

A Survey of Machine Learning Methods for Big Data

Suresh Babu ¹, Dr.M. Manoj Kumar², Bayyapuneni Anusha³,

¹Assistant Professor, Department of Computer & Science Engineering, RISE Krishna Sai Gandhi Group of Institutions, Ongole, Valluru 523 272, India

²Professor, Department of Computer & Science Engineering, RISE Krishna Sai Gandhi Group of Institutions, Ongole, Valluru 523 272, India

³ Assistant Professor, Department of Computer & Science Engineering, RISE Krishna Sai Gandhi Group of Institutions, Ongole, Valluru 523 272, India

ABSTRACT: In this research paper, we explore a diverse array of machine learning algorithms and offer a comprehensive literature survey of the most recent developments in the field of machine learning for processing big data. Our examination begins with a detailed overview of the categories and algorithms within machine learning, showcasing their versatile applications in areas such as data mining, image processing, and predictive analytics. One of the prominent benefits of employing machine learning lies in its capacity to enable automated decision-making once the algorithms acquire the necessary data comprehension. Lastly, we address a range of ongoing challenges and emerging research directions in this dynamic field.

Keywords— Machine learning, machine algorithm, Learning methodology

I. INTRODUCTION

Machine learning serves the purpose of instructing machines on how to effectively manage and interpret data. At times, when data patterns prove elusive or data information remains latent, machine learning steps in. Given the profusion of available datasets, the demand for machine learning has been steadily increasing. Various sectors, spanning from medicine to the military, actively employ machine learning to extract pertinent insights. The core objective of machine learning is knowledge acquisition from data, and numerous studies have been undertaken by mathematicians and programmers to explore various avenues for enabling machines to autonomously acquire this knowledge. Some of them are demonstrated in Fig. 1.

II. TYPES OF LEARNING

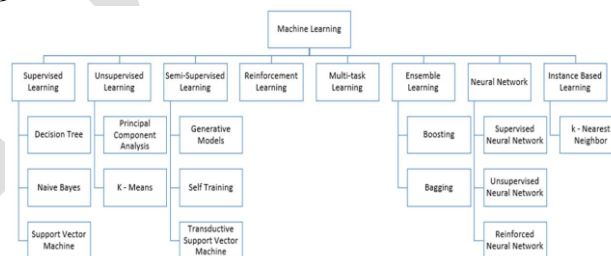


Fig:-1 Types of Learning

2.1 Supervised Learning

Supervised machine learning algorithms require external guidance for their operation. The input dataset is typically segregated into training and testing datasets. The training dataset contains the output variable that the algorithm aims to predict or classify. These algorithms glean patterns from the training data and subsequently employ these patterns to predict or classify data within the test dataset. The workflow of supervised machine learning algorithms is given in Fig. 2.

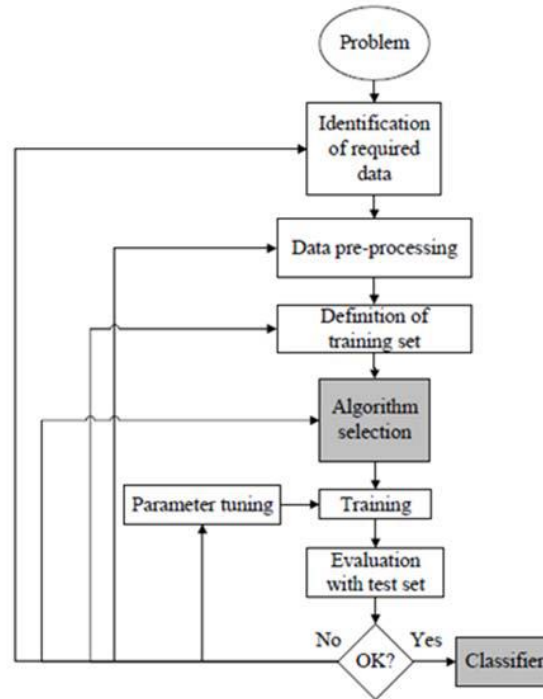


Fig:-2 Workflow of supervised machine learning algorithm

Three most famous supervised machine learning algorithms like,

1. Decision Tree:
2. Naïve Bayes
3. Support Vector Machine.

1) Decision Tree: Decision trees are a specific kind of tree structure that organizes attributes by sorting them according to their values. These trees are primarily employed for classification purposes. In a decision tree, you find nodes and branches. Each node symbolizes a group of attributes to be classified, and each branch signifies a potential value that the node may assume.

