ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.3

Machine Learning Algorithms:Trends, Perspectives and Prospects

Priyanka P. Raut[1], Namrata R. Borkar[2]
ME Student[1], Assistant Professsor[2]
Department of CSE
Dr. Sau. Kamlatai Gawai Institute of Engineering and Technology, Darapur, MS, India

Abstract:
Machine learning addresses the question of how to build computers that improve automatically through experience. It is one of today's most rapidly growing technical fields, lying at the intersection of computer science and statistics, and at the core of artificial intelligence and data science. Recent progress in machine learning has been driven both by the development of new learning algorithms and theory and by the ongoing explosion in the availability of online data and low-cost computation. The adoption of data-intensive machine-learning methods can be found throughout science, technology and commerce, leading to more evidence-based decision-making across many walks of life, including health care, manufacturing, education, financial modeling, policing, and marketing.

Keywords: Supervised Machine Learning, SVM, DT, Classifier.

I. INTRODUCTION

Machine learning is used to teach machines how to handle the data more efficiently. Sometimes after viewing the data, we cannot interpret the pattern or extract information from the data. In that case, we apply machine learning [1]. With the abundance of datasets available, the demand for machine learning is in rise. Many industries from medicine to military apply machine learning to extract relevant information. Machine learning has progressed dramatically over the past two decades, from laboratory curiosity to a practical technology in widespread commercial use. Within artificial intelligence (AI), machine learning has emerged as the method of choice for developing practical software for computer vision, speech recognition, natural language processing, robot control, and other applications. Many developers of AI systems no recognize that, for many applications, it can be far easier to train a system by showing it examples of desired input-output behavior than to program it manually by anticipating the desired response for all possible inputs. The effect of machine learning has also been felt broadly across computer science and across a range of industries concerned with data-intensive issues, such as consumer services, the diagnosis of faults in complex systems, and the control of logistics chains. There has been a similarly broad range of effects across empirical sciences, from biology to cosmology to social science, as machine-learning methods have been developed to analyze high throughput experimental data in novel ways. The purpose of machine learning is to learn from the data. Many studies have been done on how to make machines learn by themselves [2] [3]. Many mathematicians and programmers apply several approaches to find the solution of this problem.

II. TYPES OF LEARNING

A. Supervised Learning

The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification [4]. The workflow of supervised machine learning algorithms is given in Fig.1. Three most famous supervised machine learning algorithms have been discussed here.

1) Decision Tree: Decision trees are those type of trees which groups attributes by sorting them based on their values. Decision tree is used mainly for classification purpose. Each tree consists of nodes and branches. Each nodes represents attributes in a group that is to be classified and each branch represents a value that the node can take [4]. An example of decision tree is given in Fig. 2.

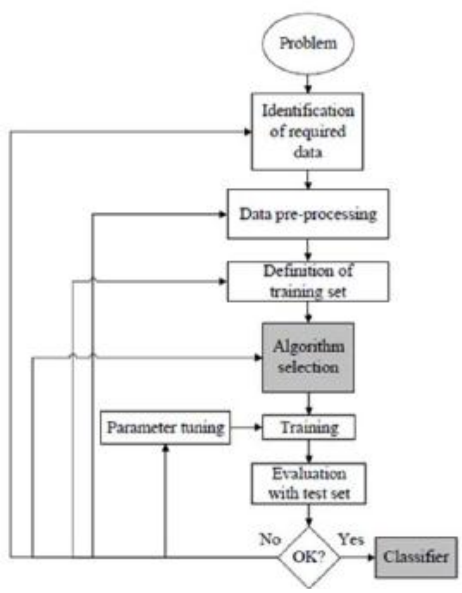

Figure.1. Workflow of supervised machine learning algorithm [4]

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.

2) Naïve Bayes: Naïve Bayes mainly targets the text classification industry. It is mainly used for clustering and classification purpose [6]. The underlying architecture of Naïve Bayes depends on the conditional probability. It creates trees based on their probability of happening. These trees are also known as Bayesian Network. An example of the network is given in Fig. 4. The pseudo code is given in Fig. 5.

