FOG ANALYTICS - A SURVEY

C.S.R. Prabhu[1], Tony Jan[2], Mukesh Prasad[3], Vijayakumar Varadarajan[4]

[1] Quantum University Roorkee, India

[1]Director, Research, KMIT. Hyderabad, India

[2]T Melbourne Institute of Technology, Australia

[3] University of Technology Sydney, Australia

[4]Visitng Professor, Kyungpook National University, Korea

Email: csrprabhu@kmit.in[1], tjan@mit.edu.au[2], mukesh.prasad@uts.edu.au[3], vijayakumar.varadarajan@gmail.com[4]

DOI: https://doi.org/10.22452/mjcs.sp2020no1.10

ABSTRACT

Fog computing has emerged as an essential alternative to the cloud. Fog computing is the future as it is nearer to the edge where actually the IOT devices and sensors are located. A Fog Server or Fog Node is located near to the IOT devices, connecting directly (wired or wireless) to them. The Fog Server has a functionality of fast accessibility to the data arising out of IOT devices or sensors, as against cloud server which may be located in data centers (near core Network Centers) located far away from the edge resulting in extreme delays in network transmission and latency, especially when the data is large volume as stream (or 'Big Data') arising out of IOT devices or sensors including cameras, etc. Real time response after completing the necessary Analytics on the data generated by IOT devices and sensors becomes critically essential for meeting the real time response requirements of critical applications such as in health care and transportation. What are the relevant techniques for Fog Analytics? In this paper we provide a brief survey of Fog Analytics techniques in stream data analytics, machine learning, deep learning techniques and also game theoretical adversarial learning.

Keywords: ***Fog computing, IoT, Analytics***

1.0 INTRODUCTION

While it is well-known that the IoT will generate massive, dynamic and heterogeneous data for Big Data Analytics, the Fog Computing architecture is becoming an essential requirement in IoT applications. If Big Data Analytics technologies depend on the cloud only, it is now emerging very clearly that it will be operationally infeasible, given the high volumes ofdata (generated by the IoT devices and sensors) and the high latency of the networks and Internet required in processing of huge data to reach the cloud from the edge and come back to the edge with the decision derived out of the analytics performed in the cloud, as depicted in Figure 1. The Fog Server, located very near to the edge devices, is capable of completing majority of the analytics functions locally. This will result in benefits such as local, fast processing, storage for geo-deducible and latency sensitive applications, drastically reduced communication overheads over the network and the Internet, thereby resulting in a substantially reduced volume and velocity of data that will finally go to the cloud [1] (only aggregation or summaries).

Fog Applications such as augmented Reality, Interactive Gaming and Event Monitoring require data stream processing, in contrast to the usually assumed ready data bank in conventional Hadoop kind of Big Data Application Systems. Stream data, arriving out of sensors and IoT devices is characterized by huge volumes of continuous data. It is not possible to store the entire data at one location and at one time for processing. The stream data examples are abundant including: RFID data, IoT device generated data and security monitoring data.

telecommunication call records, credit card transaction flows, and weblogs etc. Data streams are uniquely characterized with the transient nature of data, continuous queries, multi GB arrival rates having real time response requirements [2]. Such streams require feature extraction and classification performed by Fog Systems as well. Tools such as Tensor Flow or Keras provide for the implementation of advanced data mining and machine learning algorithms such as deep neural networks in the Fog Servers. Even then issues of load balancing among multiple Fog Servers continue to exist in performance guarantee for network resources provisioning [3]. We also have the well- known stream processing engines such as Apache Storm, Spark streaming that can be executed on the Fog Servers [4].

2.0 MACHINE LEARNING FOR FOG - CHARACTERISTICS

The characteristics of the machine learning techniques required for IoT fog network are:

1) **Scalability**: as the IoT is deployed in massive numbers in the forms of small Cyber Physical Devices (CPD), the data generated by the CPD is expected to be massive [5]. The machine learning model based on Fog computing must be able to handle massive data.

2) **High Speed**: the IoT edge devices are expected to generate high velocity data. The machine learning model for fog network must be able to handle and digest high velocity data in real time and deliver acceptable and timely decisions without degradation of the network performance.

3) **Incrementality**: the IoT edge devices are expected to generate highly heterogeneous data. Many of the small unworthy computing devices are converted to CPDs with new cyber-communication capacity; therefore, the data generated by the new CPDs is expected to be highly heterogeneous and inconsistent. The machine learning model in the Fog network must be able to handle the heterogeneity of the data and handle the inconsistency reliably by stable incremental learning.

4) **Distributed Processing**: as the fog computing devices are limited in its computing resources, the data analytic tasks must be split and run in distributed mode over the multiple virtualized small servers. The machine learning model in the Fog must be able to handle distributed data processing with reliable partitioning and reassembling of data in learning.

