politècnica de Catalunya).

[55] Tom White, Hadoop The Definitive Guide, OREILLY, 2009.

[56] D. Pop and G. Iuhasz, "Overview of Machine Learning Tools and Libraries," Inst. e-Austria Timisoara.

[57] Suralkar, S., Mujumdar, A., Masiwal, G., & Kulkarni, M. (2013). Review of distributed file systems: Case studies. *Int. J. Eng. Res. Appl*, *3*, 1293-1298.

[58] P. R. A. Preethi and P. J. Elavarasi, "Big Data Analytics Using Hadoop Tools – Apache Hive Vs Apache Pig," vol. 24, no. 3, pp. 16–20, 2017.

[59] C. Douglas, J. Lowe, O. O. Malley, and B. Reed, "Apache Hadoop YARN : Yet Another Resource Negotiator."

[60] G. S. Sadasivam and D. Selvaraj, "A novel parallel hybrid PSO-GA using MapReduce to schedule jobs in Hadoop data grids," Proc. - 2010 2nd World Congr. Nat. Biol. Inspired Comput. NaBIC 2010, pp. 377–382, 2010.

[61] J. Talbot, R. M. Yoo, and C. Kozyrakis, "Phoenix++: Modular MapReduce for Shared-Memory Systems," MapReduce '11 Proc. Second Int. Work. MapReduce its Appl., pp. 9–16, 2011.

[62] Z. Khanam and S. Agarwal, "Map-Reduce Implementations: Survey and Performance Comparison," Int. J. Comput. Sci. Inf. Technol., vol. 7, no. 4, pp. 119–126, 2015.

[63] A. K. Koundinya, N. K. Srinath, K. A. K. Sharma, K. Kumar, and M. N. Madhu, "M Ap / R Educe Design and Implementation of a Priori a Lgorithm for Handling Voluminous Data - Sets," vol. 3, no. 6, pp. 29–39, 2012.

[64] S. Humbetov, "Data-intensive computing with map-reduce and Hadoop," 2012 6th Int. Conf. Appl. Inf. Commun. Technol. AICT 2012 - Proc., 2012.

[65] G. Engelberg, O. Koren, and N. Perel, "Big data performance evaluation analysis using apache pig," Int. J. Softw. Eng. its Appl., vol. 10, no. 11, pp. 429–440, 2016.

[66] Davenport, T. H., Barth, P., & Bean, R. (2012). How big data is different. MIT Sloan Management Review, 54(1), 43.

[67] F. Andersson, M. Carlsson, J. Tourneret, and S. Member, "A New Frequency Estimation Method for Equally and Unequally Spaced Data," vol. 62, no. 21, pp. 5761–5774, 2014.

[68] F. Lin, M. Fardad, and M. R. Jovanović, "Design of Optimal Sparse Feedback Gains via the Alternating Direction Method of Multipliers," vol. 58, no. 9, pp. 2426–2431, 2011.

[69] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for Similarity Search: A Survey," vol. 3, no. 1, pp. 1–122, 2014.

[70] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58.

[71] Choudhury, A. J., Kumar, P., Sain, M., Lim, H., & Jae-Lee, H. (2011, December). A strong user authentication framework for cloud computing. In Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific (pp. 110-115). IEEE.

[72] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed GraphLab," Proc. VLDB Endow., vol. 5, no. 8, pp. 716–727, 2012.

[73] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Commun. ACM, vol. 51, no. 1, p. 107, 2008.

[74] B. Y. J. Dean and S. Ghemawat, "MapReduce: a flexible data processing tool," Commun. ACM, vol. 53, no. 1, pp. 72–77, 2010.

[75] Chu, C. T., Kim, S. K., Lin, Y. A., Yu, Y., Bradski, G., Olukotun, K., & Ng, A. Y. (2007). Map-reduce for machine learning on multicore. In Advances in neural information processing systems (pp. 281-288).

[76] M. Lenzerini, "Data Integration: A Theoretical Perspective.," Proc.\ 21st ACM SIGACT SIGMOD SIGART Symp.\ Princ. Database Syst., pp. 233–246, 2002.

[77] A. Halevy and J. Ordille, "Data Integration : The Teenage Years," Artif. Intell., vol. 41, no. 1, pp. 9–16, 2006.

[78] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," Aistats, vol. 1, no. 3, pp. 448–455, 2009.

[79] 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. 2016, no. 1, p. 67, 2016.

[80] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," IEEE Trans. Signal Process., vol. 63, no. 10, pp. 2663–2677, 2015.

[81] Wu, Q., Ding, G., Xu, Y., Feng, S., Du, Z., Wang, J., & Long, K. (2014). Cognitive internet of things: a new paradigm beyond connection. IEEE Internet of Things Journal, 1(2), 129-143.

[82] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," IEEE Trans. Signal Process., vol. 52, no. 8, pp. 2165–2176, 2004.

[83] S. Shalev-Shwartz, "Online Learning and Online Convex Optimization," Found. Trends® Mach. Learn., vol. 4, no. 2, pp. 107–194, 2011.

[84] J. Wang, P. Zhao, S. C. H. Hoi, and R. Jin, "Online feature selection and its applications," IEEE Trans. Knowl. Data Eng., vol. 26, no. 3, pp. 698–710, 2014.

[85] M. Bilenko, S. Basu, and M. Sahami, "Adaptive product normalization: Using online learning for record linkage in comparison shopping," Proc. - IEEE Int. Conf. Data Mining, ICDM, pp. 58–65, 2005.

[86] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, no. 1–3, pp. 489–501, 2006.

[87] S. Ding, X. Xu, and R. Nie, "Extreme learning machine and its applications," Neural Comput. Appl., vol. 25, no. 3–4, pp. 549–556, 2014.

[88] N. Tatbul, "Streaming data integration: Challenges and opportunities.," ICDE Work., pp. 155–158, 2010.

[89] Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2014). Data mining with big data. IEEE transactions on knowledge and data engineering, 26(1), 97-107.

[90] S. Tsang, B. Kao, K. Y. Yip, W. S. Ho, and S. D. Lee, "Decision trees for uncertain data," Knowl. Data Eng. IEEE Trans., vol. 23, no. 1, pp. 64–78, 2011.

[91] H. Hu, Y. Wen, and X. Li, "A Framework for Big Data Analytics as a Scalable Systems," IEEE Access, vol. 2, pp. 652–687, 2014.

[92] Gantz, J., & Reinsel, D. (2011). Extracting value from chaos. IDC iview, 1142(2011), 1-12.

[93] C. W. Tsai, C. F. Lai, M. C. Chiang, and L. T. Yang, "Data mining for internet of things: A survey," IEEE Commun. Surv. Tutorials, vol. 16, no. 1, pp. 77–97, 2014.

[94] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," AI Mag., vol. 17, no. 3, p. 37, 1996.