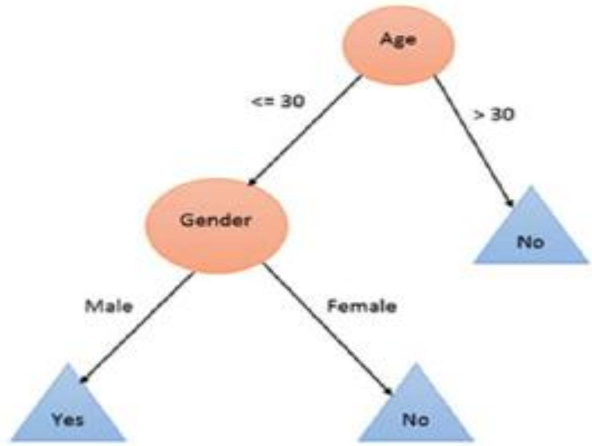

Figure.2. Decision tree [5]

```
procedure DTInducer(S, A, y)
  T = TreeGrowing(S, A, y)
  Return TreePruning(S,T)
procedure TreeGrowing(S, A, y)
  Create a tree T
  if One of the Stopping Criteria is fulfilled then
     Mark the root node in T as a leaf with the most common
     value of y in S as the class.
  else
     Find a discrete function f(A) of the input attributes val-
     ues such that splitting S according to f(A)'s outcomes
     (v1,...,vn) gains the best splitting metric.
     if best splitting metric ≥ treshold then
        Label the root node in T as f(A)
        for each outcome v_i of f(A) do
           Subtree_i = TreeGrowing(σ_{f(A)=v_i} S, A, y).
           Connect the root node of T to Subtree_i with an
           edge that is labelled as v_i
        end for
     else
        Mark the root node in T as a leaf with the most
        common value of y in S as the class.
     end if
  end if
  Return T
procedure TreePruning(S, T, y)
  repeat
     Select a node t in T such that pruning it maximally
     improve some evaluation criteria
     if t ≠ Ø then
        T = pruned(T, t)
     end if
  until t=Ø
  Return T
```

Figure. 3. Pseudo code for decision tree [5]

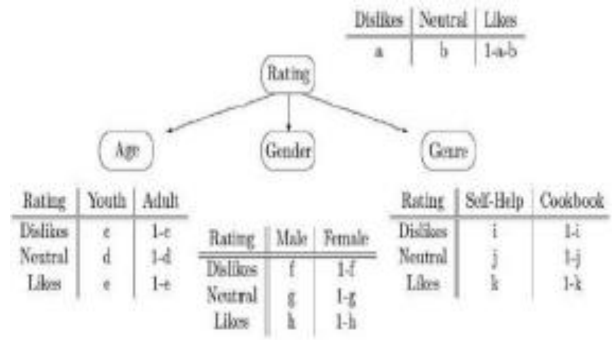

Figure. 4. An example of bayesian network [7]

```
INPUT: training set T, hold-out set H, initial number of compo-
nents k_0, and convergence thresholds δ_EM and δ_Add.

  Initialize M with one component.
  k ← k_0
  repeat
    Add k new mixture components to M, initialized using k
    random examples from T.
    Remove the k initialization examples from T.
    repeat
      E-step: Fractionally assign examples in T to mixture com-
      ponents, using M.
      M-step: Compute maximum likelihood parameters for M,
      using the filled-in data.
      If log P(H|M) is best so far, save M in M_best.
      Every 5 cycles, prune low-weight components of M.
    until log P(H|M) fails to improve by ratio δ_EM.
    M ← M_best
    Prune low weight components of M.
    k ← 2k
  until log P(H|M) fails to improve by ratio δ_Add.
  Execute E-step and M-step twice more on M_best, using exam-
  ples from both H and T.
  Return M_best.
```

Figure .5. Pseudo code for naïve bayes [6]

3) Support Vector Machine: Another most widely used state-of-the-art machine learning technique is Support Vector Machine (SVM). It is mainly used for classification. SVM works on the principle of margin calculation. It basically, draw margins between the classes. The margins are drawn in such a fashion that the distance between the margin and the classes is maximum and hence, minimizing the classification error. An example of working and pseudo code of SVM is given in Fig. 6 and Fig. 7, respectively.

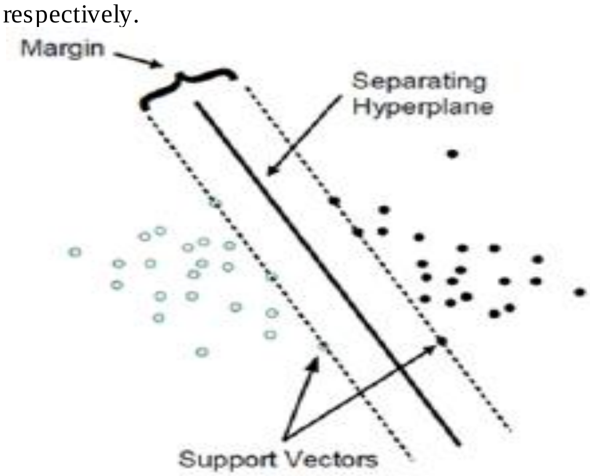

Figure. 6. Working of support vector machine [8]