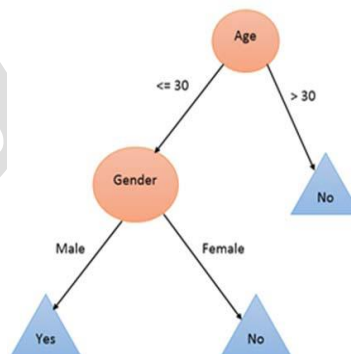


Fig. 3. Decision Tree

An example of decision tree is given in Fig. 3. The pseudo code for Decision tree is described in Fig. 4; where S, A and y are training set, input attribute and target attribute respectively.

```

procedure DTInducer( $S, A, y$ )
1:  $T = \text{TreeGrowing}(S, A, y)$ 
2: Return TreePruning( $S, T$ )
procedure TreeGrowing( $S, A, y$ )
1: Create a tree  $T$ 
2: if One of the Stopping Criteria is fulfilled then
3:   Mark the root node in  $T$  as a leaf with the most common
   value of  $y$  in  $S$  as the class.
4: else
5:   Find a discrete function  $f(A)$  of the input attributes val-
   ues such that splitting  $S$  according to  $f(A)$ 's outcomes
   ( $v_1, \dots, v_n$ ) gains the best splitting metric.
6:   if best splitting metric  $\geq$  threshold then
7:     Label the root node in  $T$  as  $f(A)$ 
8:     for each outcome  $v_i$  of  $f(A)$  do
9:        $\text{Subtree}_i = \text{TreeGrowing}(\sigma_{f(A)=v_i} S, A, y)$ .
10:      Connect the root node of  $T$  to  $\text{Subtree}_i$  with an
       edge that is labelled as  $v_i$ 
11:    end for
12:   else
13:     Mark the root node in  $T$  as a leaf with the most
     common value of  $y$  in  $S$  as the class.
14:   end if
15: end if
16: Return  $T$ 
procedure TreePruning( $S, T, y$ )
1: repeat
2:   Select a node  $t$  in  $T$  such that pruning it maximally
   improve some evaluation criteria
3:   if  $t \neq \emptyset$  then
4:      $T = \text{pruned}(T, t)$ 
5:   end if
6: until  $t = \emptyset$ 
7: Return  $T$ 

```

Fig. 4. Pseudo code for Decision Tree

2.2 Unsupervised Learning

Unsupervised learning algorithms extract key features from the data during their learning process. When presented with new data, they leverage the previously acquired features to categorize the data. Unsupervised learning primarily finds application in tasks like clustering and feature dimensionality reduction. An example of workflow of unsupervised learning is given in Fig. 5.

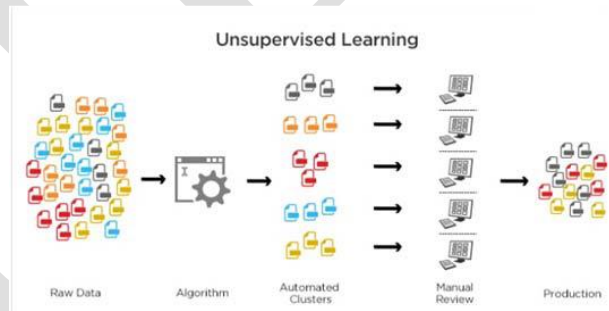


Fig. 5. Example of Unsupervised Learning

The two main algorithms for clustering and dimensionality reduction techniques are discussed below.

