Processing Sampled Big Data

Waleed Albattah, Rehan Ullah Khan
Information Technology Department
Qassim University,
Qassim, KSA

***Abstract*—Big data processing requires extremely powerful and large computing setup. This puts bottleneck not only on processing infrastructure but also many researchers don't get the freedom to analyze large datasets. This paper thus analyzes the processing of the large amount of data from machine learnt models that are built on the smaller sets of data samples. This work analyzes more than 40 GB data by testing different strategies of reducing the processed data without losing and compromising on the detection and model learning in machine learning. Many alternatives are analyzed and it is observed that 50% reduction does not drastically harm the machine learning model performance. On average, in SVM only 3.6%, and in Random Forest, only 1.8% performance is reduced, if only 50% data is used. The 50% reduction in instances means that in most cases, the data will fit in the RAM and the processing times will be considerably reduced, benefitting in execution times and or resources. From the incremental training and testing experiments, it is found that in special cases, smaller sub-sampled data can be used for model generation in machine learning problems. This is useful in cases, where there are either limitations on hardware or one has to select among many available machine learning algorithms.**

Keywords—Deep learning; content analysis; machine learning; support vector machines; random forest

I. Introduction

The immense increase in technology sophistication these days and the relevant exponential increase of data being circulated and produced has resulted into the fact that ordinary data has been turned into big data. As explained by the name, big data refers to the data type that is massive in its size, formats that it holds, and requires high velocity servers for fetching it at required time [1]. Big data constitutes of variability, volume and velocity of data that needs to be accessed. This data is usually stored in large servers and is accessed only when required [2]. This big data is then used for carrying out ordinary operations of organization like decision making, sorting and other business related tasks [3]. However, for increasing efficiency and accuracy, a tradeoff between efficiency and size of application is crucial [4]. Common example of this is global positioning system, facial recognition cameras and connected automated vehicles. The efficiency of these applications can be enhanced through the provision of increased data sets for model learning. On the other hand, this is not feasible as large data sets require high storage space which in turn, becomes hard to be processed. For this purpose, it is required that a mechanism is built that allows sub sets of big data to hold similar knowledge and information as that of original data [5].

Big data has posed some serious risks to the data computation as well which needs to be addressed in order to ensure that the end user is protected in the end. For this purpose, usually some parameters are defined which ensure big data quality and information quality [6]. These parameters include, Syntactical Validity, Appropriate Identity association, appropriate attribute association, accuracy, precision, temporal applicability, theoretical relevancy, practical relevancy, currency, completeness, controls and audibility [6]. Other than this, management of servers and data for access control, privileges, sortation and security create other issues [7]. By 2002, digital devices were more than 92% with 5 Exabyte of data [8]. This number has been increasing since then and the problem has been evolving gradually. Today, big data is about $46.4 billion industry [8], meaning that, despite the problems of data handling, interest of users is growing over years. When it comes to data mining, this task becomes extremely complex when there exist hundreds of groups that are classified on the basis of minor differences, increasing work load and compilation time [8].

Apart from its never ending applications, big data is becoming a challenging concept for data mining, machine learning, information fusion, computational intelligence, social networks and the semantic web, etc. [9]. In this regards, issues of data processing, data use for pattern mining, data storage, user behavior analysis, data visualization and data tracking have attracted considerable attention [10].

This havoc of solution search for big assembly issues has been increased due to the fact that technologies like machine learning, computational intelligence and social networks are using libraries for data processing. These libraries are in turn increasing in size as the application scope is increasing. Due to which, solutions for simplicity of big data handling are continually researched and examined. These solutions include, data sampling, data condensation, density based approaches, incremental learning, divide and conquer, grid based approaches, distributed computing and others [8].

From processing perspectives, the Big data sampling has the biggest issue of complexity, computational burden and inefficiency to complete the task properly [11]. Sampling effort is the number of data sets that can be added per sample. It is assumed generally that sampling effort data sets richness is weak only if sampling bias is done successfully through estimation [12]. Selection bias, in this regards, can be computed and determined successfully though inverse sampling procedure, in which information from external resources is used, or by digital integration technique, in which big data is combined with independent probability sample

[13]. Size of a sample is extremely critical and plays an important role in determining accuracy of the system [14]. For this purpose, as a solution to big data sampling issues, many algorithms have been presented like Zig Zag process [15], non-probability sampling [13], inverse sampling, cluster based sampling [16].