3.0 MACHINE LEARNING REVIEW FOR FOG

In this section, we examine the available machine learning methods for IoT and Fog analytics,

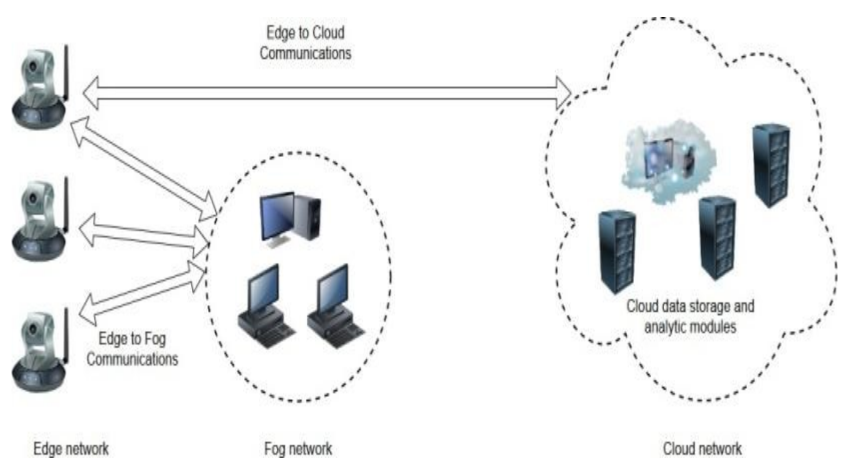

Fig. 1: Fog computing

A. Feature Selection

The IoT edge will include many small Sensors or Cyber Physical Devices (CPDs) which will generate complex data with high dimensionality because of their diverse and heterogeneous operational protocols. The data is likely to be heterogeneous and there is a need for feature selection process to reduce the dimensionality of the data [6], provided the machine learning process takes place within Fog network with limited resources. Feature selection results in identifying the most important features and drop redundant features. The recent approach in Deep Learning assumes unsupervised learning to identify significant features with an interpretable significance. Accordingly, the efficiency of predictive analytics algorithms improves substantially.

B. Supervised Learning

In supervised learning, using labeled training examples, the algorithm is trained and based on knowledge gained from the available training instances, the algorithm will predict class labels of test instances. Such supervised learning techniques can be either continuous models of regression or classification models. Recent and more advanced developments in supervised learning include distributed and parallel method of learning such as Multi hyper plane model Machine (M) Classification Model [7], Divide and Conquer SVM [8] and Neural Network classifiers. From all of them SVM is noted to be efficient within its own constraints. Access modified SVMs such as New Primal SVM [9] have emerged.

C. Distributed Decision Trees

As the Fog network is limited in computational resources, it is worth considering the popular decision trees and decision forest. The decision forest is known to provide an adequate decision support with moderate computational requirements, though their stability in dynamic data condition is a concern. The advances show that Gradient Boosted Decision Trees (GBDT) [10] runs the process of induction in a distributed manner. Reliable High Speed Decisions Tree is prepared by Calaway et al. [11] . Hall et al. [12] proposed a modified decision tree algorithm that generates the rules from a set of decision trees (connected in parallel) with traceable training set. The great advantage of interpretability of the decision tree may be of significant importance in the future IoT edge applications.

D. Clustering Methods

Parallel Clustering can handle huge volumes of data. Incremental clustering techniques can handle high velocity (stream) data. K-mode operates on categorical data and K-prototype methods [13] operate on mixed data. A variant of them by Ordonez [14] minimizes memory requirements. Bradley et al. [15] propose a framework to iteratively perform sampling from large data-set and in each iteration a model is improved to produce the final clusters.

In Wave Cluster [16] approach, the spatial domain is converted to frequency domain. Zhao et al. proposed [17] a parallel k-mean algorithm. By finding clusters distributed on multiple systems and then merging the results PDBSCAN [18] offers a distributed algorithm for clustering. Objects are partitioned by P-Cluster [19] for minimization of errors.

P-BIRCH is parallel version of BIRCH for shared nothing architectures where the incoming data is continuously distributed across multiple processors. Incremental k-means algorithm is proposed by Chakraborthy [20] for computing new centers of clusters. Widyantoro [21] proposed incremental hierarchical clustering. Density-based clustering is proposed in IGDCA. [22]. Kailing [23] proposed a clustering method with multi view as its scope. Zeng et al. [24] proposed separate clustering in different feature spaces. Chaudhuri et al. [25] and Kumar et al. [26] suggested a multi-view clustering procedure to elevate multi-views to the lower dimensional space.