1. K-Means Clustering
 2. Principal Component Analysis
- 1) **K-Means Clustering:** Clustering, a form of unsupervised learning, autonomously organizes data into groups or clusters based on their inherent similarities. Items sharing common characteristics are grouped together within a cluster. The k-means algorithm is so named because it produces k distinct clusters, where the center of each cluster is determined by the mean of the values within that cluster. A clustered data is represented in Fig. 6. The algorithm for k-means is given in Fig. 7.

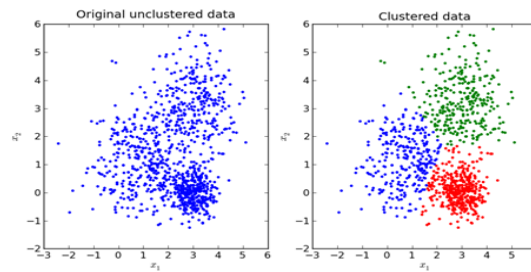


Fig. 6. K-Means Clustering

```

function Direct-k-means()
Initialize  $k$  prototypes  $(w_1, \dots, w_k)$  such that  $w_j =$ 
 $i_l, j \in \{1, \dots, k\}, l \in \{1, \dots, n\}$ 
Each cluster  $C_j$  is associated with prototype  $w_j$ 
Repeat
  for each input vector  $i_l$ , where  $l \in \{1, \dots, n\}$ ,
  do
    Assign  $i_l$  to the cluster  $C_{j^*}$  with near-
    est prototype  $w_{j^*}$ 
    (i.e.,  $|i_l - w_{j^*}| \leq |i_l - w_j|, j \in$ 
     $\{1, \dots, k\}$ )
  for each cluster  $C_j$ , where  $j \in \{1, \dots, k\}$ , do
    Update the prototype  $w_j$  to be the
    centroid of all samples currently
    in  $C_j$ , so that  $w_j = \sum_{i_l \in C_j} i_l / |$ 
     $C_j|$ 
  Compute the error function:

$$E = \sum_{j=1}^k \sum_{i_l \in C_j} |i_l - w_j|^2$$

Until  $E$  does not change significantly or cluster mem-
bership no longer changes

```

Fig. 7. Pseudo code for k-means clustering

2.3 Semi - Supervised Learning

Semi-supervised learning algorithms integrate the strengths of both supervised and unsupervised learning techniques. This approach proves valuable in machine learning and data mining contexts where unlabeled data is readily available, and acquiring labeled data is a challenging task. Semi-supervised learning encompasses various categories tailored to address this unique scenario. Some of which are discussed below:

1) **Generative Models:** Generative models, among the earliest semi-supervised learning methods, assume a structure represented as $p(x,y) = p(y)p(x|y)$, where $p(x|y)$ signifies a blended distribution, such as Gaussian mixture models. This framework allows the identification of mixed components within unlabeled data. Remarkably, merely one labeled example per component proves sufficient to validate the mixture distribution.

2) **Self-Training:** In the self-training approach, a classifier is initially trained using a subset of labeled data. Subsequently, the classifier is utilized to predict labels for unlabeled data points. These predicted labels, along with the corresponding unlabeled points, are incorporated into the training set. This iterative process is then repeated. The term "self-training" derives from the fact that the classifier refines its learning by incorporating its own predictions, leading to an iterative self-improvement process.

2.4 Reinforcement Learning

Reinforcement learning is a learning paradigm where decisions are made by selecting actions that maximize positive outcomes. In this approach, the learner starts with no prior knowledge of which actions to take until presented with a specific situation. Importantly, the learner's chosen actions can impact both the current situation and future actions. Reinforcement learning primarily relies on two fundamental principles: trial-and-error exploration and the delayed consequences of actions.. The general model for reinforcement learning is depicted in Fig.8.