Machine learning is a part of data analytics that learns from the available data to predict, decide and take insights [16]. Based upon statistics, it extracts trends from data and then computes it for supervised or unsupervised learning techniques. In machine learning, machines are made to understand information and made capable to derive some meaning out of them. This learning is done through analogies, connectionist, strategies, discovery, problem solving, search, and match by parameter adjustment. The ability of any machine to learn depends upon the amount of information it can handle and the limit to which it can process [4]. Machine learning is considered as the type of automation that gets enhanced as the amount of input data increases. However, algorithms being used for computation are usually conventional that are designed to solve simple data sets, creating a computational challenges. For example, for big data these are memory and processing for training periods, unstructured data formats, fast moving data, low scalability of algorithms, unbalanced distribution of input data sets, and unlabeled data [17].

Convolutional Neural Networks (CNNs) have been used for accurate modeling of classification data [18], [41], especially image and text data [42]. However, for large datasets, the CNNs needs tremendous amount of processing power. Also, the trend has moved from the traditional feature extraction to autonomous feature extraction as in [19], [20]. However, the main problem is still not thoroughly investigated, which is the increase of features and data instances lead to the curse of dimensionality and the tremendous amount of processing power needed. The curse of dimensionality definitely affects the final model performance. Similarly the continuous increase of data instances forces the machine learning models to be re-calculated and re-evaluated. This thus puts tremendous reliance on the powerful computing machines and resources. However, such facilities are still not available to masses and many research institutes.

This article thus investigates the reduction of data instances for classification performance and machine learning scenarios. For experimental evaluation of the proposed architecture, this article uses the dataset from the NDPI videos. Further details are available in [21]. NDPI is huge dataset and comprises of more than 40 Gigabytes of videos data. For experimental analysis, the data is divided into three classes. These are: Un-acceptable, Acceptable, and Flagged. Though the paper is based on the generic concept of data sampling and performance analysis, however, the article uses the data from image based filtering. It has three main reasons. First is that the data is well organized into three classes, which is a good representative problem for machine learning algorithms? Secondly, though the data is image, in the feature form, the data is converted to numerical values. Thus the data is equated to other datasets and similar machine learning problems. Thirdly, the data is huge, more than 40 Gigabytes in size. Therefore, it is assumed that the data that is processed in this article is big data. Therefore, the results can be extended to other datasets of similar nature.

Our previous work [1] about the role of sampling in big data analysis motivated us for further investigation about effective approaches for big data analysis. Based on the dataset processed in this article, there is considerable work available in the state of the art. The articles [22]-[25] present and models such scenarios and applications.. The work in [22] fuses AlexNet [20] and GoogLeNet [26] and for performance enhancements. The work in [24] takes advantage of colors transformations. The paper [27] presents an evidence combination. The work of [28] takes advantage of adaptive sampling approach for filtering. The paper [29] demonstrates websites filtering analysis, and [30] combines key-frame analysis. The [31] and [32] use visual features for media access and filtering. The articles [33]-[36] are based on content based image retrieval.

The rest of the paper is organized as follow. Section II presents some background about the classifiers used for the study, namely, Support Vector Machines and Random Forest. Section III explains the experimental study and the found results. Discussion of results is presented in Section IV. Finally, Section V concludes the study.

II. Classification

Classifiers draw a decision boundary between the classes in the data. There are several classifiers present. In this article, we use the Support Vector Machine (SVM) and the Random Forest. The SVM has shown great results and in a favorable choice in machine learning and computer vision tasks. Moreover, SVM has been heavily used with the DL features learning and training. The Random forest has also shown considerable good results and is the choice in many classification scenarios.

A. SVM

Support Vector Machine (SVM) is a supervised learning classifier that is introduced in 1990s by Boser, Guyon, and Vapnik [37]. It is widely used because of its accuracy, ability to deal with high-dimensional data, and its flexibility in modeling different sources of data. The SVM has two advantages: first, it has the ability to produce non-linear decision boundaries by using methods of linear classifiers; secondly, the classifier can be applied to data with no fixed-dimensional vector space representation [38]. Moreover, SVM has a robust theoretical foundation, which is statistical learning theory; and successful empirical applications as well. It has been applied to different fields such as hand written digits recognition, text classification, and objects recognition [38]. The SVM is in this article is used due to its over-all good detection performance in similar areas.