E. Parallel and Distributed Association Rule Mining

As sequential association rule mining techniques will not be capable of handling geographical spread, variety, volume and velocity of data, parallel distributed high performance association rule mining methods have been developed. Count Distribution [27] performs parallelization which is meant for the conventional sequential a priori algorithm for association rule mining. Only counts are distributed across processors, instead of data, thus minimizing communication costs. Further, it has a weakness of not utilizing memory efficiently, as it replicates entire hash tree in each processor. PDM [28] is based on DHP. FDM [29] is fast distributed mining based on count distribution [27]. FPM stands for Fast Parallel Mining.

A. Dynamic Association Mining

It is assumed in all association rule mining techniques that the data sets under process are static. In reality the datasets are dynamic. Changes in data can invalidate the conclusions of previously determined association rule mining. In Fast Update (FUP) [30] large item sets are computed in a dataset that is updated regularly. Borders [31] works on consumer border sets [32].

The Decrement Updating Algorithm (30), tries to detect dynamically deleted databases. The incremental method that generates frequent sets is described in Borders (31) algorithm for consumer border sets introduced in (32).

Big Data paradigm presses the demand for an integrated solution encompassing almost all the ways of approaches to handle dynamic, large, and heterogeneous data. For quick handling of voluminous and varied data, an association rule mining techniques are used. The GRN based gene representation in System Biology, where the dynamic behaviour of the group of genes and how it is influencing the other genes seen on steady-state time series data(52). This shows the immediate need of the scalable GRN reconstruction method that can now work to infer the most reliable GRNs. By comparing across normal and diseased networks it is possible to identify potential drug targets for the target diseases (53).

B. Virtualization

VMs are deployed to improvise within the Infrastructural constraints of Fog Server Ecosystem. A Fog server has limited computational capability. If not virtualized, it will have very limited sequential processing functionality. By virtualization and by down loading VMs from its own cloud source, it is possible to have multiple logical or virtual servers in each Fog Server which can then be deployed for parallel processing with the measure of coordination.

C. Deep Learning

Deep Learning techniques [33] are a class of machine learning algorithms which have many stages of non-linear information processing in hierarchical architectures deployed for classification of patterns and learning of features, automatically (instead of doing the same manually as done in conventional machine learning techniques). Deep Learning aims at modeling high level abstractions in data. These techniques use both supervised and unsupervised learning algorithms for this purpose for learning multiple levels of abstraction. Deep Learning is also related to representation learning wherein a hierarchy of high level features or concepts is defined from the lower level ones. They are classified into (a) generative, (b) discriminative and (c) hybrid models.

The multiple levels of abstractions in Deep learning using supervised and/or unsupervised learning algorithms are to model high-level abstractions in data for hierarchical representation for data classification. Deep learning methodologies have been deployed in many applications, viz. computer vision, natural language processing, pattern and speech recognition. Deep learning algorithms that is beneficial for learning, when handling huge amount of unsupervised data and the representation of the data by greedy layer wise [54,55]. Many experiments have shown that the data representations which are received from piling up nonlinear extractors as to examine Deep learning algorithm often shown better machine learning results. The improved classification[58], gives good quality of samples which are generated by generative probabilistic models[57]. Deep learning algorithm's efficient results in separate machine learning application areas that include computer vision, natural language processing and speech recognition. Due to the possibility of exponential increase of data, the Deep Learning algorithms provide efficient results, is used to predict the accuracy of voluminous data. In recent days researchers have developed the effective and scalable solutions using parallel algorithms for training deep models. Many institutions and organizations use deep learning algorithms for information retrieval, decision making and semantic indexing. The Advantage of deep learning algorithm is to automate the process for extraction of representations (abstractions) from the data [59,60,61]. it also uses a large amount of unsupervised data for complex set of representation. The deep learning algorithms are highly used by Artificial Intelligence (AI), that has the behaviour model of human brains as the ability to observe, learn, analyze and decisions making for the extremely new complex problems. Deep Learning algorithms emulate the hierarchical learning approach of the human brain which is a key challenge. The models that are constructed on shallow learning architectures procedures such as a case based reasoning, decision trees and vector machines fall short, when the learning algorithm is trying to extract useful information from the complicated structures and their relations in the input corpus. Deep learning architectures have capability to generalize the non-local and global ways for creating deep learning patterns and relationships from the data. Deep learning is an important concept in the domain of artificial intelligence (AI), which gives the critical representations of the collected data that are suitable for AI tasks to make the machines independent of human knowledge as a goal. Deep learning also extracts representations of data directly from unsupervised data without human involvement. The local generalization based learning improves the number of patterns produced by using distributed representation. In Deep Learning Algorithms more abstract representations are often constructed from the less abstract ones. The Deep Learning advantage is to provide abstract representation of data that can be invariant to the input data on local changes. Learning the invariant features on the input data is a major goal in pattern recognition. The invariant representations that can also be recognized are the factors of variations in the data. AI-related tasks that use real data that comes from many resources such as light, and object shapes. Deep learning algorithm's abstract representations can separate different sources of variations in the data. Deep Learning algorithms are implemented as actually Deep architectures of the consecutive layers, in this each layer applies a nonlinear-transformation of its input that provides an assigned representation in its output data. This helps in learning complicated and abstract representation of the data which passes through the multiple transformation layers in a hierarchical manner. The captured sensory image data is fed to the first layer, which consequently produces the output of the each layer as input to its next layer. The data is partitioned as many as multiple samples to create data abstractions. In the layer architectures the intermediate layers are applied to process multiple levels predictions from the input data, and final predictions is carried out at the output layer from the outputs of its intermediate upper layers.