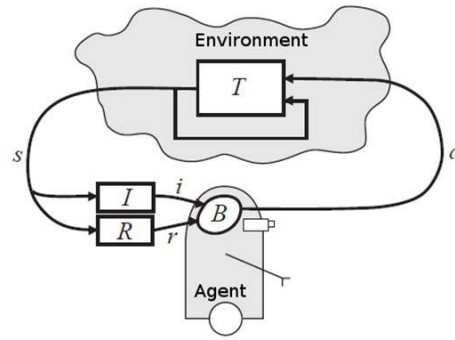


Fig. 8. The Reinforcement Learning Model

In the figure, the agent receives an input i , current state s , state transition r and input function I from the environment. Based on these inputs, the agent generates a behavior B and takes an action which generates an outcome.

2.5 Multitask Learning

Multitask learning aims to enhance the performance of multiple learners. When these algorithms are employed for a task, they memorize the steps used to solve the problem or reach specific conclusions. These learned procedures are then utilized to tackle similar problems or tasks. This process, where one algorithm assists another, is known as an inductive transfer mechanism. By sharing their experiences, learners can collectively and concurrently acquire knowledge, allowing for faster learning compared to individual efforts.

2.6 Ensemble Learning

Ensemble learning is the process of amalgamating multiple individual learners into a unified learner. These individual learners can be diverse, including Naïve Bayes, decision trees, neural networks, and more. It has been consistently observed that a collective of learners tends to outperform individual learners when tackling specific tasks. Two well-known techniques for ensemble learning are highlighted below.

1) Boosting: Boosting, a method in ensemble learning, is employed to reduce both bias and variance. This technique involves constructing a group of weak learners and transforming them into a powerful single learner. A weak learner refers to a classifier that has limited correlation with accurate classification, while a strong learner is a classifier strongly aligned with true classification. Boosting helps elevate the performance of weak learners, amalgamating their individual outputs to create a robust and accurate overall classifier. The pseudo code for AdaBoost (which is most popular example of boosting) is given in Fig. 9.

Input: Data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$;
Base learning algorithm \mathcal{L} ;
Number of learning rounds T .

Process:

$$D_1(i) = 1/m.$$

for $t = 1, \dots, T$:

$$h_t = \mathcal{L}(\mathcal{D}, D_t);$$

$$\epsilon_t = \Pr_{i \sim D_t}[h_t(\mathbf{x}_i) \neq y_i];$$

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t};$$

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(\mathbf{x}_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(\mathbf{x}_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z_t} \end{aligned}$$

end.

Output: $H(\mathbf{x}) = \text{sign}(f(\mathbf{x})) = \text{sign} \sum_{t=1}^T \alpha_t h_t(\mathbf{x})$

Fig. 9. Pseudo code for AdaBoost

G. Neural Network Learning

The neural network (or artificial neural network or ANN) is derived from the biological concept of neurons. A neuron is a cell like structure in a brain. To understand neural network, one must understand how a neuron works. A neuron has mainly four parts (see Fig. 10). They are dendrites, nucleus, soma and axon.