INPUT: S, λ, T, k

INITIALIZE: Choose $\mathbf{w}_1$ s.t. $\|\mathbf{w}_1\| \le 1/\sqrt{\lambda}$

FOR $t = 1, 2, \ldots, T$

Choose $A_t \subseteq S$, where $|A_t| = k$

Set $A_t^+ = \{(\mathbf{x}, y) \in A_t : y\langle \mathbf{w}_t, \mathbf{x}\rangle < 1\}$

Set $\eta_t = \frac{1}{\lambda t}$

Set $\mathbf{w}_{t+\frac{1}{2}} = (1 - \eta_t \lambda)\mathbf{w}_t + \frac{\eta_t}{k} \sum_{(\mathbf{x},y)\in A_t^+} y\,\mathbf{x}$

Set $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+\frac{1}{2}}\|}\right\} \mathbf{w}_{t+\frac{1}{2}}$

OUTPUT: $\mathbf{w}_{T+1}$

Figure. 7. Pseudo code for support vector machine [9]

B. Unsupervised Learning

The unsupervised learning algorithms learn few features from the data. When new data is introduced, it uses the previously learned features to recognize the class of the data. It is mainly used for clustering and feature reduction. An example of workflow of unsupervised learning is given in Fig. 8.

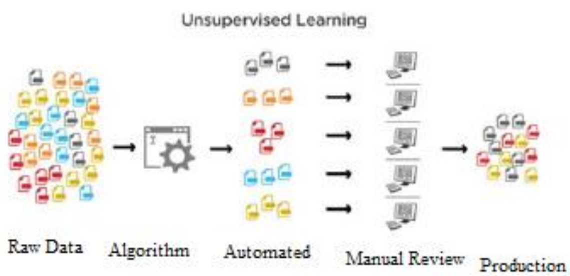

Figure. 8. Example of unsupervised learning [10]

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

1) *K-Means Clustering:*

Clustering or grouping is a type of unsupervised learning technique that when initiates, creates groups automatically. The items which possesses similar characteristics are put in the same cluster. This algorithm is called k-means because it creates k distinct clusters. The mean of the values in a particular cluster is the center of that cluster [9]. A clustered data is represented in Fig. 9. The algorithm for k-means is given in Fig. 10.

Figure .9. K-means clustering [7]

2) *Principal Component Analysis:*

In Principal Component Analysis or PCA, the dimension of the data is reduced to make the computations faster and easier. To understand how PCA works, let's take an example of 2D data. When the data is being plot in a graph, it will take up two axes. PCA is applied on the data, the data then will be 1D. This is explained in Fig. 11. The pseudo code for PCA is discussed in Fig. 12.

function Direct-k-means()

Initialize k prototypes $(w_1, \ldots, w_k)$ such that $w_j = i_l$, $j \in \{1, \ldots, k\}$, $l \in \{1, \ldots, n\}$

Each cluster C_j is associated with prototype w_j

Repeat

for each input vector i_l, where $l \in \{1, \ldots, n\}$, *do*

Assign i_l to the cluster C_{j^*} with nearest prototype w_{j^*} (i.e., $|i_l - w_{j^*}| \le |i_l - w_j|$, $j \in \{1, \ldots, k\}$)

for each cluster C_j, where $j \in \{1, \ldots, k\}$, *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 membership no longer changes

Figure.10. Pseudo code for k-means clustering [12]

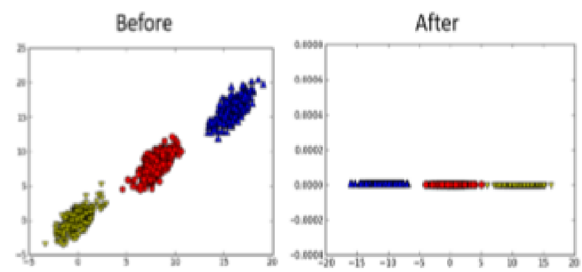

Figure.11. Visualization of data before and after applying pca [11]

$\mathbf{R} \leftarrow \mathbf{X}$

for$(k = 0, \ldots, K-1)$ do

{

$\lambda = 0$

$\mathbf{T}^{(k)} \leftarrow \mathbf{R}^{(k)}$

for$(j = 0, \ldots, J)$ do

{

$\mathbf{P}^{(k)} \leftarrow \mathbf{R}^T\mathbf{T}^{(k)}$

$\mathbf{P}^{(k)} \leftarrow \mathbf{P}^{(k)} \left\|\mathbf{P}^{(k)}\right\|^{-1}$

