Keywords
Scalable machine learning, data science, science of data,

1. INTRODUCTION
This paper is based on the experience from Big Data projects at the Oak Ridge National Laboratory (ORNL). The projects detailed in Section 2 provided the opportunity to survey the state-of-the-practice and apply state-of-the-art techniques to understand the gaps and challenges of machine learning at scale. We describe our experiences below.

We are observing a paradigm shift in the application of machine learning algorithms amongst the scientific, academic, government, and industry practitioners. Practitioners in the Big Data era are adopting what is being referred to as "the fourth paradigm" [1] of data-driven discovery as a complementary approach to time-tested popular theory, experiment, and simulation based methodologies. The classic approach to scientific discovery (understanding the "why" behind observations) would begin with building a mathematically elegant model that explains the underlying generating process and the observations. The model would then be used for making future predictions after validation. Today, we are seeing the application of data mining and machine learning algorithms in domains where we begin with the assumption that there may be no physical or mathematical model underlying the data but the desire and need to make faster (sometimes even approximate) predictions overwhelms the curiosity and effort required to understand the "why". In other words, the business opportunities in the Big Data era expect scientific thought, methodology, and metrics on problems that are not scientific or mathematically well-expressed problems. Machine learning is perceived as a potential (black-box) solution to meet the needs.

But, the Big Data challenges for data-driven discovery are manifesting in different forms for machine learning researchers. For example, the typical machine learning dataset in circa 1990 [2] consisted of a few hundred samples, tens of features and a countable (less than 5 but mostly binary) category of labels. In the Big Data era, millions if not billions of data points are available, thousands of features can be computed on those data points and thousands of category labels are commonplace [3]. With increasing hardware efficiency towards low latency retrieval, practitioners no longer want to fit a model to the data. Instead, they treat data as the model itself and argue that better data is better than better algorithms [4]. The argument is that classical learning techniques aggregate data. Models were expected to work for the average (e.g. fitting a Gaussian distribution with the mean and variance from measurements). In the Big Data era, it is not about the average or the model, but it is about every individual data point (e.g., computing the kernel density estimate for the probability distribution).

Fortunately, the ability to organize, collect, and integrate Big Data for insights has been accompanied by significant increase in computational capacity. Personal computers from being machines with a single-core processor, few megabytes (MBs) of memory, and gigabytes (GBs) of storage are now equipped with multi-core processors and thousand-fold scale up on memory and storage. We see a similar trend in the high performance computing (HPC) as well - teraflops in the 1990s to petaflops today. Such progress is encouraging machine learning researchers to think beyond the convenience, comfort, and expertise of using popular algorithm implementations [5] in tools such as MATLAB, R, etc. to build customized parallelized implementations of popular algorithms [16] or train models of increasing parametric complexity [6,7]. For example, the typical size of a neural network trained in the

1990s was in the order of a few hundred parameters. Today, billion parameter networks have been successfully trained and demonstrated to work on massive datasets [8, 9].

While all these are developments are encouraging, several questions are still left unanswered – Does increase in model complexity help us understand the data better? Does increase in model complexity provide better accuracy, precision and recall? How many different models can an algorithm learn simultaneously? How to scale up/ automate the feature engineering process? How can we recommend choice of analysis algorithms based on data? How do existing machine learning methods evolve to increasing samples, dimensionality and categories over time? Our approach with this paper is not to provide answers, but to expose the daunting challenges that level-set the big expectations on machine learning while at the same time revealing the opportunities with potential for significant impact through future research.

2. BIG DATA PROJECTS AT ORNL

This paper draws experience from three Big Data projects described in this Section. We abstract the common underlying challenges and present them in Section 3.

2.1 Healthcare

In 2011, United States (U.S.) Department of Energy's Oak Ridge National Laboratory and the Centers for Medicaid and Medicaid Services under the Department of Human Health Services collaborated via an inter-agency agreement to explore data-science and knowledge discovery opportunities in healthcare. At that time, ORNL possessed some the world's best computing resources and the Department of Human Health Services was hosting and processing the world's largest digital archive of healthcare transactions. The challenge for the inter-agency partnership was to leverage 'Big Health Data' towards smarter healthcare by discovering opportunities for better policy, quality and integrity. In other words, the challenge was to transform claims-oriented data to actionable knowledge for improving the quality of healthcare (cost-care optimization problem), detecting and preventing fraud, waste and abuse (data mining problem), and finding data-driven evidence (searching for trends, patterns and correlations) for aggressive pro-active policy decisions towards making healthcare affordable.

