Big Data Analytics for Network Traffic Monitoring and Analysis

Francesca Soro
Politecnico di Torino
AIT Austrian Institute of Technology

Supervisors
Dr. Pedro Casas
Dr. Alessandro D'Alconzo
AIT Austrian Institute of Technology

***Abstract*—The complexity of the Internet has dramatically increased in the last few years, making it more important and challenging to design scalable Network Traffic Monitoring and Analysis (NTMA) applications and tools. Critical NTMA applications such as the detection of anomalies, network attacks and intrusions, require fast mechanisms for online analysis of thousands of events per second, as well as efficient techniques for offline analysis of massive historical data. We are witnessing a major development in Big Data Analysis Frameworks (BDAFs), but the application of BDAFs and scalable analysis techniques to the NTMA domain remains poorly understood and only in-house and difficult to benchmark solutions are conceived. This thesis tackles this growing need by developing novel scalable techniques capable to analyze both online network traffic data streams and offline massive traffic datasets. We explore scalable online and offline data mining and machine learning-based techniques to monitor and characterize extremely large network traffic datasets. The work is conducted on big-data analysis technologies using off-the-shelf big data frameworks. The developed techniques are tested on top of different real network traffic measurement datasets.**

***Index Terms*—Big-Data Analytics; Machine Learning; Figh-Dimensional Data; Network Security; Anomaly Detection;**

I. Introduction & Problem Description

Network Traffic Monitoring and Analysis (NTMA) has taken a paramount role to understand the functioning of the Internet, especially to get a broader and clearer visibility of unexpected events. One of the major challenges faced by large-scale NTMA applications is the processing and analysis of large amounts of heterogeneous and fast network monitoring data. Network monitoring data usually comes in the form of high-speed streams, which need to be rapidly and continuously processed and analyzed.

A variety of methodologies and tools have been devised to passively monitor network traffic, extracting large amounts of data from live networks. What is needed is a flexible data processing system able to analyze and extract useful insights from such rich and heterogeneous sources of data, offering the possibility to apply complex Machine Learning (ML) and Data Mining (DM) techniques. The introduction of Big Data processing led to a new era in the design and development of large-scale data processing systems. This new breed of tools and platforms are mostly dissimilar, have different requirements, and are conceived to be used in specific situations for specific needs. Each Big Data practitioner is forced to muddle through the wide range of options available, and NTMA is not an exception. A similar problem arises in the case of Big Data analytics through ML and DM based techniques. Despite the existence of ML libraries for Big Data Analysis Frameworks (BDAFs), there is a big gap to the application of such techniques for NTMA when considering fast online streams and massive offline datasets.

A. Objectives

One of the main questions that the Big-DAMA team posses itself as researchers in the NTMA domain is straightforward: if one wants to tackle NTMA applications with (near) real-time requirements in current massive traffic scenario, which would be the best system one should use to the task? In addition, if the main target is to perform complex data analytics on top of this massive traffic, considering both supervised and unsupervised ML approaches, how should it be done? Which are the best ML algorithms for doing so? The Big-DAMA project will accelerate NTMA practitioners' and researchers' understanding of the many new tools and techniques that have emerged for Big Data Analytics in recent years. Big-DAMA will particularly identify and test the most suitable BDAFs and available Big Data Analytics implementations of ML and DM algorithms for tackling the problems of Anomaly Detection and Network Security in an increasingly complex network scenario. The Big-DAMA project proposes three main objectives:

- Explore, conceive and test scalable online and offline Machine Learning and Data Mining-based techniques to monitor and characterize extremely fast and/or extremely large network traffic datasets.
- Conceive novel frameworks for Big Data Analytics tailored to Anomaly Detection and Network Security, evaluating and selecting the best BDAFs matching NTMA needs. Such frameworks would target traffic stream data processing (online processing) and massive offline data processing (offline processing).
- Conceive a novel benchmark for BDAFs and Big Data Analytics tailored for NTMA applications, particularly focusing on stream analysis algorithms and online processing tasks.