$\mathbf{T}^{(k)} \leftarrow \mathbf{R}\mathbf{P}^{(k)}$

$\lambda' \leftarrow \left\|\mathbf{T}^{(k)}\right\|$

if$(|\lambda' - \lambda| \le \varepsilon)$ then break

$\lambda \leftarrow \lambda'$

}

$\mathbf{R} \leftarrow \mathbf{R} - \mathbf{T}^{(k)}(\mathbf{P}^{(k)})^T$

}

return $\mathbf{T}, \mathbf{P}, \mathbf{R}$

Figure. 12. Pseudo code for pca [12]

C. *Semi - Supervised Learning*

Semi – supervised learning algorithms is a technique which combines the power of both supervised and unsupervised learning. It can be fruit-full in those areas of machine learning and data mining where the unlabeled data is already present and getting the labeled data is a tedious process [14]. There are many categories of semi-supervised learning [15]. Some of which are discussed below:

1) *Generative Models:* Generative models are one of the oldest semi-supervised learning method assumes a structure like p(x,y) = p(y)p(x|y) where p(x|y) is a mixed distribution e.g. Gaussian mixture models. Within the unlabeled data, the mixed components can be identifiable. One labeled example per component is enough to confirm the mixture distribution.

2) *Self-Training:* In self-training, a classifier is trained with a portion of labeled data. The classifier is then fed with unlabeled data. The unlabeled points and the predicted labels are added together in the training set. This procedure is then repeated further. Since the classifier is learning itself, hence the name self-training.

3) *Transductive SVM:* Transductive support vector machine or TSVM is an extension of SVM. In TSVM, the labeled and unlabeled data both are considered. It is used to label the unlabeled data in such a way that the margin is maximum between the labeled and unlabeled data. Finding an exact solution by TSVM is a NP-hard problem.

D. Reinforcement Learning

Reinforcement learning is a type of learning which makes decisions based on which actions to take such that the outcome is more positive. The learner has no knowledge which actions to take until it's been given a situation. The action which is taken by the learner may affect situations and their actions in the future. Reinforcement learning solely depends on two criteria: trial and error search and delayed outcome [16]. The general model [17] for reinforcement learning is depicted in Fig. 13.

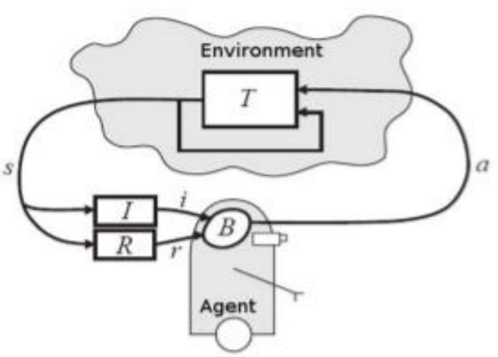

Figure.13. The reinforcement learning model [17]

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 a which generates an outcome.

E. Multitask Learning

Multitask learning has a simple goal of helping other learners to perform better. When multitask learning algorithms are applied on a task, it remembers the procedure how it solved the problem or how it reaches to the particular conclusion. The algorithm then uses these steps to find the solution of other similar problem or task. This helping of one algorithm to another can also be termed as inductive transfer mechanism. If the learners share their experience with each other, the learners can learn concurrently rather than individually and can be much faster [18].

F. Ensemble Learning

When various individual learners are combined to form only one learner then that particular type of learning is called ensemble learning. The individual learner may be Naïve Bayes, decision tree, neural network, etc. Ensemble learning is a hot topic since 1990s. It has been observed that, a collection of learners is almost always better at doing a particular job rather than individual learners [19]. Two popular Ensemble learning techniques are given below [20]:

1) Boosting:

Boosting is a technique in ensemble learning which is used to decrease bias and variance. Boosting creates a collection of weak learners and convert them to one strong learner. A weak learner is a classifier which is barely correlated with true classification. On the other hand, a strong learner is a type of classifier which is strongly correlated with true classification [20]. The pseudo code for AdaBoost (which is most popular example of boosting) is give in Fig. 14.

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

Process:

$D_1(i) = 1/m$.

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

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

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

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

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

$$= \frac{D_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$$

end.

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

Figure .14. Pseudo code for adaboost [20]

2) Bagging: Bagging or bootstrap aggregating is applied where the accuracy and stability of a machine learning algorithm needs to be increased. It is applicable in classification and regression. Bagging also decreases variance and helps in handling overfitting [7]. The pseudo code for bagging in given in Fig. 15.