B. Random Forest

Two popular methods of classification trees have grabbed researchers’ attention: bagging and boosting. These two methods can generate many classifiers and aggregate their results [39]. One of the important advantages of Random forest is that it can be used for regression or classification problems. In an enhancement addition to bagging, Breiman [40] proposed random forests as an additional layer of randomness. Either in regression or classification problems, Random forest can help in ranking the importance of variables. Random forest has only two parameters: the number of trees in the forest and the number of variables in the node. These two parameters constitute to the straightforwardness of Random forest. Moreover, it constructs every tree with a different bootstrap sample of data, which changes how trees are constructed in regression and classification. Each node is split by the best predictor chosen at the node randomly among a subset of predictors [40]. Many trees are grown and every tree vote for a particular class. The class with high number of trees is the final class assigned to particular data instance.

III. Experimental Setup and Results

A. Features and Dataset

For an evaluation experiments, the article uses datasets from the NDPI videos. Further details of the NDPI dataset is available in [21]. NDPI is huge dataset and comprises of more than 40 Gigabytes of videos data. For experimental analysis, the data is divided into three classes. These are: Un-acceptable, Acceptable, and Flagged. Fig. 1 shows some samples. The experiment setup uses the data from image based filtering and large amount of data. It has three main reasons. First is that the data is well organized into three classes, which is a good representative problem for machine learning algorithms? Secondly, though the data is image, in the feature form, the data is converted to numerical values. Thus the data is equated to other datasets and similar machine learning problems. Thirdly, the data is huge, more than 40 Gigabytes in size. Therefore, it is assumed that the data that is processed in this article is big data. Therefore, the results can be extended to other datasets of similar nature.

For feature extraction, the article uses the Autocorrelogram. We use the F-measure as an evaluation parameter as it is mostly used in the state of the art for similar problems and applications and is favorable for this evaluation as well. The F-measure takes into account the Precision and the Recall.

Fig. 1. Sample Images from NDPI [21].

B. Instance Sampling

In data analysis domains, an instance represents the individual object of which the problem is composed of. This means that if the problem is based on the color, let’s say in computer vision, the instance is the set of pixels for the problem concerned. The instance may also represent a complete image if the features are globally extracted from the images. In most cases, the instance is directly related to the number of objects available for training and testing. If instances are reduced, the training data and ultimately testing data is reduced. If instances are increased, it will mean that the training and testing data is increased. If a ten folds cross validation is used, instance increase results in the increase of 90% training samples, and 10% testing samples. This can have one of three impacts on the results. The result could stay neutral. It can increase in certain cases, and it can also decrease in certain cases.

The neutral case can occur generally in two ways. First one is if the instance added has similar nature to the previous data. This means that the instance is already represented in the model of the machine learning classifier. The addition of this new instance thus has either contributed no extra information. This thus increases the dataset without any benefit to the machine learning model. The second neutral case is when contribution of the instance addition is negligible due to the large number of data samples. This can also mean that the data is already covering most of the model generation cases and no extra addition of data is required.

The increase in classification results due to instance increase can be due to the fact that the new instances contribute strong classification information in the model. It means that the new addition strongly represents the classes in the dataset and also exhibiting strong correlation with the attributes for that instance. This type of scenario is always the objective in machine learning paradigm. However, every machine learning algorithm has certain limits and adding more strong instances may not contribute any information for classification. One of the interesting phenomenon that can occur by adding strong instances is over-fitting. The model can become much diverted to special cases and does not generalize well.

The decrease in performance can be due to either the new instances are not related to the classes in specific problem, or the instance added is representing (adding) strong noise to the model. This phenomenon is most common and collecting correct dataset is the challenge for most machine learning related problems. Therefore, the data cleaning task is essential in many classification tasks for reliable model generation. The decrease in classification performance can also be due to less number of data instances. Many machine learning algorithms require considerable amount of data (not big data) for reliable model generation and generalization for unseen test data instances. However, this does not mean that data increase beyond certain limit will keep on increasing the performance. Every classifier has limits for certain problems and thus thorough analysis is required for the final model generation and the amount of data needed for the particular problem.