The starting point of Big-DAMA is DBStream [32], a Data Stream Warehouse we have recently developed between FTW and the University of Waterloo, and benchmarked against new big data analysis platforms such as Spark, in collaboration with Politecnico di Torino, showing very promising results in the field of NTMA [32]. DBStream has been running at FTW on a core ISP network since two years now, and we have a large accumulated experience to take the next step.

B. Specific Research questions

While the proposed objectives of the Big-DAMA project might look a-priori more practical than theoretical, we will address several fascinating topics related to the application of ML and DM techniques in the Big Data domain, for the specific purposes of network traffic exploration and extraction of information. So besides the aforementioned objectives, the Big-DAMA project will provide answers to the following research questions:

- How to automatically construct proper data representations (i.e., computing good features or descriptors) given a certain ML algorithm and a huge dataset of unlabeled data?
- How to perform unsupervised feature selection?
- How to cluster big and fast evolving data streams? This is still an open problem, and result would be highly useful when thinking on unsupervised NTMA.
- Which supervised learning approaches can be applied in an online manner with big amounts of streaming data?
- Which are the statistical implications of divide and conquer algorithms when dealing with Big Data?
- Which are the impacts of traffic sampling and aggregation in the results of Big Data Analytics for NTMA?
- Is Big Data a curse when dealing with Anomaly Detection and Network Security, or it can be useful to improve traditional approaches?

II. State of the Art & Contributions

The Big-DAMA project deals with Big-Data Analytics and Network Traffic Monitoring and Analysis applications for Anomaly Detection and Network Security. In this context, we structure the state of the art analysis in five different sections: (i) Big Data Analysis Frameworks, (ii) Machine Learning, Data Mining and Big Data Analytics, (iii) Network Anomaly Detection and Security, (iv) Application of Big Data Analysis Frameworks for Traffic Monitoring and Analysis, and finally (v) Benchmarks for Big Data Analysis Solutions. For the sake of brevity we do not reference all the systems, solutions and algorithms we discuss next, but we mention them to give the reader an idea of the range of the problem.

A. Big Data Analysis Frameworks

The introduction of Big Data processing led to a new era in the design and development of large-scale data processing systems [33]. This new breed of tools and platforms are mostly dissimilar, have different requirements, and are conceived to be used in specific situations for specific needs. Each Big Data practitioner is forced to muddle through the wide range of options available, and NTMA is not an exception. A basic yet complete taxonomy of Big Data Analysis Frameworks includes traditional Database Management Systems (DBMS) and extended Data Stream Management Systems (DSMSs), noSQL systems (e.g., all the MapReduce-based systems), and Graph-oriented systems. While the majority of these systems target the offline analysis of static data, some proposal consider the problem of analyzing data coming in the form of online streams. DSMSs such as Gigascope [34] and Borealis [35] support continuous online processing, but they cannot run offline analytics over static data. The Data Stream Warehousing (DSW) paradigm provides the means to handle both types of online and offline processing requirements within a single system. DataCell and DataDepot are examples of this paradigm [36]. NoSQL systems such as e.g. MapReduce [40] have also rapidly evolved, supporting the analysis of unstructured data. Apache Hadoop [41] and Spark [42] are very popular implementations of MapReduce systems. These are based on offline processing rather than stream processing. There has been some promising recent work on enabling real-time analytics in NoSQL systems, such as Spark Streaming [43], Indoop [37], Muppet [38] and SCALLA [39], but these remains unexploited in the NTMA domain. The offer of solutions available is overwhelming; more examples include Storm, Samza, Flink (NoSQL); Hawq, Hive, Greenplum (SQL-oriented); Giraph, GraphLab, Pregel (graph-oriented), as well as well known DBMSs commercial solutions such as Teradata, Dataupia, Vertica and Oracle Exadata (just to name a few of them).