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

Process:

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

$\mathcal{D}_t = Bootstrap(\mathcal{D})$;

$h_t = \mathcal{L}(\mathcal{D}_t)$

end.

Output: $H(\boldsymbol{x}) = \text{argmax}_{y \in \mathcal{Y}} \sum_{t=1}^{T} 1(y = h_t(\boldsymbol{x}))$

Figure.15. Pseudo code for bagging [20]

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. 16). They are dendrites, nucleus, soma and axon.

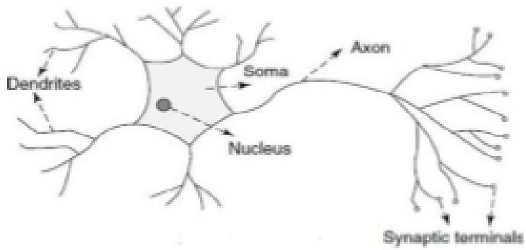

Figure. 16. A neuron [21]

The dendrites receive electrical signals. Soma processes the electrical signal. The output of the process is carried by the axon to the dendrite terminals where the output is sent to next neuron. The nucleus is the heart of the neuron. The inter-connection of neuron is called neural network where electrical impulses travel around the brain.

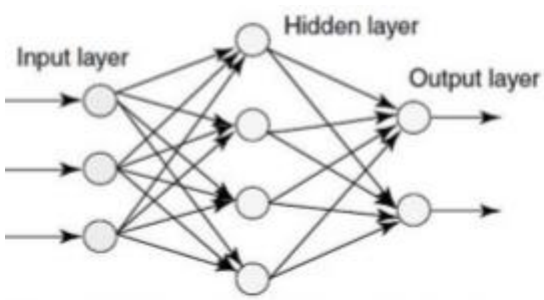

Figure. 17. Structure of an artificial neural network [21]

An artificial neural network behaves the same way. It works on three layers. The input layer takes input (much like dendrites). The hidden layer processes the input (like soma and axon). Finally, the output layer sends the calculated output (like dendrite terminals) shown in fig. 17 [21]. There are basically three types of artificial neural network: supervised, unsupervised and reinforcement [22].

1) Supervised Neural Network: In the supervised neural network, the output of the input is already known. The predicted output of the neural network is compared with the actual output. Based on the error, the parameters are changed, and then fed into the neural network again. Fig. 18 will summarize the process. Supervised neural network is used in feed forward neural network.

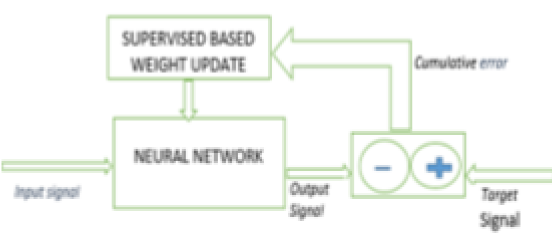

Figure. 18. Supervised neural network [22]

2) Unsupervised Neural Network: Here, the neural network has no prior clue about the output the input. The main job of the network is to categorize the data according to some similarities. The neural network checks the correlation between various inputs and groups them. The schematic diagram is shown in Fig. 19.

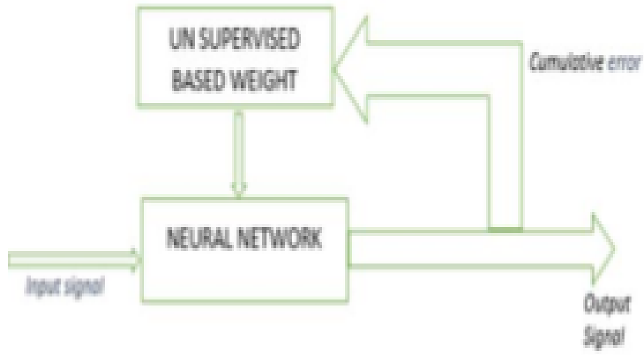

Figure.19. Unsupervised neural network [22]

3) Reinforced Neural Network: In reinforced neural network, the network behaves as if a human communicates with the environment. From the environment, a feedback has been provided to the network acknowledging the fact that whether the decision taken by the network is right orwrong. If the decision is right, the connection which points to that particular output is strengthened. The connections are weakened otherwise. The network has no previous information about the output. Reinforced neural network is represented in Fig. 20.