Deep learning algorithm represents the data inside multiple layers, it processes the high volume of data effectively, where the shallow learning method fails to explore the data pattern complexities. The deep learning is little suitable for heterogeneous and unstructured data analyzation. The Deep Learning algorithm uses stacking up nonlinear transformation layers method as the one of the idea. The layers of the data, that are more complicated with the nonlinear transformations which goes through the deep learning architecture, that were constructed, and these nonlinear transformations represents the data where deep learning is proposed as a special case for learning representation, which learns Deep architecture of the available data with the multiple layers of representations, the final designation of the input data is the data achieved through highly non-linear function. the deep architecture's transformations in the layers are non-linear transformations to extract factors from the data. The Principal Component Analysis (PCA) can not be used the transformation algorithm as it is non-linear transformation for the deep structures because of the linear transformation compositions, which yield another linear transformation that can be of no use in applying in the deep architecture.

Most commonly used Deep Learning models like Auto encoders, Deep Belief Networks, Convolution Neural Networks are generative models which are made effective by an additional unsupervised pre-training step on unlabelled training data for extracting structures and regularities in the input features followed by a top layer to perform discriminative tasks. Such generative models avoid difficulties with global optimization by introducing a stack of Restricted Boltzmann Machines (RBM) in a greedy layer by layer learning algorithm optimizing parameters on data. A Boltzmann Machine is a network of symmetrically connected, neuron like (neural network) units that make stochastic decisions about whether to be on or off. For RBM, a special case of BM with a layer of visible units and another layer of hidden units with no visible – visible or hidden-hidden connections within the same layer. Such Deep Learning models when built perfectly and fined tuned are useful for end to end learning of complex systems embedding domain knowledge and interpreting uncertainty. In a Fog Server, the Deep Learning platform like Tenor flow or Keras can be operationalized, for highly efficient and accurate results in identifying patterns and higher level abstractions from the input unlabelled data arriving out of IoT devices or sensors. Goodfellow and Shlens [34] state that Deep Learning models are vulnerable to adversarial examples. Adversarial examples are generated by applying small and intentional worst care perturbation to cross validation data. Thus, the perturbed input results in incorrect output with high level of confidence. The main cause identified for such vulnerability is the linear nature of Deep Learning mode in high dimensional search spaces. The vulnerability can be changed from linear to non-linear.

D. Adversarial Learning

Adversarial learning algorithms are specifically classified to exploit vulnerabilities in a given machine learning or deep learning algorithm. There vulnerabilities are simulated by training the learning algorithm under various attack scenario sand policies, formulated supportably by an intelligent adversary [35] such as a human hacker or even a software bot. A learning algorithm designed over adversarial settings becomes robust to such vulnerabilities in the training and testing data distributions. The various adversarial learning algorithms differ in assumptions regarding the adversary's knowledge, security violation, attack strategies and attack influence [36]. The goal of adversarial learning is designing robust data mining models, computing systems and machine learning algorithms for non stationary data analytics. To learn mathematical patterns in input data, the machine learning methods or algorithms make assumptions on the data distributions for training and testing the learning algorithm. Some specific Adversarial Learning algorithms (ADVMA) generate testing data distribution which is non stationary with respect to training data distribution. Adversaries can be single or multiple Deep Learning Algorithms have been developed for multiple adversaries [37] based on game theory.

Adversarial learning has application in areas including spam filtering, virus detection, intension detection, fraud detection, biometric authentication, network protocol verification, computation advertising, recommender systems social media web mining and performance modeling of complex systems.

The Fog ecosystem with its IoT Application does provide for most of not all of these above mentioned applications.

4.0 DISCUSSION ON FOG ANALYTICS

In this section, we review the machine learning methods in terms of their usefulness in the IoT Fog computing.