In the experimentation setup, the objective is to analyze instance addition and removal on the results of machine learning. The experimental approach however proceeds in reverse manner. The proposed experimentation setup proceeds by reducing instances and analyzing the results. The 50% instances are sampled from original 100% data and the results are noted. The 50% sample is randomly selected from the original data. The sample is thus analyzed based on 90% training data and 10% testing data. This means that from particular 50% data, the 90% of the data is used for training the classifier for model generation. This model generated thus is used to test the 10% data of the 50% sample and performance is noted. This 90% training data the 10% training data from the 50% of the original data is also selected ten times to take average 10 ten permutations. This is to remove biasness. This of taking 50% training data is then repeated 100 times. This generates 1000 experiments.

Fig. 2 shows experiments where the 50% instances are randomly selected for the actual 100% data. Fig. 2 shows performance in terms of an F-measure for the 3 classes' data based on the SVM classifier. The "Actual Data" label in Fig. 2 represents the F-measure of the original 100% data. From Fig. 2, an F-measure of 0.784 is obtained for the 100% data. The F-measure for the average of first ten experiments (labelled "10" in Fig. 2) is 0.733 which is less than the F-measure of the 100% instances. The F-measure for the average of next ten experiments (represented as "20" in Fig. 2) is 0.703. The third set has an F-measure of 0.755. The fourth, fifth, and sixth set has an F-measure of 0.762, 0.757, and 0.77, respectively. The seventh set has a reduced F-measure of 0.722. The eighth and ninth sets have an F-measure of 0.744 and 0.741. The last set of experiments gets an increased F-measure of 0.797. This is even higher than the F-measure of the 100% original set. The average of the all 100 experiments (labelled as "Average of 100") is 0.748. As the F-measure for 100% data is 0.784, therefore, the difference is 0.036. This means the total difference of 3.6% to the original 100% data. This thus means that the model of 100% data is 3.6% more accurate than the sampled data.

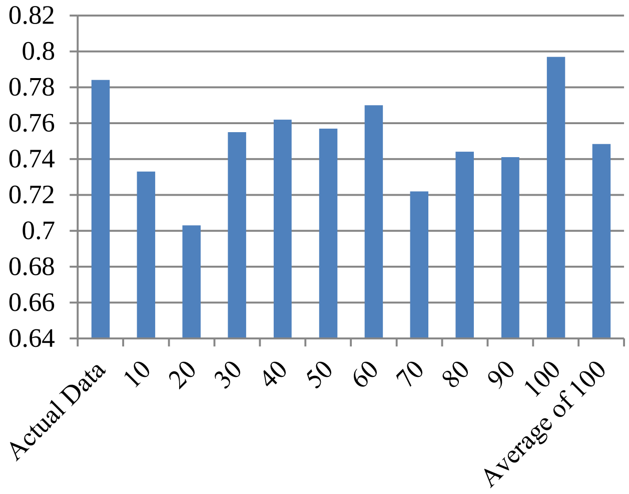

Fig. 2. F-measure for the 3 Classes' Data based on the SVM Classifier for 50% Random Instances. The 10 Interval Sets Show the Average F-measure of 10 Experiments.

Fig. 3 shows experiments with the Random Forest and almost follows on average the SVM scenario. The "Actual Data" label in Fig. 3 represents the F-measure of the original 100% data. From Fig. 3, an F-measure of 0.841 is obtained for the 100% data. The F-measure for the average of first ten experiments (labelled "10") is 0.83 which is less than the F-measure of the 100% instances as was with the case of SVM. The F-measure for the average of next ten experiments (represented as "20") is 0.809. The third set has an F-measure of 0.822. The fourth, fifth, and sixth set has an F-measure of 0.836, 0.824, and 0.814 respectively. The seventh set has a reduced F-measure of 0.818. The eighth and ninth sets have an F-measure of 0.84 and 0.816. The last set of experiments gets an F-measure of 0.83. The average of the all 100 experiments (labelled as "Average of 100") is 0.823. As the F-measure for 100% data is 0.841, therefore, the difference is 0.018. This means the total difference of 1.8% to the original 100% data. This thus means that the model of 100% data is 1.8% more accurate than the sampled data.