B. Machine Learning, Data Mining and Big Data Analytics

The fields of Machine Learning (ML) and Data Mining (DM) have been studied for more than 50 years, and today there is a comprehensive list of options [46]. Popular supervised ML algorithms include Locally Weighted Linear Regression, Naive Bayes, Gaussian Discriminative Analysis, Logistic Regression, Neural Networks, Principle Components Analysis, Independent Component Analysis, Expectation Maximization, Support Vector Machine, Decision Trees, and many more. Clustering algorithms (unsupervised analysis) [58] include partition-based clustering (K-means), density-based clustering (DBSCAN), hierarchical-clustering, spectral clustering [60], distribution-based clustering, etc.. Two particularly promising unsupervised and supervised algorithms which exemplify the NTMA needs in terms of complex traffic analysis are Sub-Space Clustering [59] and Adaptive Trees [61], [62] respectively. The former because of its capabilities for exploring big dimensional data, even when working with Big Data, the latter because of its direct applications in stream, supervised-based analysis. In the same context, work such as [63] shows potential and yet unexplored directions to perform clustering with stream data, which is also highly appealing for NTMA applications, such as those in our work on autonomous network security [44]. We can find today an increasing number of MapReduce-based implementations of the most important ML algorithms. Indeed, today there is a reasonably high number of ML libraries available, including Apache Mahout [74] and MLlib [75] (NoSQL), MADlib (SQL-based), as well as frameworks implementing machine learning and data mining algorithms (e.g., Weka, MOA, SAMOA, etc.). To describe some of them, Mahout is a scalable machine learning library with a relatively long history, containing implementations of algorithms in the areas such as classification, clustering, recommendation systems, etc., whereas MADlib provides machine learning in SQL, including classification, regression, clustering, association rule mining, descriptive statistics, etc.

C. Network Anomaly Detection and Security

Anomaly detection provides the basis to detect novel incidents such as network failures, misconfigurations or network attacks that cannot be discovered by signature-based approaches. While many approaches exist today for applying a wide variety of different machine learning techniques to network intrusion detection [44], [49]–[53], we are still not able to cope with todays attack techniques. There are many reasons for this. While machine learning performs extremely well in other fields (such as SPAM detection), network traffic introduces much larger challenges to the detection task [54]. Reasons are the high dynamics of network traffic, the high costs of misclassifications and active attackers that adapt to detection techniques. The specific nature of network traffic confines the applicability of machine learning techniques. It requires very well focused problem statements and carefully tuned learning approaches. Also, while existing approaches make use of the whole range of available machine learning techniques, the extremely important steps of feature generation and feature selection is underrepresented in current literature [55]. Feature generation and feature selection are essential to provide the input and significantly determine the detection performance [56]. Classical machine learning approaches are often computationally expensive. Data rates and the overall amount of network traffic is permanently increasing. In addition, network traffic analysis demands more and more detailed analysis results and online detection of anomalies requires high response times, i.e. fast processing and analysis, in order to react to severe situations and reduce the damage during critical incidents. With these characteristics, network traffic data analysis is a perfect candidate to profit from the benefits of big data analysis techniques [57], [64].

D. Application of Big Data Analysis Frameworks for Traffic Monitoring and Analysis

The application of Big Data Analysis Frameworks for NTMA tasks requires certain system capabilities: i) scalability: the framework must offer, possibly inexpensively, storage and processing capabilities to scale with huge amounts of data generated by in-network traffic monitors and collectors. ii) Real-time processing: the system must be able to ingest and process data in real-time fashion. iii) Historical data processing: the system must enable the analysis of historical data. iv) Traffic data analysis tools: embedding libraries or plugins specifically tailored to analyze traffic data. In the following we present the main categories in which currently available data analysis technologies can be classified. For each of them, we highlight pros and cons, and explain why none of them fits for NTMA. Traditional SQL-like databases are inadequate for the continuous real-time analysis of data. As we mentioned before, Data Stream Warehouses have been introduced to extend traditional database systems with continuous data ingest and processing. These technologies leverage arbitrary SQL framework to perform rolling data analysis, i.e., they periodically import and process batches of data arriving at the system. In some cases, these technologies have been proven to be able to outperform in terms of processing speed Big Data technologies [32]. However, differently from Big Data technologies, they can not scale with the huge amounts of traffic data generated by nowadays networks. This represents a severe limitation to their applicability to NTMA purposes. Big Data Analysis Frameworks based on the MapReduce paradigm have been recently started to be adopted for NTMA applications [65]. Considering the specific context of network monitoring, some solutions to adapt Hadoop to process traffic data have been proposed [66]. However, the main drawback of Big Data technologies in general is their inherent offline processing, which is not suitable for real-time traffic analysis, highly relevant in NTMA tasks. The only approach that leverages Hadoop for rolling traffic analysis is described in [67]. As we mentioned before, there have also been some Big Data Analysis Frameworks for online data processing, but none of these has been applied to the NTMA domain.