It was a complex 'Big Data' challenge in every possible clichéd way – volume, velocity, variety, and value. We were looking at multiple years of healthcare transactions that summed up more than a trillion dollars for a year. A population of over 100 million eligible beneficiaries generating more than a billion claims (at the rate of approximately 30 claims/second) for thousands of diagnostic and treatment procedures. In addition, we had to deal with variety from different warehouses streaming different types of datasets (transaction claims vs. electronic health records) in different schemas from different operational schemes (Medicare vs. Medicaid), different data models (Texas fee-for-service vs. Tennessee managed-care) and different formats (mainframes vs. modern databases). To tackle this Big Data problem with national and societal impact, we designed a knowledge discovery infrastructure (KDI) – a hardware platform and an analyst sandbox that supported data integration workflows, traditional retrieval-based analysis and modern algorithms – based on the design principles in [19, 20]. On this infrastructure, we implemented a software suite of algorithms for descriptive, diagnostic, predictive and prescriptive analysis. These algorithms doubled in application while staging datasets for integration (quality checks, identify disambiguation), predictive modeling (feature extraction stages) and as components of the predictive model itself. The detailed description of the algorithms and the results from several data scientists that helped tackle the Big Data challenge is documented in papers [18-22].

2.2 National Security and Law Enforcement

In 2012, after the Boston Marathon Bombing, the FBI was quickly inundated with multiple terabytes of videos, photos, tips, and social media data of the bombing event. Analyzing this data required a team of agents to manually review for clues, ultimately taking four days to identify a suspect. That four-day window provided the suspects with additional opportunities to either escape or commit additional crimes. Law enforcement agencies face a similar challenge of sifting through evidence in the fight against human trafficking. Evidence of crimes against children average four terabytes of data for each apprehended perpetrator - their hard drives include massive collections of videos and images of children being raped along with e-mail, social media connections to potential victims, and potential links to trafficking networks. Due to the limitations of existing forensic tools, there is a six-month backlog in analyzing a suspect hard drive. At ORNL, we evaluated the art-of-the-possible with machine learning to provide solutions to the common image triaging problem by proposing a system that can automatically describe or tag image content.

We conducted a feasibility study by scraping millions of images from the web (a few of them pre-labelled or tagged) and using state-of-the-art learning methods to automatically come up with a conceptual description of the image. We integrated terabytes of unstructured datasets such as WordNet [27] (~150,000 words organized in ~117,000 sets), ImageNet [28] (~10 million images with word associations to WordNet), ConceptNet [29] (~10 million sentences) with labelled samples from LabelMe [30] and MIRFLICKR [31] (~1 million images with eight tags per image) as our knowledgebase and trained learning algorithms that were then packaged into a tool that automatically provides a conceptual description of a previously unknown image. We were unable to reach the desired accuracy levels for the image understanding problem with our tool but were quite successful in the filtering and sifting required while pruning large image collections. We also gained a lot of experience in understanding the open research challenges in machine learning at scale.

2.3 Advanced Manufacturing

The Department of Energy's Manufacturing Demonstration Facility was established at ORNL to engage with the industry to evaluate modern manufacturing innovations in sensing, large-area 3D printing and additive manufacturing to prepare the country for manufacturing in the internet-of-things era where sensor technologies will significantly improve the performance of production lines. In one of the industry partnerships in 2011, we were given access to sensor-data from an operational production line from one of the Fortune-500 companies. The Big Data Challenge was the following: Given a 3 terabyte (TB) sample of data from a variety of structured and unstructured data sources such as operator control log, sensor-data (e.g. pressure, temperature, and accelerometer sensor streams) that were generated at 27 gigabytes an hour, build a predictive model for future rejects. The complexity of the problem was further

augmented by the following factors. The first one was that there were 500 possible types of rejects on each product from the quality control department – unlike the traditional binary classification (good/bad labels) we are used to in most machine learning tasks. Secondly, the production line was operating at an efficient level of yield at 99% or higher) over several hours or even days at a stretch. This meant that in spite of having TBs of data, the labelled data were not going to have enough examples to characterize the 500 different reject labels. From domain experts, we also learned that there is a hierarchy about the 500 different quality control labels.