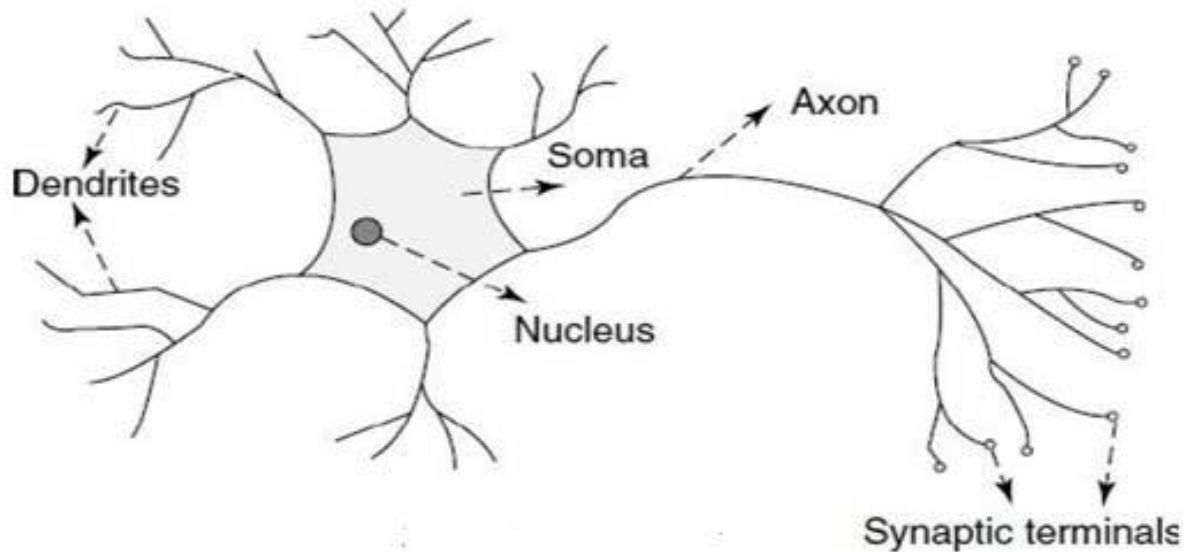


Fig. 10. A Neuron

2.7 Genetic algorithm

Genetic algorithm is adaptive heuristics search algorithm based on the evolutionary ideas of natural selection and genetics. It also mimics the process of natural selection. This heuristic sometimes identified as met heuristic, is used to generate the useful solutions to optimization and research problems. In order to implement feature subset selection method is Hadoop platform using genetic algorithm. GAs begins with a set of k randomly generated states called population. Each state is called individual and is represented as a string over a finite alphabet. This string is called chromosome and each symbol gene. Genetic algorithm are the way of solving problem by mirroring processes nature uses i.e. Selection, Crossover, Mutation and Accepting to develop solution to the problem. GAs uses the random search technique to solve the optimization problems. Every iteration of the algorithm generates a new population and consists of the following steps: 1) Selection: Each state is evaluated by the fitness function (b). Each value influences the random choice among the successors for the next step. Once chosen the k successors are grouped into couples (c); 2) Crossover: The algorithm chooses for each individual a division point, which is called crossover point. At this point the sexual reproduction (d) in which two children are created begins: the first takes the first part of the first parent and the second of the second parent; the other takes the second part of the first parent and first part of the second parent, the logic is used to identify the individual in the set. The crossover parents form a new offspring, or if no crossover is performed offspring copy the parents. To make individuals meet their constraint, newly generated individuals need to be modified. Using the crossover probability P_c , empty pool set PI for each individual a population is generate a real number $q \in [0,1]$, For each individuals in the set generate a new individual thus the crossover operation accomplished. 3) Mutation: When the offspring's are generated, each gene is subjected to a random mutation with a small independent probability (e). It selects the individual from the offspring of crossover according to the mutation probability P_m . Genetic algorithm is good at taking large, potentially huge search spaces and navigating them, looking for optimal combination of things, the solution one might not find

anywhere. Genetic algorithm begins with the set of solution called population, solution from one population are taken to form a new population where new population is better than exist one. Solution is selected according to the fitness from the existing, is repeated until the some condition is satisfied.