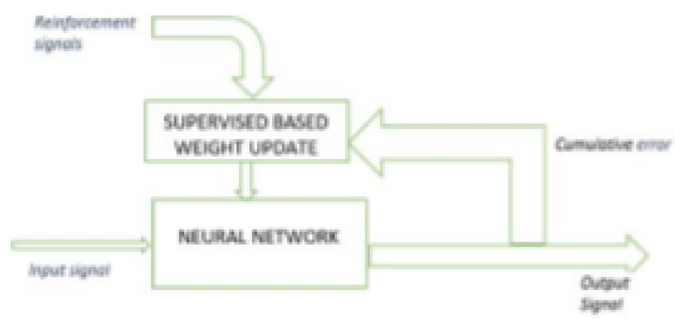

Figure .20. Reinforced neural network [22]

H. Instance-Based Learning

In instance-based learning, the learner learns a particular type of pattern. It tries to apply the same pattern to the newly fed data. Hence the name instance-based. It is a type of lazy learner which waits for the test data to arrive and then act on it together with training data. The complexity of the learning algorithm increases with the size of the data. Given below is a well-known example of instance-based learning which k-nearest neighbor [7] is.

1) K-Nearest Neighbor: In k-nearest neighbor (or KNN), the training data (which is well-labeled) is fed into the learner. When the test data is introduced to the learner, it compares both the data. k most correlated data is taken from training set. The majority of k is taken which serves as the new class for the test data [23]. The pseudo code for KNN is given in Fig. 21.

```
Let W = {x1, x2, ..., xn} be a set of n labeled samples. The
algorithm is as follows:

BEGIN
  Input y, of unknown classification.
  Set K, 1 ≤ K ≤ n.
  Initialize i = 1.
  DO UNTIL (K-nearest neighbors found)
    Compute distance from y to xi.
    IF (i ≤ K) THEN
      Include xi in the set of K-nearest neighbors
    ELSE IF (xi is closer to y than any previous nearest
      neighbor) THEN
      Delete farthest in the set of K-nearest neighbors
      Include xi in the set of K-nearest neighbors.
    END IF
    Increment i.

END DO UNTIL
Determine the majority class represented in the set of K-
nearest neighbors.
IF (a tie exists) THEN
  Compute sum of distances of neighbors in each class
  which tied.
  IF (no tie occurs) THEN
    Classify y in the class of minimum sum
  ELSE
    Classify y in the class of last minimum found.
  END IF
ELSE
  Classify y in the majority class.
END IF
END
```

Figure. 21. Pseudo code for k-nearest neighbor [24]

III. DRIVERS OF MACHINE LEARNING PROGRESS

The past decade has seen rapid growth in the ability of networked and mobile computing systems to gather and transport vast amounts of data, a phenomenon often referred to as "Big Data." The scientists and engineers who collect such data have often turned to machine learning for solutions to the problem of obtaining useful insights, predictions, and decisions from such data sets. Indeed, the sheer size of the data makes it essential to develop scalable procedures that blend computational and statistical considerations, but the issue is more than the mere size of modern data sets; it is the granular, personalized nature of much of these data. Mobile devices and embedded computing permit large amounts of data to be gathered about individual humans, and machine-learning algorithms can learn from these data to customize their services to the needs and circumstances of each individual. Moreover, these personalized services can be connected, so that an overall service emerges that takes advantage of the wealth and diversity of data from many individuals while still customizing to the needs and circumstances of each. Instances of this trend toward capturing and mining large quantities of data to improve services and productivity can be found across many fields of commerce, science, and government. Historical medical records are used to discover which patients will respond best to which treatments; historical traffic data are used to improve traffic control and reduce congestion; historical crime data are used to help allocate local police to specific locations at specific times; and large experimental data sets are captured and curated to accelerate progress in biology, astronomy, neuroscience, and other data intensive empirical sciences. We appear to be at the beginning of a decades-long trend toward increasingly data-intensive, evidence-based decision making across many aspects of science, commerce, and government. With the increasing prominence of large-scale data in all areas of human endeavor has come a wave of new demands on the underlying machine learning algorithms. For example, huge data sets require computationally tractable algorithms, highly personal data raise the need for algorithms that minimize privacy effects, and the availability of huge quantities of unlabeled data raises the challenge of designing learning algorithms to take advantage of it. The next sections survey some of the effects of these demands on recent work in machine-learning algorithms, theory, and practice.