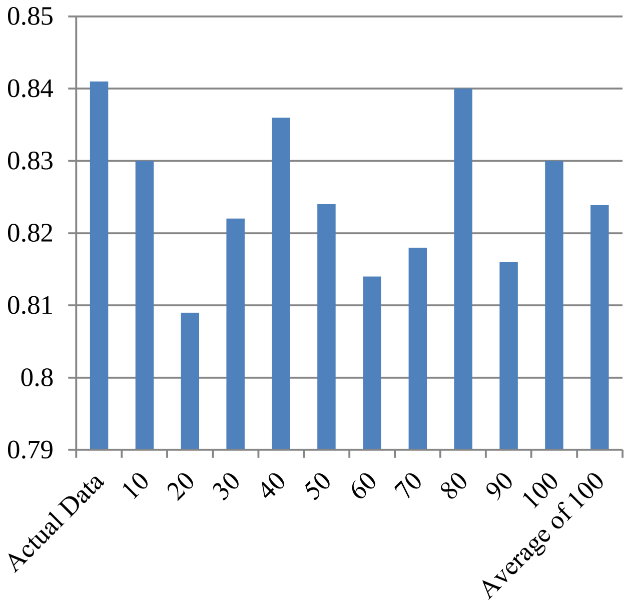

Fig. 3. Random Forest Classifier F-measure for the 3 Classes' Data based on the 50% Random Instances. The 10 Interval Sets Show the Average F-measure of 10 Experiments.

C. Incremental Training and Test Data

The instances of data can also be analyzed based on the amount of training and testing data available. Generally, the more training data, the better is the performance. This may be true in most problems; however, this may not be true in every case because the more the data, the more is the chance of noisy data and thus wrong models. Therefore, to analyze this, in the following set of experiments, the objective is to see if the impact of the amount of training data affects the performance of machine learning algorithms and classifiers. If a smaller set of data can generate good model that can generalize well and have good classification performance, then large of data may not be needed to process, thus saving time and computing resources. Moreover, this enables researchers to quickly analyze datasets on many algorithms which is useful for research and development activities.

Table I shows the F-measure based analysis of the SVM classifier with the incremental increase of training data. In Table I, Training data of "10" means that the 10% of the data is used for generating the model of the Random Forest classifier and 90% data is used for testing this model. The F-measure for such scenario is 0.648. Increasing the training data to 20% and reducing the testing data to 80% gets an F-measure of 0.718. At 30% training data, the F-measure increases to 0.742. With 40, 50, and 60 percent, the F-measure obtained is 0.744, 0.776, and 0.745 respectively. At 70%, an increase F-measure of 0.768 is obtained. At 80%, the F-measure keeps increasing to 0.799. At 90% training data and 10% testing data, the F-measure normalizes at 0.806.

Table I shows that the increase in training data and its relation to the F-measure is not consistent in all cases. For example, the F-measure at the 60% training is lower than the 50% training, which in theory should be higher. Similarly, in case of the 70% training, the F-measure is less than 50% training.

TABLE I. F-MEASURE OF SVM FOR INCREMENTALLY INCREASING (BY 10%) THE TRAINING DATA STARTING FROM 10%. SIMILARLY, INCREMENTALLY REDUCING (BY 10%) THE TESTING DATA STARTING FROM 90%

Training Data	**Testing Data**	**F-measure (SVM)**
10	90	0.648
20	80	0.718
30	70	0.742
40	60	0.744
50	50	0.776
60	40	0.745
70	30	0.768
80	20	0.799
90	10	0.806

Table II shows the F-measure based analysis of the Random Forest classifier with incremental increase of training data. The F-measure 10% training data is 0.706. Increasing the training data to 20% and reducing the testing data to 80% gets an F-measure of 0.77. At 30% training data, the F-measure increases to 0.785. With 40, 50, and 60 percent, the F-measure obtained is 0.769, 0.806, and 0.803 respectively. At 70%, an increase F-measure of 0.854 is obtained. At 80%, the F-measure keeps increasing to 0.865. At 90% training data and 10% testing data, the F-measure normalizes at 0.854.

Table II shows the increase in training data and its relation with the F-measure is not consistent in all cases. For example, the F-measure at the 40% training is lower than the 30% training. Similarly, in case of the 90% training, the F-measure is less than 80% training.