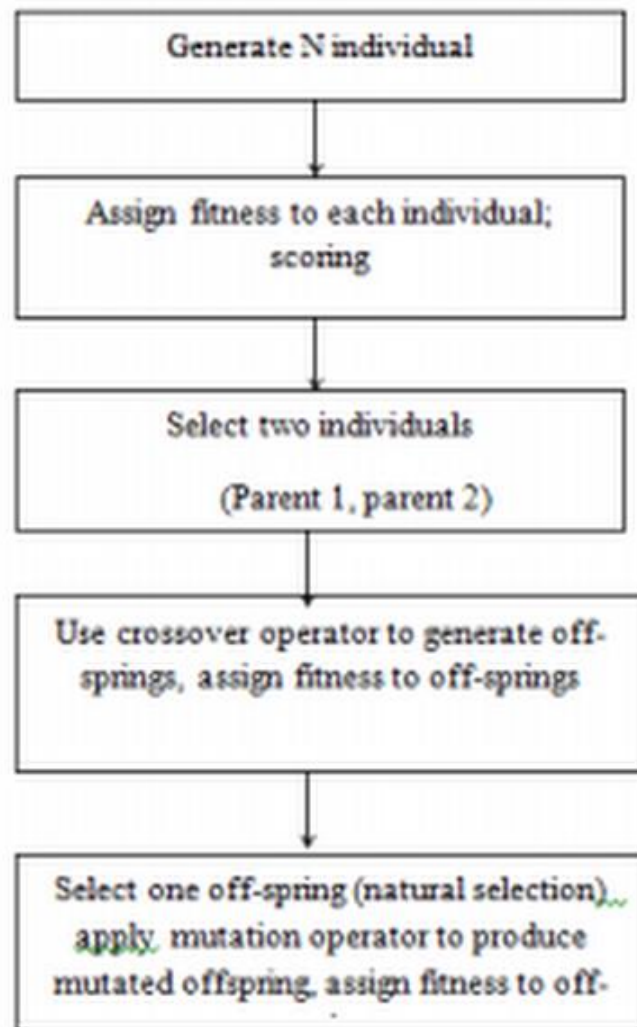


Fig. 11. A Genetic algorithm

III. BRIEF REVIEW OF MACHINE LEARNING TECHNIQUES

In this subsection, we introduce a few recent learning methods that may be either promising or much needed for solving the big data problems. The outstanding characteristic of these methods is to focus on the idea of learning, rather than just a single algorithm.

3.1 Representation Learning: Datasets with high dimensional features have become increasingly common nowadays, which challenge the current learning algorithms to extract and organize the discriminative information from the data. Fortunately, representation learning, a promising solution to learn the meaningful and useful representations of the data that make it easier to extract useful information when building classifiers or other predictors, has been presented and achieved impressive performance on many dimensionality reduction tasks. Representation learning aims to achieve that a reasonably sized learned representation can capture a huge number of possible input configurations, which can greatly facilitate improvements in both computational efficiency and statistical efficiency.

3.2 Deep learning: Nowadays, there is no doubt that deep learning is one of the hottest research trends in machine learning field. In contrast to most traditional learning techniques, which are considered using shallow-structured learning architectures, deep learning mainly uses supervised and/or unsupervised strategies in deep architectures to automatically learn hierarchical representations. Deep architectures can often capture more complicated, hierarchically launched statistical patterns of inputs for achieving to be adaptive to new areas than traditional learning methods and often outperform state of the art achieved by hand-made features.

3.3 Distributed and parallel learning: There is often exciting information hidden in the unprecedented volumes of data. Learning from these massive data is expected to bring significant science and engineering advances which can facilitate the development of more intelligent systems. However, a bottleneck preventing such a big blessing is the inability of learning algorithms to use all the data to learn within a reasonable time. In this context, distributed learning seems to be a promising research since allocating the learning process among several workstations is a natural way of scaling up learning algorithms. Different from the classical learning framework, in which one requires the collection of that data in a database for central processing, in the framework of distributed learning, the learning is carried out in a distributed manner. In the past years, several popular distributed machine learning algorithms have been proposed, including decision rules, stacked generalization, meta-learning, and distributed boosting. With the advantage of distributed computing for managing big volumes of data, distributed learning avoids the necessity of gathering data into a single workstation for central processing, saving time and energy.

IV. BRIEF REVIEW OF CRITICAL ISSUE

4.1 Critical issue one: learning for large scale of data: Critical issue It is obvious that data volume is the primary attribute of big data, which presents a great challenge for machine learning. Taking only the digital data as an instance, every day, Google alone needs to process about 24 petabytes of data. Moreover, if we further take into consideration other data sources, the data scale will become much bigger. Under current development trends, data stored and analyzed by big organizations will undoubtedly reach the petabyte to exabyte magnitude soon.