E. Benchmarks for Big Data Analysis Solutions

Benchmarking Big Data Analysis Frameworks (BDAFs) is not a trivial task, and at the end of the day, it has a paramount role on the Big Data Analytics domain. How to know if a certain BDAF is fast and accurate enough to tackle the specific needs of NTMA applications, and how to select the best BDAFs to perform complex analytics on the monitoring traffic? The literature offers many specifically tailored solutions: Gridmix [72], the Hive Benchmark [71], HiBench [69] and the Berkeley Big Data Benchmark [45], to name a few of them. The latter is based on a comparative study between MapReduce and parallel DBMSs frameworks conducted in [68], and consists of a carefully designed query workload, aiming to test the capability of data processing systems for online analytical processing queries. Other BDAFs (Spark, SimSQL, GraphLab and Giraph) have been benchmarked in terms of completion-time when running complex, hierarchical Machine Learning models on very large datasets [47]. To the best of our knowledge, none of the available benchmarks addresses specifically NTMA applications. Another important aspect when working with Internet measurements is the need to integrate raw data with other data sources (e.g., for geolocation, routing and topology, classification, etc.), which is not addressed by those benchmarks.

F. Scientific Challenge

The extensive presented state of the art presents several limitations related to the application of Big Data Analytics to NTMA applications. Firstly, Big Data Analytics results on NTMA applications are seldom available, specially when considering online, stream based traffic analysis. This creates a major gap between the developments of Big Data Analytics and Analysis Frameworks and the development of NTMA systems capable of analyzing huge amounts of network traffic. In addition, while there is a vast number of Big Data Frameworks, the offer is so big and difficult to track that makes it very challenging to determine which one to choose for the purpose of NTMA. Secondly, considering the theory of Big Data Analytics applied to the NTMS domain, most of the proposed Machine Learning frameworks and libraries do not scale well in fast big data scenarios, as their main target is offline data analytics. In addition, while some supervised and unsupervised learning algorithms are already available for Big Data Analytics, we are at a very early stage development and there is big room for improvement. The most notable example is explorative data analysis through clustering. Available algorithms are either too simple (e.g., no techniques such as Sub-Space clustering are available, most of the work done on traditional k-means), or too tailored to specific domains not related to traffic analysis. Clustering data streams is still an open problem, and a very useful one for unsupervised Anomaly Detection and Network Security. Similar unsolved problems such as unsupervised feature selection become more challenging as well, due to scalability issues in the Big Data scenario. Also when considering supervised approaches, we do not have today much evidence on how supervised online learning approaches perform with big stream-based traffic. There are also limitations in the analysis and comparison of different machine learning and data mining techniques running in Big Data Frameworks, because available benchmarks are very ad-hoc and tailored to specific types of systems (e.g., tailored for MapReduce-like frameworks). The Big-DAMA project will advance many of this open issues, as we discussed in the next Section.

III. The Big-DAMA Project

IV. Big-Data Frameworks

V. Big-Data Analytics for Off-line Analysis

VI. Big-Data Analytics for On-line Analysis

VII. Use Cases: Network Security & Anomaly Detection

VIII. Discussion

IX. Concluding Remarks

References

[1] M. Van der Laan, et al., “Super learner”, in Statistical applications in genetics and molecular biology, vol. 6, no. 1, 2007.

[2] P. Casas, et al., “Big-DAMA: Big Data Analytics for Network Traffic Monitoring and Analysis”, in *ACM SIGCOMM LANCOMM Workshop*, 2016.

[3] P. Casas, et al., “POSTER:(Semi)-Supervised Machine Learning Approaches for Network Security in High-Dimensional Network Data”, in *ACM CCS*, 2016.

[4] P. Casas, et al., “Machine-learning based approaches for anomaly detection and classification in cellular networks”, in *TMA*, 2016.

[5] Y. Freund, et al., “Using and combining predictors that specialize”, in *ACM STOC*, 1997.

