Hybrid Machine Learning-Based Intelligent Technique for Improved Big Data Analytics

Andronicus A. Akinyelu
Department of Computer Science and Informatics
University of the Free State
Bloemfontein, South Africa
e-mail: akinyeluaa@ufs.ac.za

***Abstract*—The average volume of data produced daily is estimated to be over 2.5 quintillion byte. Moreover, by year 2020, it is estimated that 1.79MB of data will be created every second by each person in the world. Apparently, big datasets contain tremendous amount of valuable information that can be used for improved decision making. However, big data requires incredible amount of storage and computational resources for effective processing. Machine Learning (ML) algorithms are effective tools popularly used to analyze and extract concealed insights from datasets. However, some ML algorithms were not originally designed to handle big datasets, hence their computational complexity decreases with increase in data size. Consequently, this makes big data analytics extremely slow or unrealistic. Therefore, there is an obvious need for fast and effective techniques for big data analytics. This paper introduces an intelligent hybrid ML-based technique suitable for big data analytics (called EDISA_ML). EDISA_ML is a boundary detection and instance selection algorithm, inspired by edge detection in image processing. It was evaluated on four ML algorithms and big datasets, and the results show that it achieved a storage reduction of over 50% and simultaneously improved the training speed of the evaluated ML algorithms by over 93% (in some cases), without meaningfully affecting their prediction accuracy.**

Keywords-machine learning; big data analytics; instance selection; data reduction; boundary detection

I. INTRODUCTION

Due to the wide spread of information technology, huge amount of data is currently exchanged digitally or on the internet. This has led to the dispensation of big datasets, and consequently led to the speed optimization problem experienced by big data analytics. In view of this, some techniques [1-6] have been presented in literature to handle this problem, and some of them were designed to improve the speed of data analytics, including sampling [7], Machine Learning (ML) [8], and metaheuristics [6]. Leyva et al. [3] proposed three data reduction techniques using the concept of local set [9]. Besides, Carbonera introduced two techniques (LDIS [4] and XLDIS [10]) for selecting relevant instances from datasets. In addition, Rathee et al. [6] introduced a data reduction technique for multi-objective frameworks using Genetic Algorithm. Some of these techniques were used alone or combined with traditional data mining techniques to achieve improved data analytics [11]. Although, some traditional data analytics methods perform excellently when applied to small-scale dataset, they were not originally designed to handle large-scale datasets [11]. Therefore, there is an obvious need for improved big data analytics techniques. The contributions of this paper are as follows:

1) This paper introduces a hybrid ML-based technique suitable for improved big data analytics. The technique is divided into two stages: boundary instance selection stage and model building stage. At the first stage, an instance selection technique is used to select relevant instances from large or medium-scale datasets. At the second stage, the selected instances are used to build improved and fast learning models.
2) The proposed technique is applied to five large or medium-scale datasets and four ML algorithms. Experimental results show that they achieved a storage reduction capacity of over 50% and concurrently improved the training speed of ML algorithms by over 93%, without significantly affecting their prediction accuracy. The improved performance comes with the following advantages: improved computational complexity, improved prediction accuracy, improved computational storage space, and improved big data analytics.

II. EDGE DETECTION

In image processing, edge detection is used to find the boundaries or edges of objects in images. Object boundaries refers to locations in images that has sharp discontinuities in image brightness. Images generally contains redundant or irrelevant data, and these data does not contribute significantly to the prediction accuracy of classifiers. Hence, to reduce computational complexity, the irrelevant data need to be removed from the dataset. In view of this, edge detection is applied to images with the objective of selecting relevant features and consequently reducing the size of the image. Edge detection preserves important structural properties of images and computer space. Inspired by edge detection, the proposed technique is designed to select border instances from big datasets. Border instances provide useful information for segregating distinct classes.