A. Scalability

The majority of machine learning algorithms work well with the limited dataset; however, as the data size increases, many of the machine learning methods do not perform appropriately. IoT network is expected to be the main source of big data in the next generation of computing and the new machine learning methods are expected to be able to digest and interpret massive flow of data with are reliable accuracy.

B. High Velocity

The future of data analytics is to analyze streaming high velocity data. The data may be multimedia or transactional data from major retail shops. When we have only the limited view of the data because of its high velocity, we need an intelligent machine learning solution that can generalize well to provide robust learning in real time HIGHVELOCITY applications.

C. Incremental Learning

The learning in real-time big data analytics is to be incremental. There may not be sufficient resources to perform global learning in the IoT fog network, and as such, we are limited to learning in incremental mode with adaptation to the incoming flow of new data. The online incremental learning is a key to sustain accurate modeling in a highly volatile future IoT network.

D. Distributed Processing

As the fog devices will be limited in their computational capacity, it is inevitable that we have to rely on distributed processing. Some machine learning methods are more adamant to distributed processing whereas some machine learning methods are not suitable for distributed processing.

The Machine learning algorithm methods of the different categories, like deep learning will be applied for various data streams analysis purposes.

The main nature of Fog needs a computing paradigm which provides solution for latency, and intelligent control for data analytics for highly intelligent decision making. The Fog computing performs the latency sensitive applications at the end of the network called edge network. However many of the latency tolerant tasks for deep analysis [38] are computed at the Cloud data center.

The main advantage of the Cloud computing is its on-demand, the scalable or incremental storage and the services that can accommodate IoT based applications. Some IoT applications like connected vehicle, urgent response, healthcare, and applications where latency is main criteria, where transferring data delay to the cloud from applications and then response back to application causes unacceptable [39][40][41].

The Fog Computing in time critical services is mostly cost effective when it is compared with the cloud computing because of the time delay is less. This study [40] helps us to address the number of time critical applications performed by Fog is extremely applied in power consumption and the cost.

Certain algorithms like cluster removal methods in CURE and ROCK based Hierarchical Agglomerative Clustering (HAC) algorithms, are susceptible to infeasible for data streaming as it need many forms of scanned data [42].

In the case of Parzen Probability density model, which is a memory based function and the closer neighbour function, the complete training set will be stored for future predictions using data points and a measurement unit is defined for measuring the same set of two vectors in the input data set, and these two are memory utilization methods that are slow at deciding the predictions on the test data input points. The Fog computing is used for stream data analysis.

The SVM, DT and ANN (MLP) are the regularly used machine learning procedures in the survey of IoT applications. In the analysis, it is noted that the machine learning performances are task dependent. In [50] it is pointed that good classifiers perform as per the weather environment conditions. Most of the MLP based classifiers works in a better way than SVM for foggy and sunny environment conditions. In the rainy environment conditions, the SVM will be the appropriate suitable model. In [43], it is noticed that the automatic SMM (Stereotypical Motor Movement) recognition, SVM both outperform the DT for an overall data accuracy of at least up to ~6 points percentage. In [52], the classification operation time or regression operation time is observed as low latency as compared with DT. In [51], it is noticed that SVM is not good enough in the real-time classification. The classification time is nearly 2.2 seconds. When the same classification time is compared with ANN with is all the seven nodes (hidden), the classification time is almost 100 milliseconds.

The ANN, reinforcement learning and HMM methods consume huge training data for iterations to fit into convergence on overall it is suggested that the training tasks can be deployed on the Cloud, and we may move the on-line analytics of the tasks to the Edge and /or Fog computing. For deep Neural Network learning called CNN, its weights in the layers will be trained and then the it is updated in the same way for traditional MultiLayer Perceptions (MLP) such that the weights and layers are created in the order of magnitude, which is greater than the MLPs. To train the deep NN, we may require huge amount of time and resources to accommodate training data. and it has to be performed on the Cloud. For operational latency which is corresponding to the count of the neurons irrespective of training data size, its online analysis are performed at the Edge/Fog computing.

In [48,49], the accuracy, and operational time of sliding windows sizes are calculated with the help of machine learning algorithms for predictions and forecast time lags.

The data analysis can be improved with pre-processing of dependent data. In [51], ANN classifier, which a gradient detector and also an intensity-bump detector, the loose threshold values used to filter non-lane representations, it also helps in reducing classification time because of smaller number.

The classification time also can be reduced further by deploying the parallel and distributed computing environment with the help of machine learning algorithms, which can will be parallelly executed.

The ANN is more suitable to deal data sets with multivariant compared to reinforcement method. Because the data stream behaviour can affect the selection of the methods. The fuzzy logic is one such kind of example to use fuzzy information on the large data volumes.