Both in the Tables I and II, the increase in training data and its relation to the F-measure is not consistent in all cases. This could be due to many reasons. One of the reasons is that the increasing number of samples can add noise and thus more training data does not mean good final trained model. Secondly, since the selection of training data is random, the training sample does not pick many instances of the “good” representative samples.

TABLE II. F-MEASURE OF RANDOM FOREST FOR INCREMENTALLY INCREASING (BY 10%) THE TRAINING DATA STARTING FROM 10%. SIMILARLY, INCREMENTALLY REDUCING (BY 10%) THE TESTING DATA STARTING FROM 90%

Training Data	**Testing Data**	**F-measure (Random Forest)**
10	90	0.706
20	80	0.77
30	70	0.785
40	60	0.769
50	50	0.806
60	40	0.803
70	30	0.854
80	20	0.865
90	10	0.854

IV. DISCUSSION OF RESULTS

Experiments in both the Fig. 2 and 3 depict interesting results. With these experiments, it is observed that 50% reduction does not drastically harm the overall model. On average, in SVM only 3.6%, and in Random Forest, only 1.8% performance is reduced if only 50% data is used. This is acceptable in most cases unless there is a serious nature of the problem in hand. The benefit one gets is the processing of extremely reduced size and sets of data instances. This is useful in number of scenarios. 50% reduction in instances means that in most cases, the data will fit easily in the RAM and the processing times will be considerably reduced, benefitting in terms or resources. Other benefits are that since data generation and gathering is not an easy task, the less number of instances means that less but clean data can be useful in many cases.

Fig. 4 shows the follow of the F-measure plotted against the amount of training data. The X-Axis shows the training samples. The Y-Axis shows the F-measure. Fig. 4 shows slightly incremental increase in F-measure for both the SVM and the Random Forest in many cases. However, interestingly, it can be seen that even with the 10% training data, there is not a huge jump and difference in the F-measure of consecutive incremental sets of data in both the SVM and the Random Forest cases. From this it can be deduced that in special cases, smaller sub-sampled data can be used for model generation in machine learning problems. This is useful in cases, where there are either limitations on hardware or one has to select among many available machine learning algorithms. The second being the most common scenario. The first case of hardware resources is more defined in the case of processing big datasets. The ever increasing data has put tremendous limitations on the processing power of many available machines. Many researchers need special hardware to process large amount of data that is expensive and is mostly still not available to some research groups and teaching environments. This experimental setup shows that in special cases, generating many random models from the smaller samples and averaging its performance can represent larges datasets. This type of reliance on smaller models is thus useful in quick model analysis and experimentation, where the final model can be then generated by processing big datasets.

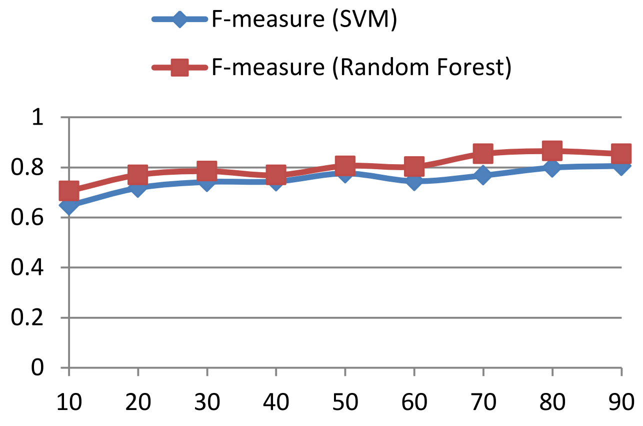

Fig. 4. F-measure for Incrementally Increasing (by 10%) the Training Data Starting from 10%. Similarly, Incrementally Reducing (by 10%) the Testing Data Starting from 90%.