A. Boundary Detection Algorithm

978-1-7281-4577-8/19/$31.00 ©2019 IEEE

This paper introduces a hybrid ML-based boundary detection and instance selection technique for big data analytics, called Edge Detection Instance Selection Algorithm for Machine Learning algorithms (EDISA_ML). It borrows the concept of edge detection in images. Edge detection algorithms selects object located at edges, and EDISA_ML (and other boundary detection algorithms) is designed to select instances (called border instances) close to the boundary. The full algorithm of EDISA_ML is shown in Figure 1. EDISA_ML is divided into two stages: boundary instance selection stage and model building stage. At the first stage, the algorithm starts by initializing the vote count for each instance in a dataset (line 1). The vote count shows the number of times each instance is voted as an edge instance by other instances. Moreover, in line 3, the algorithm selects *M* instances from the training dataset. Furthermore, EDISA_ML calculate the neighborhood list for each $instance_j$ in the dataset. It achieves this by calculating the squared Euclidian distance between $instance_j$ and the other instances in the dataset (line 6). Besides, for each $instance_j$, a corresponding edge instance ($instance_k$) is voted (line 8). $Instance_k$ is voted as the edge instance of $instance_j$, if it has the largest Euclidean distance among all the instances in the neighborhood list of $instance_j$. In addition, the count for every voted instance is increased in line 12. The process continues until all instances and their respective neighborhood list has been processed. Finally, the algorithm selects the instance with the highest vote count (line 14), and then use *k*-NN to select the nearest neighbors to the selected (or voted) instance. The second stage of the algorithm simply involves training the ML algorithms with the border instances selected from the first stage.

TABLE I. DATASET INFORMATION

Dataset name	Dataset size	Feature size	Class size	#Train samples	#Test samples
Letter	20,000	16	26	16,000	4000
Optdigit	5620	64	10	3823	1797
Pentdigit	10,992	16	10	7494	3498
Twitter	140,707	77	2	112566	28141
USPS	9298	256	10	7291	2007

III. EXPERIMENTS

The performance of the proposed technique is compared to two instance selection techniques adopted in this research for comparasion: MCIS [12] and CBD [13]. The aspects of comparison is as follows (a) ability to preserve the classification accuracy (b) data reduction ability (c) training speed, and (d) time for instance selection. Specifically, we performed data reduction on each dataset five times and calculated their average prediction accuracy (Acc), storage reduction percentage (Stor), training time (Train-T), and algorithm time (Sel-T). Storage reduction percentage refers to the fraction of instance selected after data reduction.

A. Experiments Settings and Datasets

As shown in Table I, the proposed technique was evaluated on five medium or large-scale datasets. Four of the datasets were obtained from UCI dataset repository [14], and the USPS (US Postal Service) dataset is provided by Hull [15]. All the datasets (except Twitter dataset) were divided into training and test set by their various providers. We used 80% of the Twitter datasets for training and used the remaining for testing. During the experiments, the boundary detection algorithm was used to select relevant instances from the big datasets, and the selected instances were used to train four ML algorithms, including ANN, RF, NB and BayesNet. In this paper, we refer to the classification models produced by the reduced subset as hybrid models and the models produced by the whole datasets as standard models. All the experiments were performed on a popular ML library platform, called WEKA [8]. Besides, all the experiments were performed on a computer with the following specifications: windows 10, 8GB RAM, Intel Core i5, 64-bit, 1.70GHz (4CPUs). The parameters used by EDISA_ML were selected through experiments.

B. Result and Discussion

Table II shows the results obtained from all the experiments. As shown in the results, in all cases, the hybrid models ourperformed the standard models, in terms of training speed. Speicifically, Table III shows that the proposed technique improved the training speed of BayesNet, ANN, NB, and RF by an average of 64%, 55%, 62%, and 58% respectively. The speed improvement is calculated using equation (1). Moreover, as shown in the results, EDISA_ML is very efficient in processing big datasets; it require very little time to select relevant instances from big datasets. Specifically, it used an average of approximately 3 seconds, 54 seconds, 4 seconds, 45 seconds, and 86 seconds to select instances from Optdigit, Letter, Pentdigit, USPS, and Twitter datasets, respectively. Interestingly, the storage reduction did not compromise the prediction accuracy of the evaluated ML algorithms. Infact, as shown in Figure 2, in some cases, the hybrid models produced better prediction accuracy than the standard models. This demostrate their capacity to preserve and improve the prediction accuracy of ML algorithms. Moreover, as shown in Figure 3, EDISA_ML has excellent data reduction capacity, making it very useful for big data analytics. It reduced all the big datasets by over 50% in most cases (and over 93% in some cases), without compromising the dataset quality. The reduced dataset improves the speed, complexity and quality of big data analytics. Morever, it simplifies & enhances the process of decision making. It is noteworthy to mention that EDISA_ML reduced the Twitter dataset by over 93% without significantly affecting the

classification accuracy of the resultant models. This demostrate the data reduction ability of the proposed technique.

$$((\alpha - \beta)/\alpha) * 100 \quad (1)$$

where α *and* β refers to the training speed produced by the standard and hybrid models, respectively.