The Machine learning algorithms that are used for analyzing growing data streams in the Cyber Physical System (CPS), for using as the advantage of parallel and distributed computing environment of the Cloud and Fog Computing [44] for composable and hierarchical machine learning algorithms, which portioned the execution among the Cloud and Edge Computing to know the continual and transfer learning's to deal on the non-movable data streams. In the future, there should be more studies to be carried on the Cloud and Edge computing systems that help the CPS for data stream analysis to accommodate heterogeneity and discrepancy among data centers and edge devices to provide APIs[45] and the services [46,47].

5.0 CONCLUSION

In this paper we have presented a survey of possible considerations and the existing techniques of machine learning for Fog Analytics. Fog computing environment is characterized by fast analytics process in real-time on the data coming out of the IoT devices connected to the Fog server. However, the Fog server is having limited memory and CPU speed. Further, the requirements of the IoT applications have to be met in real time by the Machine Learning algorithms deployed in the Fog server. The IoT being the stream data , many conventional ML algorithms do not comply with these requirements. This paper presents an analysis of various ML algorithms in terms of their suitability for the Fog Ecosystem and its requirements. SVM is reported to outperform DT in most cases. In the context of deep learning, a brief comparison of MLP,ANN, CNN is made for selection of appropriate technique for the Fog. Whenever the training is required massive data , it will be possible to train the ML/DL algorithms only in the cloud and not in Fog. The Fog environment permits only testing execution of the ML/DL algorithms.

REFERENCES

[1] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, “Industrial IoT data scheduling based on hierarchical fog computing: a key for enabling smart factory,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4590–4602,2018.

[2] C.-H. Lu and C.-H. Yu, “Online Data Stream Analytics for Dynamic Environments Using Self-Regularized Learning Framework,” *IEEE Sys- tems Journal*,2019.

[3] J. Du, L. Zhao, J. Feng, and X. Chu, “Computation offloading and resource allocation inmixed fog/cloud computing systems with min-max fairness guarantee,” *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608,2018.

[4] Y. Kim, S. Son, and Y.-S. Moon, “SPMgr: Dynamic workflow manager for sampling and filtering data streams over Apache Storm,” *Inter- national Journal of Distributed Sensor Networks*, vol. 15, no. 7, p. 1550147719862206,2019.

[5] M. M. Rathore, A. Paul, W.-H. Hong, H. Seo, I. Awan, and S. Saeed, “Exploiting IoT and big data analytic Defining smart digital city using real-time urban data,” *Sustainable cities and society*, vol. 40, pp. 600– 610,2018.

[6] H.-N. Dai, H. Wang, G. Xu, J. Wan, and M. Imran, “Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies,” *Enterprise Information Systems*, pp. 1–25,2019.

[7] N. Djuric, “Big data algorithms for visualization and supervised learning,”2014.

[8] C.-J. Hsieh, S. Si, and I. Dhillon, “A divide-and-conquer solver for kernel support vector machines,”in *International conference on machine learning*, 2014, pp.566–574.

[9] F. Nie, Y. Huang, X. Wang, and H. Huang, “New primal SVM solver with linear computational cost for big data classifications,” in *Proceed- ings of the 31st International Conference on International Conference onMachineLearning-Volume32*,2014,p.505.

[10] J. Ye, J.-H. Chow, J. Chen, and Z. Zheng, “Stochastic gradient boosted distributed decision trees,” in *Proceedings of the 18th ACM conference onInformationandknowledgemanagement*,2009,pp.2061–2064.

[11] R. Calaway, L. Edlefsen, L. Gong, and S. Fast, “Big data decision trees with r,” *Revolution*,2016.

[12] L. O. Hall, N. Chawla, and K. W. Bowyer, “Decision tree learning on very large data sets,” in *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*,vol.3,1998,pp.2579–2584.

[13] Z. Huang, “Extensions to the k-means algorithm for clustering largedata sets with categorical values,” *Data mining and knowledge discovery*, vol.2,no.3,pp.283–304,1998.

[14] C.OrdonezandE.Omiecinski,“Efficientdisk-basedK-meansclustering for relational databases,” *IEEE Transactions on Knowledge & Data Engineering*,no.8,pp.909–921,2004.

[15] P. S. Bradley, U. M. Fayyad, C. Reina *et al.*, “Scaling Clustering Algorithms to Large Databases.” In *KDD*,vol.98,1998,pp.9–15.

[16] G. Sheikholeslami, S. Chatterjee, and A. Zhang, “WaveCluster: a wavelet-based clustering approach for spatial data in very large databases,” *The VLDB Journal—The International Journal on Very LargeDataBases*,vol.8,no.3-4,pp.289–304,2000.

[17] W. Zhao, H. Ma, and Q. He, “Parallel k-means clustering based onmapreduce,” in *IEEE International Conference on Cloud Computing*, 2009, pp.674–679.