IV .CORE METHODS AND RECENT PROGRESS

One high-impact area of progress in supervised learning in recent years involves deep networks, which are multilayer networks of threshold units, each of which computes some simple parameterized function of its inputs [25, 26].Deep learning systems make use of gradient-based optimization algorithms to adjust parameters throughout such a multilayered network based on errors at its output. Exploiting modern parallel computing architectures, such as graphics processing units originally developed for video gaming, it has been possible to build deep learning systems that contain billions of parameters and that can be trained on the very large collections of images, videos, and speech samples available on the Internet. Such large-scale deep learning systems have had a major effect in recent years in computer vision [27] and speech recognition [28], where they have yielded major improvements in performance over previous approaches shown in Fig. 22. Deep network methods are being actively pursued in a variety of additional applications from natural language translation to collaborative filtering.

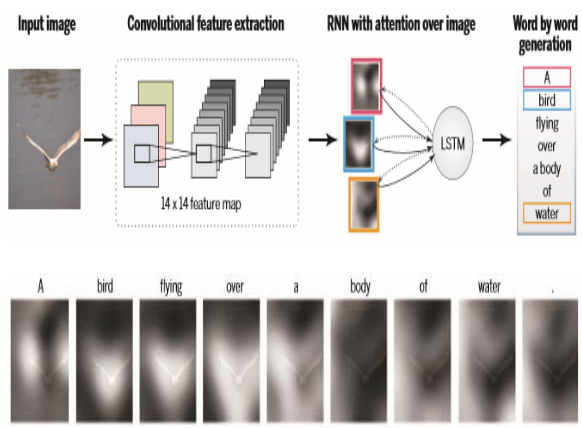

Figure.22. Automatic generation of text captions for images with deep networks

V. EMERGING TREND

The field of machine learning is sufficiently young that it is still rapidly expanding, often by inventing new formalizations of machine-learning problems driven by practical applications. (An example is the development of recommendation systems, as described in Fig. 23. A recommendation system is a machine-learning system that is based on data that indicate links between a set of users (e.g., people) and a set of items (e.g., products). A link between a user and a product means that the user has indicated an interest in the product in some fashion (perhaps by purchasing that item in the past).The machine learning problem is to suggest other items to a given user that he or she may also be interested in, based on the data across all users.

VI. OPPORTUNITIES AND CHALLENGES

Despite its practical and commercial successes, machine learning remains a young field with many underexplored research opportunities. Some of these opportunities can be seen by contrasting current machine-learning approaches to the types of learning we observe in naturallyoccurring systems such as humans and other animals, organizations, economies, and biological evolution.

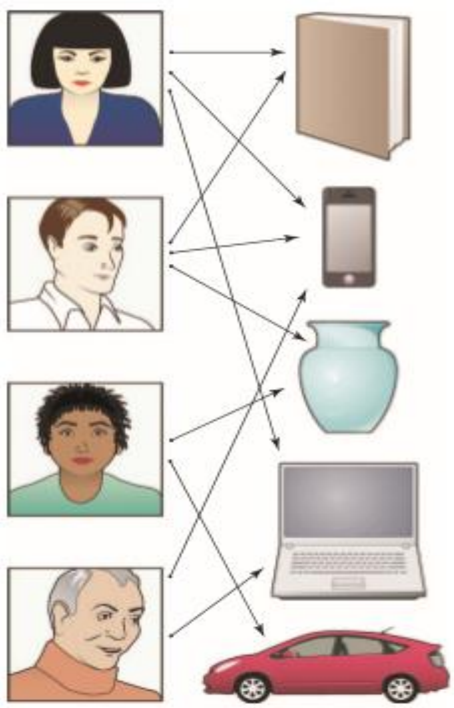

Figure. 23. Recommendation systems

For example, whereas most machinelearning algorithms are targeted to learn one specific function ordata model fromonesingle data source, humans clearly learn many different skills and types of knowledge, from years of diverse training experience, supervised and unsupervised, in a simple-to-more-difficult sequence (e.g., learning to crawl, then walk, then run). This has led some researchers to begin exploring The question of how to construct computer lifelong or never-ending learners that operate nonstop for years, learning thousands of interrelated skills or functions within an overall architecture that allows the system to improve its ability to learn one skill based on having learned another [29,30]. Another aspect of the analogy to natural learning systems suggests the idea of team-based, mixed-initiative learning. For example, whereas current machine learning systems typically operate in isolation to analyze the given data, people often work in teams to collect and analyze data (e.g., biologists have worked as teams to collect and analyze genomic data, bringing together diverse experiments and perspectives to make progress on this difficult problem). New machine-learning methods capable of working collaboratively with humans to jointly analyze complex data sets might bring together the abilities of machines to tease out subtle statistical regularities from massive data sets with the abilitiesof humans to drawon diverse background knowledge to generate plausible explanations and suggest new hypotheses. Many theoretical results inmachine learning apply to all learning systems, whether they are computer algorithms, animals, organizations, or natural evolution. As the field progresses, we may see machine-learning theory and algorithms increasingly providing models for understanding learning in neural systems, organizations, and biological evolution and see machine learning benefit from ongoing studies of these other types of learning systems.