On this data, line managers posed questions such as – What sequence of events (a temporal combination of environmental, process, operator-induced events) are leading to a particular type of product defect? Can we generate/discover such a list of hypotheses from the data? Can we mine for relationships between events and production line health assuming an unknown but finite time delay in evidence supporting a hypothesized relationship? Can we automatically associate product defects with operator maintenance events and/or data feed events? Can algorithms automatically learn high-risk changes to the production line settings (new data feed, new operating mode, etc.)? How can we quickly convert volumes of data collected to intelligent actionable items–both as preventive and condition-based management strategy for the production line? Although, traditional model-based, quantitative and qualitative approaches have attempted to answer similar questions, these techniques are exposed when asked to handle data of high volume, high feature dimensionality, and inter-relatedness amongst sub-systems.

We were able to answer most of the aforementioned questions by building a data management solution along with data analysis/machine learning algorithms. The algorithms were tailored to deal with production-line sensor-data streams that arrive at different velocities. We implemented time-synchronization algorithms, extracted statistical meta-tags as features [23] and implemented scalable multi-task learning algorithms that helped discover correlations between combinations of sensor activity and rejects [24]. This approach allowed us to both answer ad-hoc queries from the domain expert and use automated algorithms to build a comprehensive and data-adaptive predictive model with in-situ training capabilities such as the one described in [25].

3. CHALLENGES AT SCALE

Although healthcare, national security and manufacturing appear as tangential application domains, the use cases shared a core formulation of the machine learning problem statement:- Given a matrix **M** of data points ***x*** with **N** samples, along ***d*** feature dimensions and ***k*** categories or classes, find a function ***f*** that can predict categories for new samples of ***x***. Translating it into the respective domains, given longitudinal history of several patients, predict future needs – diagnoses, procedures and thereby cost for current and future patients. For the image triaging use-case, based on examples of tagged/labelled images, predict word association for new images. In the manufacturing use-case the task is to extract thousands of meta-tags from streaming sensor data to predict one of the 500 categories of rejects. The size **N**, ***d*** and ***k*** (the volume, velocity and variety) are comparable in all three applications — millions of patients making billions of claims in a year along thousands of possible diagnoses and millions of users on photo-sharing sites uploading billions of pictures with thousands of word tags. When posed with such Big Data, where the data sizes are huge and data scientists tasked with building predictive models, be it for fraud detection/prevention or for recommending healthcare products and services or recommend conceptually and perceptually similar advertisements – the state of the practice would involve defining a feature space of ***d*** dimensions as illustrated in Figure 1 that will categorize ***k*** classes to a desired level of accuracy, precision and recall.

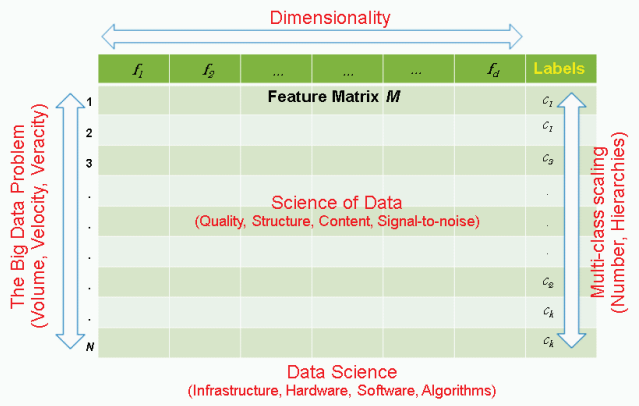

Figure 1. The challenges to Machine learning in the Big Data era.

Based on the experience from the three projects, we have abstracted three challenges to machine learning in the Big Data era. We explain each aspect of the challenge in the following sections.

3.1 Data Science

The state-of-the-practice for scalable machine learning from a data science perspective occurs in three stages– the staging for model construction, the training of the model, and deployment and evaluation of the model to future data. The data staging phase is a disk-intensive process while the model deployment and model learning stages are typically compute and memory intensive processes. Today, most of the advances for scalable machine learning (e.g. Madlib [17], Apache Mahout [12], etc.) are happening in the massively parallel database processing community. While that is progress, not all algorithms can be implemented using the set-theoretic algebra of databases. Linear-algebra based algorithms that involve a matrix inversion, Eigen-value decomposition, iterative objective-function optimization, etc., are very hard to implement at scale in databases. While some of the machine learning algorithms can be implemented on distributed storage solutions, the majority of the algorithms are better implemented as in-memory operations.