[18] X. Xu, J. Jäger, and H.-P. Kriegel, “A fast parallel clustering algorithm forlargespatialdatabases,”pp.263–290,1999.

[19] D. Judd, P. K. McKinley, and A. K. Jain, “Large-scale parallel data clustering,” *IEEE Transactions on Pattern Analysis and Machine Intel- ligence*,vol.20,no.8,pp.871–876,1998.

[20] S. Chakraborty and N. K. Nagwani, “Analysis and study of incremental k-means clustering algorithm,” in *International Conference on High PerformanceArchitectureandGridComputing*,2011,pp.338–341.

[21] D. H. Widyantoro, T. R. Ioerger, and J. Yen, “An incremental approach to building a cluster hierarchy,” in *2002 IEEE International Conference onDataMining,2002.Proceedings.*,2002,pp.705–708.

[22] N. Chen, A. zhou Chen, and L. xiang Zhou, “An incremental grid density-based clustering algorithm,”*Journal of software*, vol. 13, no. 1, pp. 1–7,2002.

[23] K. Kailing, H.-P. Kriegel, A. Pryakhin, and M. Schubert, “Clustering multi-represented objects with noise,” Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2004, pp. 394–403

[24] H.-J. Zeng, Z. Chen, and W.-Y. Ma, “A unified framework for clustering heterogeneous web objects,” in Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002., 2002, pp. 161–170.

[25] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, “Multi- view clustering via canonical correlation analysis,” in Proceedings of the 26th annual international conference on machine learning, 2009, pp. 129–136.

[26] A. Kumar and H. Daumé, “A co-training approach for multi-view spectral clustering,” in Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 393–400.

[27] R. Agrawal and J. C. Shafer, “Parallel mining of association rules,” IEEE Transactions on Knowledge & Data Engineering, no. 6, pp. 962–969, 1996.

[28] J. S. Park, M.-S. Chen, and P. S. Yu, “E cient parallel data mining for association rules,” in Proceedings of the 4th Int’l Conf. on Information and Knowledge Management, 1995.

[29] D. W. Cheung, J. Han, V. T. Ng, A. W. Fu, and Y. Fu, “A fast dis- tributed algorithm for mining association rules,” in Fourth International Conference on Parallel and Distributed Information Systems, 1996, pp. 31–42.

[30] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka, “An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases.” in KDD, 1997, pp. 263–266.

[31] Y. Aumann, R. Feldman, O. Lipshtat, and H. Manilla, “Borders: An efficient algorithm for association generation in dynamic databases,” Journal of Intelligent Information Systems, vol. 12, no. 1, pp. 61–73, 1999.

[32] H. Mannila and H. Toivonen, “Levelwise search and borders of theories in knowledge discovery,” Data mining and knowledge discovery, vol. 1, no. 3, pp. 241–258, 1997.

[33] L. Deng, “Three classes of deep learning architectures and their applica- tions: a tutorial survey,” APSIPA transactions on signal and information processing, 2012.

[34] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” arXiv preprint arXiv:1412.6572, 2014.

[35] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, “Adversarial machine learning,” in Proceedings of the 4th ACM workshop on Security and artificial intelligence, 2011, pp. 43–58.

[36] B. Biggio, G. Fumera, and F. Roli, “Security evaluation of pattern classifiers under attack,” IEEE transactions on knowledge and data engineering, vol. 26, no. 4, pp. 984–996, 2013.

[37] A. S. Chivukula and W. Liu, “Adversarial Deep Learning Models with Multiple Adversaries,” IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 6, pp. 1066–1079, 2018.

[38] B. Tang, Z. Chen, G. Hefferman, S. Pei, T.Wei, H.He, Q. Yang, Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities, IEEE Trans. Ind. Informatics. 13 (2017) 2140–2150. doi:10.1109/TII.2017.2679740.

[39] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, E. Riviere, Edgecentric Computing: Vision and Challenges, ACM SIGCOMM Comput. Commun. Rev. 45 (2015) 37–42. doi:10.1145/2831347.2831354.

[40] L. Gu, D. Zeng, S. Guo, A. Barnawi, Y. Xiang, Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System, IEEE Trans. Emerg.Top.Comput. 5 (2017) 108–119. doi:10.1109/TETC.2015.2508382.B. Xu, L. Xu, H. Cai, L. Jiang, Y. Luo, Y. Gu, The design of an m-Health monitoring system based on a cloud computing platform, Enterp. Inf. Syst. 11 (2017) 17– 36. doi:10.1080/17517575.2015.1053416.

[41] Q. Tu, J.F. Lu, B. Yuan, J.B. Tang, J.Y. Yang, Density-based hierarchical clustering for streaming data, Pattern Recognit.