[6] J. Hansen, “Combining predictors: Some old methods and a new method”, available online at Citeseer, 1998.

[7] T. Dietterich, “Ensemble learning”, The handbook of brain theory and neural networks, vol. 2, pp. 110–125, MIT Press: Cambridge, MA, 2002.

[8] P. Sollich, A. Krogh, “Learning with ensembles: How overfitting can be useful”, Advances in neural information processing systems, pp. 190–196, MORGAN KAUFMANN PUBLISHERS, 1996.

[9] R. Fontugne, et al., “MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking”, in *ACM CoNEXT*, 2010

[10] T. Nguyen, G. Armitage, “A survey of techniques for Internet Traffic Classification using Machine Learning”, in IEEE Comm. Surv. & Tut., vol. 10, no. 4, pp. 56–76, 2008.

[11] A. Ghosh, A. Schwartzbard, “A Study in Using Neural Networks for Anomaly and Misuse Detection”, in *USENIX Security Symposium*, 1999.

[12] A. Mitrokotsa et al., “Detecting denial of service attacks using emergent self-organizing maps”, in *IEEE ISSPIT*, 2005.

[13] M. Ostaszewski et al., “A non-self space approach to network anomaly detection”, in *IEEE IPDPS*, 2006.

[14] W. Chimphlee, et al., “Integrating genetic algorithms and fuzzy C-means for anomaly detection”, in *IEEE Indicon*, 2005.

[15] G.Prashanth, et al., “Using random forests for network-based anomaly detection”, in *IEEE ICSCN*, 2008.

[16] Y. Li et al., “An efficient network anomaly detection scheme based on TCM-KNN algorithm and data reduction mechanism”, in *IAW*, 2007.

[17] P. Casas et al., “Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge”, in *Computer Communications*, vol. 35 (7), pp. 772-783, 2011.

[18] T. Ahmed, et al., “Machine Learning Approaches to Network Anomaly Detection”, in *USENIX SYSML Workshop*, 2007.

[19] V. Chandola, et al., “Anomaly detection: A survey”, ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, 2009.

[20] M. Ahmed, et al., “A Survey of Network Anomaly Detection Techniques”, J. Netw. Comput. Appl., vol. 60, pp. 19–31, 2016.

[21] W. Zhang, et al., “A Survey of Anomaly Detection Methods in Networks”, in *CNMT Symposium*, 2009.

[22] A. Moore et al., “Internet Traffic Classification using Bayesian Analysis Techniques”, in *Proc. ACM SIGMETICS*, 2005.

[23] M. Roughan et al., “Class-of-Service Mapping for QoS: a Statistical Signature-Based Approach to IP Traffic Classification”, in *IMW*, 2004.

[24] N. Williams el al., “A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification”, in *ACM CCR*, vol. 36 (5), pp. 5-16, 2006.

[25] S. Valenti et al., “Accurate, Fine-Grained Classification of P2P-TV Applications by Simply Counting Packets”, in *TMA*, 2009.

[26] J. Erman et al., “Traffic Classification using Clustering Algorithms”, in *MineNet*, 2006.

[27] P. Casas et al., “MINETRAC: Mining Flows for Unsupervised Analysis & Semi-Supervised Classification”, in *ITC*, 2011.

[28] J. Erman et al., “Semi-Supervised Network Traffic Classification”, in *ACM SIGMETRICS*, 2007.

[29] T. Nguyen et al., “A Survey of Techniques for Internet Traffic Classification using Machine Learning”, in *IEEE Comm, Surv. & Tut.*, vol. 10 (4), pp. 56-76, 2008.

[30] R. Ravinder, et al., “Real Time Anomaly Detection Using Ensembles”, in *ICISA International Conference*, 2014.

[31] M. Ozdemir, I. Sogukpinar, “An Android Malware Detection Architecture based on Ensemble Learning”, in Transactions on Machine Learning and Artificial Intelligence, vol. 2, no. 3, pp. 90–106, 2014.

[32] A. Baer, A. Finamore, P. Casas, L. Golab, M. Mellia, “Large-Scale Network Traffic Monitoring with DBStream, a System for Rolling Big Data Analysis,” in *Proc. IEEE International Conference on Big Data*, 2014.