Clearly, the choice of hardware and software optimizes certain class of algorithms. These choices can only be made with guiding benchmarks. In one of our investigations for graph computing, we observed that the graph-theoretic algorithms executed orders of magnitude faster in shared-memory architectures as opposed to shared-storage systems [10]. Unfortunately, we do not have such benchmarks of machine learning algorithms executed in different scalable computer architectures to make the data science decisions for learning purposes. Today, the market availability and affordability are driving the choice of hardware configurations for deploying machine learning solutions. In spite of the need for

performance (speed-up and scale-up), we are unable to predict the performance of a data analysis algorithm like we have succeeded with traditional high-performance computing (HPC) codes. We do not understand the trade-offs of using available hardware to optimal hardware for the different classes of machine learning algorithms. This lack of predictability can be attributed to the fact that performance of data analysis algorithms is a function of architecture, data characteristics and algorithmic-workflow. Reasoning with this challenge from an Amdahl's law perspective, we are unable to guarantee expected linear/sub-linear scalability with HPC on Big Data architectures. HPC problems are forward problems better understood for parallelism and scalability while Big Data problems are inverse problems designed for functionality. Scalability of data-intensive computing algorithms is less studied and less understood. The off-the-shelf options available therefore are either to buy/invest in customized proprietary solutions that will not allow commercial third-party and open-source tools or implement/port algorithms to scale on commodity hardware – which can be a time-consuming (and often frustrating) software development process without apriori estimation of performance guarantees.

In the Big Data era, porting and deploying machine learning algorithms at scale is not a single person task anymore. It is combination of expertise in systems (infrastructure, architecture and databases), data management (provenance, governance, etc.), high performance programming model (MapReduce, MPI, CUDA, Spark [26] etc.) and query languages (SQL, SPARQL, etc.) along with algorithm design (in-memory, in-database, and in-disk) and theory (statistics, information, foraging, etc.). Although we understand that performance of a learning algorithm is dependent on the underlying architecture, most algorithm specialists are used to in-memory tools that work on desktop computers that require the entire dataset in memory. The learning curve for translating an algorithm that works with data (in-memory) on desktop computers to work optimally in shared-storage cloud architecture or shared-nothing high performance computer architectures is steep and cumbersome.

3.2 Science of Data

The second lesson we learned is that the best performing learning algorithms are ones that understand the dataset the best. This insight is critical because often practitioners treat machine learning algorithms as a black box of tools that they can apply on data. Data quality, the structure of data (matrix, schema, image, text, etc.), the organization of data all plays a significant role in the choice and design of a scalable machine learning algorithm. Applying learning algorithms without understanding the science (fundamental statistical characteristics) of the data is unfortunately common practice today. The mismatch of data characteristics with assumptions made while deriving the algorithm often leads to misleading inference. For example, most machine learning methods assume that samples are identically and independently distributed. While in most cases, classifiers and predictive algorithms may still work in a situation where samples are not independent, an anomaly detection method would produce counter-intuitive results because the probability distribution estimated with that assumption of independence will ignore the long tail or the skewness in the data resulting in spurious inference.

Also, certain algorithms have assumptions about noise levels in the data. Noise can appear in the form of sensor measurement errors, bias in sampling, bias in labeling, misleading or missing data, and a multitude of other factors. The performance difference of an algorithm on carefully curated data compared against datasets with low signal-noise ratio can be significant. For example, a learning algorithm that is trained to predict a flu outbreak from healthcare claims or drug events is a lot more trustable than training the same learning algorithm based on social media interactions and internet searches about the flu. The difference being that the correlative data sources (social media and internet search) are noisier and may not be causative data sources useful for prediction. Individuals may be searching and tweeting about the flu because they are hearing about it in the news. The data point about the search based on a news article as opposed to a symptom of flu is noise to the learning algorithm.