V. Conclusion

Big data processing requires large computing resources. This puts bottleneck not only on processing data but also many researchers don’t get the freedom to analyze large datasets. This article analyzed large amount of data from different perspectives. One of them is the processing reduced sets of large data with less computing resources. Thus the article analyzed 40 GB data, by testing different strategies of reducing the processed data without losing and compromising on the detection and model learning in machine learning. Many alternatives were analyzed and it is observed that 50% reduction does not drastically harm the machine learning model performance. On average, in SVM only 3.6%, and in Random Forest, only 1.8% performance is reduced if only 50% data is used. This is acceptable in most cases unless there is a serious nature of the problem in hand. The benefit one gets is the ability and freedom of processing of extremely reduced size and sets of data instances. This is useful in number of scenarios. The 50% reduction in instances means that in most cases, the data will fit easily in the RAM and the processing times will be considerably reduced, benefitting in execution, time and or resources. From the incremental training and testing experiments, it is found that in special cases, smaller sub-sampled data can be used for model generation in machine learning problems. This is useful in cases, where there are either limitations on hardware or one has to select among many available machine learning algorithms. The second point being the most common scenario in machine learning research. In future, the experimentation setup will be expended to massive parallel architecture for large collection of data sets including textual data. Also, the DL will be analyzed for sampled based training and testing.

Acknowledgement

The work in this article is funded in its entirety by the Dean of Scientific Research (SRD), Project number: coc-2018-1-14-S-3603 at the Qassim University, Kingdom of Saudi Arabia

References

[1] W. Albattah, “The Role of Sampling in Big Data Analysis,” in Proceedings of the International Conference on Big Data and Advanced Wireless Technologies - BDAW ’16, 2016, pp. 1–5.

[2] M. Hilbert, “Big Data for Development: A Review of Promises and Challenges,” Dev. Policy Rev., vol. 34, no. 1, pp. 135–174, Jan. 2016.

[3] D. A. Reed and J. Dongarra, “Exascale computing and big data,” Commun. ACM, vol. 58, no. 7, pp. 56–68, 2015.

[4] A. L’Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, “Machine Learning With Big Data: Challenges and Approaches,” IEEE Access, vol. 5, no. 1, pp. 7776–7797, 2017.

[5] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, “Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests,” Inf. Sci. (Ny)., vol. 278, pp. 488–497, 2014.

[6] R. Clarke, “Big data, big risks,” Inf. Syst. J., vol. 26, no. 1, pp. 77–90, Jan. 2016.

[7] D. Sullivan, “Introduction to big data security analytics in the enterprise.” [Online]. Available: https://searchsecurity.techtarget.com/feature/Introduction-to-big-data-security-analytics-in-the-enterprise. [Accessed: 31-Jul-2018].

[8] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, “Big data analytics: a survey,” J. Big Data, vol. 2, no. 1, p. 21, Dec. 2015.

[9] G. Bello-Orgaz, J. J. Jung, and D. Camacho, “Social big data: Recent achievements and new challenges,” Inf. Fusion, vol. 28, pp. 45–59, Mar. 2016.

[10] J. Zakir, T. Seymour, and K. Berg, “Big Data Analytics,” Issues Inf. Syst., vol. 16, no. 2, pp. 81–90, 2015.

[11] U. Sivarajah, M. M. Kamal, Z. Irani, and V. Weerakkody, “Critical analysis of Big Data challenges and analytical methods,” J. Bus. Res., vol. 70, pp. 263–286, Jan. 2017.

[12] K. Engemann et al., “Limited sampling hampers ‘big data’ estimation of species richness in a tropical biodiversity hotspot.,” Ecol. Evol., vol. 5, no. 3, pp. 807–820, 2015.

[13] J. K. Kim and Z. Wang, “Sampling techniques for big data analysis in finite population inference,” Jan. 2018.

[14] S. Liu, R. She, and P. Fan, “How Many Samples Required in Big Data Collection: A Differential Message Importance Measure,” Jan. 2018.

[15] J. Bierkens, P. Fearnhead, and G. Roberts, “The Zig-Zag Process and Super-Efficient Sampling for Bayesian Analysis of Big Data,” Jul. 2016.

[16] J. Zhao, J. Sun, Y. Zhai, Y. Ding, C. Wu, and M. Hu, “A Novel Clustering-Based Sampling Approach for Minimum Sample Set in Big Data Environment,” Int. J. Pattern Recognit. Artif. Intell., vol. 32, no. 2, pp. 1–10, Feb. 2018.

[17] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, “Machine learning on big data: Opportunities and challenges,” Neurocomputing, vol. 237, no. 1, pp. 350–361, 2017.