[33] M. Stonebraker, “SQL Databases vs. noSQL Databases,” in *Communications of the ACM*, vol. 53(4), pp. 10-11, 2010.

[34] C. Cranor, T. Johnson, O. Spataschek, V. Shkapenyuk, “Gigascope: A Stream Database for Network Applications,” in *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 647-651, 2003.

[35] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, S. Zdonik, “Aurora: A New Model and Architecture for Data Stream Management,” in *The VLDB Journal*, vol. 12(2), pp. 1020-1039, 2003.

[36] L. Golab, T. Johnson, J. Seidel, V. Shkapenyuk, “Stream Warehousing with DataDepot,” in *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 847-854, 2009.

[37] P. Bhatotia et al., "Indoop: Mapreduce for Incremental Computations," in *Proc. of the ACM Symposium on Cloud Computing*, pp. 7-14, 2011.

[38] , W. Lam, L. Liu, S. Prasad, A. Rajaraman, Z. Vacheri, A. Doan, "Muppet: Mapreduce-style processing of fast data," in *Proc. VLDB Endow.*, vol. 5(12), pp.1814-1825, 2012.

[39] B. Li, E. Mazur, Y. Diao, A. McGregor, P. Shenoy, "Scalla: A platform for scalable one-pass analytics using mapreduce," in *ACM Trans. Database Syst.* 37(4), pp. 27-43, 2012.

[40] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in *Communications of the ACM*, vol. 51(1), pp. 107-113, 2008.

[41] T. White, "Hadoop: the Definitive Guide," *O'Reilly Media, Inc.*, ISBN:0596521979 9780596521974, 2009.

[42] M. Zaharia, M. Chowdhury, M. Franklin, S. Shenker, I. Stoica, "Spark: Cluster Computing with Working Sets," in *Proc. of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, pp. 10-16, 2010.

[43] M. Zaharia, T. Das, H. Li, S. Shenker, I. Stoica, "Discretized Streams: An Efficient and Fault-tolerant Model for Stream Processing on Large Clusters," in *Proc. of the 4th USENIX Conference on Hot Topics in Cloud Computing*, pp. 10-16, 2012.

[44] P. Casas, J. Mazel, P. Owezarski, "Knowledge-Independent Traffic Monitoring: Unsupervised Detection of Network Attacks," in *IEEE Network Magazine*, vol. 26(1), pp. 13-21, 2012.

[45] Berkeley AMPLab. Big Data Benchmark. https://amplab.cs.berkeley.edu/benchmark/, 2014.

[46] C. Bishop, "Pattern Recognition and Machine Learning", 2007.

[47] Z. Cai, Z. Gao, S. Luo, L. Perez, Z. Vagena, C. Jermaine, "A Comparison of Platforms for Implementing and Running Very Large Scale Machine Learning Algorithms," in *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 1371-1382, 2014.

[48] F. Huici, A. di Pietro, B. Trammell, J. Gomez Hidalgo, D. Martinez Ruiz, N. d'Heureuse, "Blockmon: A High-Performance Composable Network Traffic Measurement System," in *Proc. of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp. 79-80, 2012.

[49] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, B. Stiller, "An Overview of IP Flow-Based Intrusion Detection," in *IEEE Communications Surveys Tutorials*, vol. 12(3), pp. 343-356, 2010.

[50] M. Panda, A. Abraham, S. Das, M.R. Patra, "Network Intrusion Detection Systems: A Machine Learning Approach," in *Int. Decision Technologies*, vol. 5 (4), pp. 347-356, 2011.

[51] I. Syarif, A. Prugel-Bennett, G. Wills, "Data Mining Approaches for Network Intrusion Detection: from Dimensionality Reduction to Misuse and Anomaly Detection," in *Journal of Information Technology Review*, vol. 3(2), pp. 70-83, 2012.

[52] L. Khan, M. Awad, B. Thuraisingham, "A New Intrusion Detection System Using Support Vector Machines and Hierarchical Clustering," in *The VLDB Journal*, vol. 16(4), pp. 507-521, 2007.