VII. CONCLUSION

This paper surveys various machine learning algorithms, progress and trends in machine learning with their applications. Today each and every person is using machine learning knowingly or unknowingly. Machine learning is useful 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 and emerging trends.

VIII.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]. M. Bowles, “Machine Learning in Python: Essential Techniques for Predictive Analytics”, John Wiley & Sons Inc., ISBN: 978-1-11896174-2

[4]. S.B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques”, Informatica 31 (2007) 249-268

[5]. L. Rokach, O. Maimon, “Top – Down Induction of Decision Trees Classifiers – A Survey”, IEEE Transactions on Systems,

[6]. D. Lowd, P. Domingos, “Naïve Bayes Models for Probability Estimation”

[7]. AyonDey, “Machine Learning Algorithms: A Review” International Journal of Computer Science and Information Technologies, Vol. 7 (3), ISSN:0975-9646, 1174-1179, 2016

[8]. D. Meyer, “Support Vector Machines – The Interface to libsvm in package e1071”, August 2015

[9]. 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

[10].http://www.simplilearn.com/what-is-machine-learning-and-why-itmatters-article

[11]. P. Harrington, “Machine Learning in action”, Manning Publications Co., Shelter Island, New York, 2012

[12]. K. Alsabati, S. Ranaka, V. Singh, “An efficient k-means clustering algorithm”, Electrical Engineering and Computer Science, 1997

[13]. M. Andrecut, “Parallel GPU Implementation of Iterative PCA Algorithms”, Institute of Biocomplexity and Informatics, University of Calgary, Canada, 2008

[14]. X. Zhu, A. B. Goldberg, “Introduction to Semi – Supervised Learning”, Synthesis Lectures on Artificial Intelligence and Machine Learning, 2009, Vol. 3, No. 1, Pages 1-130

[15]. X. Zhu, “Semi-Supervised Learning Literature Survey”, Computer Sciences, University of Wisconsin-Madison, No. 1530, 2005

[16]. R. S. Sutton, “Introduction: The Challenge of Reinforcement Learning”, Machine Learning, 8, Page 225-227, Kluwer Academic Publishers, Boston, 1992

[17]. L. P. Kaelbing, M. L. Littman, A. W. Moore, “Reinforcement Learning: A Survey”, Journal of Artificial Intelligence Research, 4, Page 237-285, 1996

[18]. R. Caruana, “Multitask Learning”, Machine Learning, 28, 41-75, Kluwer Academic Publishers, 1997

[19]. D. Opitz, R. Maclin, “Popular Ensemble Methods: An Empirical Study”, Journal of Artificial Intelligence Research, 11, Pages 169198, 1999

[20]. Z. H. Zhou, “Ensemble Learning”, National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

[21]. V. Sharma, S. Rai, A. Dev, “A Comprehensive Study of Artificial Neural Networks”, International Journal of Advanced Research in Computer Science and Software Engineering, ISSN 2277128X, Volume 2, Issue 10, October 2012

[22]. S. B. Hiregoudar, K. Manjunath, K. S. Patil, “A Survey: Research Summary on Neural Networks”, International Journal of Research in Engineering and Technology, ISSN: 2319 1163, Volume 03, Special Issue 03, pages 385-389, May, 2014

[23]. P. Harrington, “Machine Learning in Action”, Manning Publications Co., Shelter Island, New York, ISBN 9781617290183, 2012

[24]. J. M. Keller, M. R. Gray, J. A. Givens Jr., “A Fuzzy K-Nearest Neighbor Algorithm”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15, No. 4, August 1985.

[25]. J. Schmidhuber,“Neural Network”. 61, 85 –117, 2015.

[26] .Y. Bengio, “Foundations and Trends in Machine Learning” 2 (Now Publishers, Boston, 2009), pp. 1–127.

[27]. A. Krizhevsky, I. Sutskever, G. Hinton, “Adv. Neural Information Process System”. 25, 1097–1105 (2015).

[28]. G. Hinton, Signal Processing Management, IEEE 29, 82 – 97 (2012).

[29]. T. Mitchel, Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI-15), 25 to 30 January 2015, Austin, TX.

[30]. S. Thrun, L. Pratt, Learning To Learn (Kluwer Academic Press, Boston, 1998).