[18] D. Kotzias, M. Denil, N. de Freitas, and P. Smyth, “From Group to Individual Labels Using Deep Features,” Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD ’15, pp. 597–606, 2015.

[19] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning Hierarchical Features for Scene Labeling,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. Curran Associates Inc., pp. 1097–1105, 2012.

[21] “Pornography Database.” [Online]. Available: https://sites.google.com/site/pornographydatabase/. [Accessed: 09-Nov-2017].

[22] M. Moustafa, “Applying deep learning to classify pornographic images and videos,” Nov. 2015.

[23] A. P. B. Lopes, S. E. F. de Avila, A. N. A. Peixoto, R. S. Oliveira, M. de M. Coelho, and A. de A. Araújo, “Nude Detection in Video Using Bag-of-Visual-Features,” in 2009 XXII Brazilian Symposium on Computer Graphics and Image Processing, 2009, pp. 224–231.

[24] A. Abadpour and S. Kasaei, “Pixel-Based Skin Detection for Pornography Filtering,” Iran. J. Electr. Electron. Eng., vol. 1, no. 3, pp. 21–41, 2005.

[25] R. Ullah and A. Alkhalifah, “Media Content Access: Image-based

Filtering," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 3, 2018.

[26] C. Szegedy et al., "Going Deeper with Convolutions," Sep. 2014.

[27] E. Valle, S. Avila, F. Souza, M. Coelho, and A. de A. Araujo, "Content-Based Filtering for Video Sharing Social Networks," in XII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais—SBSeg 2012, 2011, p. 28.

[28] P. Monteiro, S. Eleuterio, M. De, and C. Polastro, "An adaptive sampling strategy for automatic detection of child pornographic videos."

[29] N. Agarwal, H. Liu, and J. Zhang, "Blocking objectionable web content by leveraging multiple information sources," ACM SIGKDD Explor. Newsl., vol. 8, no. 1, pp. 17–26, Jun. 2006.

[30] C. Jansohn, A. Ulges, and T. M. Breuel, "Detecting pornographic video content by combining image features with motion information," in Proceedings of the seventeen ACM international conference on Multimedia - MM '09, 2009, p. 601.

[31] J.-H. Wang, H.-C. Chang, M.-J. Lee, and Y.-M. Shaw, "Classifying Peer-to-Peer File Transfers for Objectionable Content Filtering Using a Web-based Approach."

[32] Hogyun Lee, Seungmin Lee, and Taekyong Nam, "Implementation of high performance objectionable video classification system," in 2006 8th International Conference Advanced Communication Technology, 2006, p. 4 pp.-pp.962.

[33] D. Liu, X.-S. Hua, M. Wang, and H. Zhang, "Boost search relevance for tag-based social image retrieval," in 2009 IEEE International Conference on Multimedia and Expo, 2009, pp. 1636–1639.

[34] J. A. Da, S. Júnior, R. E. Marçal, and M. A. Batista, "Image Retrieval: Importance and Applications."

[35] S. Badghaiya and A. Bharve, "Image Classification using Tag and Segmentation based Retrieval," Int. J. Comput. Appl., vol. 103, no. 15, pp. 20–23, Oct. 2014.

[36] A. N. Bhute and B. B. Meshram, "Text Based Approach For Indexing And Retrieval Of Image And Video: A Review," Apr. 2014.

[37] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in Proceedings of the fifth annual workshop on Computational learning theory - COLT '92, 1992, pp. 144–152.

[38] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," J. Mach. Learn. Res., vol. 2, no. 11, pp. 45–66, 2001.

[39] A. Liaw and M. Wiener, "Classification and Regression by randomForest," R News, vol. 2, no. 3, pp. 1–10, 2002.

[40] L. Breiman, "Random Forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[41] Kowsari, Kamran, Mojtaba Heidarysafa, Donald E. Brown, Kiana Jafari Meimandi, and Laura E. Barnes. "RMDL: Random Multimodel Deep Learning for Classification." In Proceedings of the 2nd International Conference on Information System and Data Mining, pp. 19-28. ACM, 2018.

[42] Kowsari, Kamran, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. "Hdltex: Hierarchical deep learning for text classification." In Machine Learning and Applications (ICMLA), pp. 364-371, 2017.