[42] Lett. 33 (2012) 641–645. doi:10.1016/j.patrec.2011.11.022.

[43] M.S. Goodwin, M. Haghighi, Q. Tang, M. Akcakaya, D. Erdogmus, S. Intille, Moving towards a real-time system for automatically recognizing stereotypical motor movements in individuals on the autism spectrum using wireless accelerometry, in: Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. - UbiComp '14 Adjun., ACM Press, New York, New York, USA, 2014: pp. 861–872. doi:10.1145/2632048.2632096.

[43] N. Verba, K.-M.Chao, A. James, D. Goldsmith, X. Fei, S.-D. Stan, Platform as a service gateway for the Fog of Things, Adv. Eng.Informatics. 33 (2017) 243–257. doi:10.1016/j.aei.2016.11.003.

[44] I. Stoica, D. Song, R.A. Popa, D. Patterson, M.W. Mahoney, R. Katz, A.D. Joseph, M. Jordan, J.M. Hellerstein, J. Gonzalez, K. Goldberg, A. Ghodsi, D. Culler, P. Abbeel, A Berkeley View of Systems Challenges for AI, 2017. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-159.html.

[45] G. Baryannis, K. Kritikos, D. Plexousakis, A specification-based QoS-aware design framework for service-based applications, Serv. Oriented Comput. Appl. 11 (2017) 301–314. doi:10.1007/s11761-017-0210-4.

[46] W. Wang, K. Lee, D. Murray, A global generic architecture for the future Internet of Things, Serv. Oriented Comput. Appl. 11 (2017) 329–344. doi:10.1007/s11761-017-0213-1.

[47] J. Ferreira, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, G. Pessin, Driver behavior profiling: An investigation with different smartphone sensors and machine learning, PLoS One. 12 (2017)e0174959. doi:10.1371/journal.pone.0174959.

[48] P. Zito, Haibo Chen, M.C. Bell, Predicting Real-Time Roadside CO and CO2 Concentrations Using Neural Networks, IEEE Trans. Intell. Transp. Syst. 9 (2008) 514–522. doi:10.1109/TITS.2008.928259.

[49] F. Castano, G. Beruvides, R. Haber, A. Artunedo, Obstacle Recognition Based on Machine Learning for On- Chip LiDAR Sensors in a Cyber-Physical System, Sensors.17 (2017) 2109.doi:10.3390/s17092109.

[50] Z. Kim, Robust Lane Detection and Tracking in Challenging Scenarios, IEEE Trans. Intell. Transp. Syst. 9 (2008) 16–26. doi:10.1109/TITS.2007.908582.

[51] T.P. Vuong, Cyber-physical intrusion detection for robotic vehicles, University of Greenwich, 2017.http://gala.gre.ac.uk/17445/7/TuanVuong 2017.pdf.

[52] S.A. Thomas and Y.Jin, "Reconstructing biological gene regulatory networks: where optimization meets big data", Evolutionary Intelligence, vol.7, no.1, pp.29-47, 2014.

[53] P.B. Madhamshettiwar, S.R.Maetschke, M.J.Davis, A.Reverter, and M.A. Ragan, "Gene regulatory network inference evaluation and application to ovarian cancer allows the prioritization of drug targets", Genome medicine vol.4, no.5, pp. 1-16, 2012.

[54] Hinton GE, Osindero S, The Y-W(2006) A fast learning algorithm for deep belief nets. Neural Compute 18 (7): 1527-1554.

[55] Bengio Y. Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks, Vol.19.

[56] Larochelle H, Bengio Y, Louradour J, Lamblin P (2009) Exploring strategies for training deep neural networks.J Mach Learn Res 10: 1-40.

[57] Salakhutdinov R, Hinton GE (2009) Deep Boltzmann machines. IN: International Conference on, Artificial Intelligence and Statistics. JMLR.org. pp 448-455.

[58] Goodfellow I, Lee H, Le QV, Saxe A, Ng AY (2009) Measuring invariances in deep networks. In: Advances in Neural Information Processing Systems Curran Associates, inc. pp.646-654.

[59] Bengio Y. Courville A, Vincent P (2013) Representation learning A review and new perspectives, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35(8): 1798-1828, doi: 10.1109/TPAMI.2013.50.

[60] Bengio Y (2009) Learning Deep Architectures for AI, Now Publishers Inc. Hanover, MA.USA.

[61] Bengio Y (2013) Deep Learing of representations: Looking forward in: Proceedings of the 1st international Confernece on Statistical Language and Speech Processing SLSP'13, Springer, Tarragone, Spain PP 1-37 http://dx.doi.org/10.1007/978-03-562-39593-21.