The ability to understand the characteristics of the data before designing or applying an algorithm is the most time-consuming task facing an analyst today. Majority of the analyst time towards building predictive models is spent in the data pre-processing and feature construction/engineering phase. The ability to characterize data with noise bounds, signal to noise ratios, properties such as stationarity, ergodicity, periodicity, self-similarity that helps interpret the underlying generating mechanism of the data is critical. Meticulous understanding of the science behind the data can make sure that we are not violating assumptions made during the algorithm design. This ability is particularly critical in the Big Data era because diverse and weakly relevant data sources collected at different levels of quality are often integrated and presented for analysis.

Another challenge for machine learning in the Big Data era is that not all data is available as a matrix $\boldsymbol{M}$ of data points $\boldsymbol{x}$ with $\boldsymbol{N}$ samples, along $\boldsymbol{d}$ feature dimensions with labels of $\boldsymbol{k}$ categories. Datasets can include unstructured data inputs such as images, text and sensors. In fact, 80% of data archived in the world today is estimated to be unstructured. Taking unstructured data sources and transforming them into a meaningful feature matrix illustrated in Figure 1 (also called the process of feature engineering) is not a trivial task. Creating structured data from unstructured sources can be domain-specific and can depend on inputs from subject matter experts on what are potential predictive observables from raw data. There is no principled automatic way to construct features from data today. Subject matter experts resort to computing linear and non-linear combinations and aggregates that encode temporal and/or spatial variations with the hope that domain-specific features will reveal more. Today, data scientists do not have approximations or a deterministic theoretical bound on the number of dimensions $\boldsymbol{d}$ required for a feature matrix $\mathbf{M}$ for a desired accuracy**.** There are no guarantees on the expected accuracy and precision and no automated way to design feature spaces given raw datasets. A lot is expected from the "art" of deriving features from data using subject matter/domain expertise – which can be daunting given the velocity, variety and volume aspects of Big Data.

3.3 Science of Scalable Predictive Functions

Once data is structured, the next challenge is increasing $\boldsymbol{N}$, $\boldsymbol{d}$ and $\boldsymbol{k}$. The design of learning algorithms begin with the assumption that the data presented is a sample from a population. The algorithm is studied for statistical significance using popular tests and cross-validation methods. Typically, $\boldsymbol{N} \gg \boldsymbol{d}$ and $\boldsymbol{d} > \boldsymbol{k}$ and the rule of the thumb is that a statistically significant model can be built if $\boldsymbol{N} > \boldsymbol{2^d}$ for a $\boldsymbol{k}$ =2 problem. To date, there is no evidence

yet that increasing N translates to better accuracy, sensitivity, specificity, precision or recall of the learning algorithm. Increasing N only contributes to increased latency for the feature construction and evaluation phase.

Increasing d is a different problem that arises from automated data collection from a variety of sources while not knowing which ones will be relevant to the discovery of the interest. This problem also called as the curse of dimensionality causes the challenge of intractability of search through the d-dimensional space to accurately find a general predictive function that draws decision boundaries for the k classes. The machine learning literature has very few methods for situations where $d > N$.

Furthermore, there can be latent hierarchies and groupings within the N-d-k aspects of the data – as was the case with the manufacturing and image-triaging use-cases discussed earlier. The hierarchy in dimensionality of the data could arise from integrating newer data sources or subject matter experts. In some cases, k categories may be grouped as sub-categories. For example, samples may come from micro-segments of a population in the healthcare case. For the image triage use case, words in a dictionary could be grouped into concepts. In such situations, one model may not be enough to predict the interactions in the dataset and the model may have to be a function of a family predictive functions. Unfortunately, even parallel implementations of classical machine learning methods (Parallel R [11], Mahout [12], etc.) that parallelize the training of one predictive function do not scale to train a family of predictive functions.

Another emerging challenge is incremental learning (i.e., what happens when N=N+1, d = d+1 *or* k = k+1). Let us suppose we have a fraud detection algorithm trained on several samples and is deployed to detect and label suspicious activity from future streams. Over time, we either have more examples or we have newer data elements that are more predictive of suspicious activity. The idea of being able to update the predictive model (its structure and parameters) without having to retrain the model over the N samples has not received much research attention. In practice, we still have to derive learning algorithms that keep pace with the velocity of data collection for increasing N while d and k are fixed. The image-triaging use-case exposes another shortcoming with scaling classical machine learning algorithms along the k dimension. Prior work [3] has already shown that the accuracy, precision and recall of existing classification algorithms drops significantly when $k \sim 10{,}000$. We experienced a similar issue with our image-triaging use case. We learned that adding new categories for the same matrix M (without introducing a sample bias), proved to be a bigger challenge compared to increasing N or d. This is because learning a new category expects increased discriminatory power from a feature set that was sufficient for the previously trained k category case. In addition, to no guarantees on accuracy and possibility of reduced predictive accuracy, the learning algorithm still has to be retrained all over again.