[53] A. Marnerides, A. Schaeffer-Filho, A. Mauthe, "Traffic Anomaly Diagnosis in Internet Backbone Networks: A Survey," in *Computer Networks*, vol. 73, pp. 224-243, 2014.

[54] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proc. IEEE Symposium on Security and Privacy*, pp. 305-316, 2010.

[55] M. Tavallaee, N. Stakhanova, A.A. Ghorbani, "Toward Credible Evaluation of Anomaly-Based Intrusion-Detection Methods," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40(5), pp. 516-524, 2010.

[56] F. Iglesias and T. Zseby, "Analysis of Network Traffic Features for Anomaly Detection," in *Machine Learning*, pp. 1-26, 2014.

[57] S. Sagiroglu and D. Sinanc, "Big Data: A Review," in *Proc. Int. Conf. on Collaboration Technologies and Systems*, pp. 42-47, 2013.

[58] H. Kriegel, P. Kroger, A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," in *ACM Transactions on Knowledge Discovery from Data*, 2009.

[59] L. Parsons, E. Haque, H. Liu, "Subspace Clustering for High Dimensional data: a Review,", in *SIGKDD Explor. Newsl.*, 2004.

[60] W. Chen et al., "Parallel Spectral Clustering in Distributed Systems Pattern Analysis and Machine Intelligence," in *IEEE Transactions*, vol 33(3), pp. 568-586, 2011.

[61] P. Domingos et al., "Mining High-Speed Data Streams," in *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 71-80, 2000.

[62] P. Biswanath, J. Herbach , S. Basu , R. Bayardo, "PLANET: Massively Parallel Learning of Tree Ensembles with MapReduce", in *Proc. VLDB Endow.*, pp. 1426-1437, 2009.

[63] C. Aggarwal et al., "A Framework for Clustering Evolving Data Streams," in *Proc. of the International Conference on Very Large Data Bases*, pp. 81-92, 2003.

[64] P. Costa, A. Donnelly, A. Rowstron, G. O'Shea, "Camdoop: Exploiting In-network Aggregation for Big Data Applications," in *Proc. of the 9th USENIX Symposium on Networked Systems Design and Implementation*, pp. 29-42, 2012.

[65] R. Fontugne, J. Mazel, K. Fukuda, "Hashdoop: A MapReduce Framework for Network Anomaly Detection," in *Proc. of the IEEE Conference on Computer Communications Workshops*, pp. 494-499, 2014.

[66] Y. Lee et al., "Toward scalable internet traffic measurement and analysis with Hadoop," in SIGCOMM Comput. Commun. Rev., 43(1), pp. 5-13, 2012.

[67] J. Liu, F. Liu, N. Ansari, "Monitoring and analyzing big traffic data of a large-scale cellular network with Hadoop," in IEEE Network, 28(4), pp. 32-39, 2014.

[68] A. Pavlo, E. Paulson, A. Rasin, D. Abadi, D. Dewitt, S. Madden, M. Stonebraker, "A Comparison of Approaches to Large-Scale Data Analysis," in *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 165-178, 2009.

[69] S. Huang, J. Huang, J. Dai, T. Xie, B. Huang, "The HiBench Benchmark Suite: Characterization of the MapReduce-based Data Analysis," in *Proc. of the IEEE SIGMOD International Conference on Data Engineering Workshops*, pp. 41-51, 2010.

[70] Y. Chen, A. Ganapathi, R. Griffith, R. Katz, "The Case for Evaluating MapReduce Performance Using Workload Suites," in *Proc. of the IEEE Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, pp. 25-27, 2011.

[71] Y. Jia, "A Benchmark for Hive, PIG and Hadoop," available at https://issues.apache.org/jira/browse/hive-396

[72] C. Douglas, H. Tang, "Gridmix3 – Emulating Production Workload for Apache Hadoop," available at https://developer.yahoo.com/blogs/hadoop/gridmix3-emulating-production-workload-apache-hadoop-450.html

[73] H. Karloff, S. Suri, S. Vassilvitskii, "A Model of Computation for MapReduce," in *Proc. ACM SODA*, 2010.

[74] Apache Mahout, http://mahout.apache.org.

[75] MLlib: https://spark.apache.org/mllib/