4.2 Critical issue two: learning for different types of data: Critical issue the enormous variety of data is the second dimension that makes big data both interesting and challenging. This is resulted from the phenomenon that data generally come from various sources and are of different types. Structured, semi-structured, and even entirely unstructured data sources stimulate the generation of heterogeneous, high-dimensional, and nonlinear data with different representation forms. Learning with such a dataset, the great challenge is perceivable and the degree of complexity is not even imaginable before we deeply get there.

4.3 Critical issue three: learning for high speed of streaming data: Critical issue for big data, speed or velocity really matters, which is another emerging challenge for learning. In many real-world applications, we have to finish a task within a certain period of time; otherwise, the processing results become less valuable or even worthless, such as earthquake prediction, stock market prediction and agent-based autonomous exchange (buying/selling) systems, and so on. In these time-sensitive cases, the potential value of data depends on data freshness that needs to be processed in a real-time manner.

4.4 Critical issue four: learning for uncertain and incomplete data: Critical issue In the past, machine learning algorithms were typically fed with relatively accurate data from well known and quite limited sources, so the learning results tend to be unerring, too; thus, veracity has never been a serious issue for concern. However, with the sheer size of data available today, the precision and

trust of the source data quickly become an issue, due to the data sources are often of many different origins and data quality is not all verifiable. Therefore, we include veracity as the fourth critical issue for learning with big data to emphasize the importance of addressing and managing the uncertainty and incompleteness on data quality.

V. RESEARCH TRENDS AND OPEN ISSUES

While significant progress has been made in the last decade toward achieving the ultimate goal of making sense of big data by machine learning techniques, the consensus is that we are still not quite there. The efficient preprocessing mechanisms to make the learning system capable of dealing with big data and effective learning technologies to find out the rules to describe the data are still of urgent need. Therefore, some of the open issues and possible research trends are given in Fig. 12.

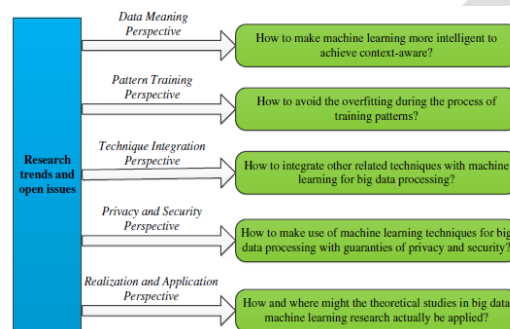


Fig:-12 Research trends and open issues

VI. CONCLUSION

This paper surveys various machine learning algorithms. Today each and every person is using machine learning knowingly or unknowingly. From getting a recommended product in online shopping to updating photos in social networking sites. This paper gives an introduction to most of the popular machine learning algorithms. Then, a discussion about the challenges of learning with big data at last, open issues and research trends were presented.

REFERENCES

- [1]. w. Richert, L. P. Coelho, “*Building Machine Learning Systems with Python*”, Packt Publishing Ltd., ISBN 978-1-78216-140-0.
- [2]. M. Welling, “*A First Encounter with Machine Learning*.”
- [3]. S.B. Kotsiantis, “*Supervised Machine Learning: A Review of Classification Techniques*”, Informatica 31 (2007) 249-268.
- [4]. L. Rokach, O. Maimon, “*Top – Down Induction of Decision Trees Classifiers – A Survey*”, IEEE Transactions on Systems.
- [5]. D. Meyer, “*Support Vector Machines – The Interface to libsvm in package e1071*”, August 2015.
- [6]. S. S. Shwartz, Y. Singer, N. Srebro, “*Pegasos: Primal Estimated sub - Gradient Solver for SVM*”, Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, 2007.
- [7]. M. Andrecut, “*Parallel GPU Implementation of Iterative PCA Algorithms*”, Institute of Biocomplexity and Informatics, University of Calgary, Canada, 2008
- [8]. P. Harrington, “*Machine Learning in action*”, Manning Publications Co., Shelter Island, New York, 2012