4. CONCLUDING THOUGHTS

Knowledge discovery from Big Data is a greedy never-ending thirst. Big Data becoming bigger data and the vicious cycle of discoveries leading to better/smarter data collection is almost inevitable. Big Data products that enable discoveries by interrogation (staging and querying data), discoveries by association (integrating with newer data sources and finding weak associations and correlations) and discoveries by modeling (extracting novel features and training predictive functions) should be available off-the-shelf in the next few years. Machine learning research is making tremendous progress towards that goal. Jointly working with the database community the tools on shelf today that treat "analysis as a retrieval problem" solve most business and scientific application needs. Scalable appliances and software services of user-friendly software such as R, SAS and MATLAB are also emerging. However, we can do better in the Big Data era by designing and implementing machine learning algorithms with scale-friendly predictive functions that are both data-aware and infrastructure-aware.

Today, algorithms are designed for the von-Neumann architecture and forced into the "store-fetch-execute" paradigm [14]. The performance of the algorithms is dependent on the infrastructure (i.e., communications between processors, memory chips, and storage-disks). Some algorithms perform better in HPC architectures while some in cloud architectures. This is because in-core in-memory computations (that supports task-parallelism) are order(s) of magnitude faster than out-of-core in-disk computations (data-parallelism). Linear complexity algorithms and algorithms that are based on parameter estimation (e.g. regression analysis) scale pretty well on parallel databases while higher computational complexity algorithms (e.g. Eigen-value decomposition etc.) and iterative optimization problems (e.g. regularization) are better executed in HPC architectures. Benchmarks to understand algorithms and their dependencies to different scalable compute architectures (shared-memory, shared-storage, and shared-nothing, etc.) are currently underway at ORNL. The benchmarking exercise is a fundamental first step towards implementing algorithms that are infrastructure cognizant and also building future architectures that suit machine learning algorithms better and understanding the cost of performance at scale.

To address the science of data challenge and to better design predictive functions, we are exploring and evaluating the following methods: (i) deep learning algorithms [6] that automate the feature engineering process by learning to create and sift through data-driven features, (ii) incremental learning algorithms in associative memory architectures [13] that can seamlessly adapt to future data samples and sources, (iii) faceted learning that can learn hierarchical structure in the data, and (iv) multi-task learning that can learn several predictive functions in parallel. Some of these techniques (in theory) address the curse of dimensionality while leveraging its blessings [15]. In conclusion, the key to successful demonstration of scalable machine learning in future will rely on choosing the optimal hardware based on guiding benchmarks, designing algorithms that are mature in understanding the science of data, and innovative design of predictive functions that can optimally leverage task-parallelism and data-parallelism to handle the expectations with increasing N, d, and k.

In the future, industry practitioners will have to start evolving enterprise data analytics to an agile and flexible workflow-lifecycle management approach – moving away from the project management approach followed with business-intelligence projects today. As data sizes grow and algorithms scale – integrated datasets will be expected to answer more than one question. We will soon need smart learning algorithms that assist humans in formulating questions for the Big Data. We will need

analytical algorithms that are able to discover insights from data even if there is more noise than signal. We will need reasoning algorithms that can automatically discover "newer" insights when new data sources are posed in context with a constantly evolving and increasing body of knowledge in order to keep up with the thirst for innovation. The marriage of scalable relational databases with graph data structures can enable the "knowledge-nurturing" next generation analytics.

6. REFERENCES

[1] Tansley, Stewart, and Kristin Michele Tolle, eds. "The fourth paradigm: data-intensive scientific discovery." (2009).

[2] Asuncion, Arthur, and David Newman. "UCI machine learning repository." (2007).

[3] Deng, Jia, et al. "What does classifying more than 10,000 image categories tell us?." Computer Vision–ECCV 2010. Springer Berlin Heidelberg, 2010. 71-84..

[4] http://data-informed.com/why-more-data-and-simple-algorithms-beat-complex-analytics-models/

[5] Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern classification. John Wiley & Sons, 1999.

[6] Bengio, Y."Learning deep architectures for AI." Foundations and trends in Machine Learning 2.1 (2009): 1-127.

[7] Ando, Rie Kubota, and Tong Zhang. "A framework for learning predictive structures from multiple tasks and unlabeled data." The Journal of Machine Learning Research 6 (2005): 1817-1853.

[8] http://www.wired.com/2012/06/google-x-neural-network

[9] Coates, Adam, et al. "Deep learning with cots hpc systems." Proceedings of the 30th International Conference on Machine Learning. 2013.

[10] Lim, Seung-Hwan, et al. "Graph Processing Platforms at Scale: Practices and Experiences." Proceedings of the 2015 IEEE International Symposium on Performance Analysis of Systems and Software. 2015.

[11] Yoginath, Srikanth B., et al. "RScaLAPACK: High-Performance Parallel Statistical Computing with R and ScaLAPACK." ISCA PDCS. 2005.

[12] Anil, Robin, Ted Dunning, and Ellen Friedman. Mahout in action. Manning, 2011.

[13] Duncan, Ralph. "A survey of parallel computer architectures." Computer 23.2 (1990): 5-16.

[14] Asanovic, Krste, et al. The landscape of parallel computing research: A view from Berkeley. Vol. 2. Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, 2006.

[15] Donoho, David L. "High-dimensional data analysis: The curses and blessings of dimensionality." AMS Math Challenges Lecture (2000): 1-32.

[16] Chu, Cheng, et al. "Map-reduce for machine learning on multicore." Advances in neural information processing systems 19 (2007): 281.

[17] Hellerstein, Joseph M., et al. "The MADlib analytics library: or MAD skills, the SQL." Proceedings of the VLDB Endowment 5.12 (2012): 1700-1711.

[18] Chandola, Varun, Sreenivas R. Sukumar, and Jack C. Schryver. "Knowledge discovery from massive healthcare claims data." In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, (2013) pp. 1312-1320.

[19] Begoli, Edmon, and James Horey. "Design principles for effective knowledge discovery from big data." In Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on, pp. 215-218. IEEE, 2012.

[20] Sukumar, S. R., Olama, M. M., McNair, A. W., & Nutaro, J. J. (2013, May). Concept of operations for knowledge discovery from Big Data across enterprise data warehouses. In *SPIE Defense, Security, and Sensing* – Next Generation Analyst (pp. 875805-875805).

[21] V. Chandola, J. Schryver, and S.R. Sukumar, "Fraud Detection in Healthcare", in Health Data Analytics, (eds.) Charu C. Agarwal, and Chandan Reddy, Chapman and Hall/CRC 2015, Print ISBN: 978-1-4822-3211-0, eBook ISBN: 978-1-4822-3212-7, 2015.

[22] Park, Yubin, et al. "Graph databases for large-scale healthcare systems: A framework for efficient data management and data services." *Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on*. IEEE, 2014.

[23] Sukumar, Sreenivas R., and Mallikarjun Shankar. "Value-of-information based middleware for autonomic querying of distributed sensor databases." *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2010.

[24] Ando, Rie Kubota, and Tong Zhang. "A framework for learning predictive structures from multiple tasks and unlabeled data." The Journal of Machine Learning Research 6 (2005): 1817-1853.

[25] Symons, Christopher T., and Justin M. Beaver. "Nonparametric semi-supervised learning for network intrusion detection: combining performance improvements with realistic in-situ training." Proceedings of the 5th ACM workshop on Security and artificial intelligence. ACM, 2012.

[26] Karau, Holden, et al. *Learning Spark: Lightning-Fast Big Data Analysis*. " O'Reilly Media, Inc.", 2015.

[27] Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.

[28] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database."*Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[29] Liu, Hugo, and Push Singh. "ConceptNet—a practical commonsense reasoning tool-kit." BT technology journal 22.4 (2004): 211-226.

[30] Russell, Bryan C., et al. "LabelMe: a database and web-based tool for image annotation." International journal of computer vision 77.1-3 (2008): 157-173.

[31] Huiskes, Mark J., and Michael S. Lew. "The MIR Flickr retrieval evaluation." Proceedings of the 1st ACM international conference on Multimedia information retrieval. ACM, 2008.