EDISA_ML

Input: *N, Nsub, K /* N = size of training dataset, Nsub=size of training subset, K=number of k nearest neighbours */*

Output: EI[] */* edge instances for training */*

```
Initialize Vote[N] /* Initialize vote count*/
DECLARE farthest, dist[,], index
Select M instances from dataset, where M = Nsub
For j = 1 to N
    For k = 1 to N
        dist[j,k] = SquaredEuclideanDistance(instance_j, instance_k), where j ≠ k
        if dist[j,k] > farthest
            farthest ← dist[j,k] /*get the farthest instance_k from instance_j */
            index ← k /*save the position of selected instance_k */
        end if
    End k
    Vote[index]+= 1 /*Increase vote count */
End j
E ← Vote.Max() /*Select the instance with majority vote (i.e. edge instance) */
EI ← ComputeKNN(E) /*Select k instances close to the selected edge instance */
Return EI
```

Figure 1. Edge detection instance selection algorithm for machine learning algorithms (EDISA_ML).

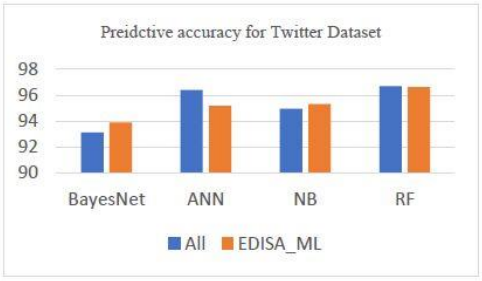

Figure 2. Preidctive accuracy for the standard and hybrid model.

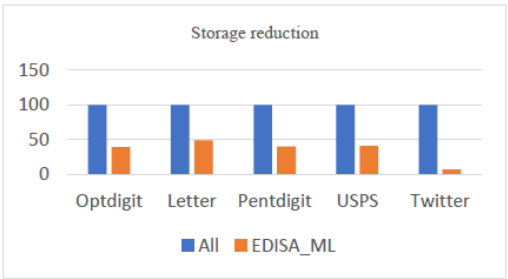

Figure 3. Storage reduction for the standard and hybrid models.

The robustness of the proposed technique was further demostrated by comparing it to two existing instance selection techniques (MCIS and CBD) adopted in this study for comparison. As shown in Table II, EDISA_ML outperformed MCIS and CBD in terms of storage reduction, making it a preferred option for big data analytics. In addition, EDISA_ML produced better prediction accuracy than MCIS in most cases, and CBD in some cases. Moreover, EDISA_ML outperfrom CBD in term of training speed and instance selection speed. Overall, the result produced by EDISA_ML shows that it is very competitive and effective. Besides, it show that it can efficiently reduce big datasets, improve the performance of big data analytics and ML algorithms.

IV. CONCLUSION

Due to the rapid production of data from different sources, big data analytics is becoming a big problem that requires serious attention. Many effective traditional data analytics techniques have been proposed to handle this problem, however, some of them were not originally designed to tackle big data analytics. This paper presents a hybrid ML-based technique for improved big data analysis (called EDISA_ML). The technique is divided into two stages. At the first stage, a boundary detection algorithm is used to select boundary instances from big datasets, and at the second stage, the selected boundary instances are used to build fast and efficient ML models. The proposed technique was evaluated on four ML algorithms and big datasets.

Experimental results show that EDISA_ML achieved a storage reduction of over 50% in most cases, and 93% in some cases. The improved storage reduction makes data analytics computationally inexpensive and faster. It also improved the training speed of ML algorithms by over 90% without significantly affecting their prediction accuracy (and in some cases, improved their prediction accuracy). In addition, the proposed technique was compared to two existing instance selection techniques and it produced competitive results. Summarily, the results produced by the proposed technique shows that it is fast and efficient, with very good storage reduction capacity. The results also show that EDISA_ML can satisfactorily reduce the training speed of ML algorithms without simultaneously affecting their prediction accuracy. Finally, the results show that EDISA_ML is very useful for big data analytics and ML speed optimization.

TABLE II. RESULTS FOR THE MEDIUM OR LARGE-SCALE DATASETS

		BayesNet				ANN				Naïve Bayes				Random Forest			
Datasets	**Param**	**All**	**EDISA_ML**	**MCIS**	**CBD**	**All**	**EDISA_ML**	**MCIS**	**CBD**	**All**	**EDISA_ML**	**MCIS**	**CBD**	**All**	**EDISA_ML**	**MCIS**	**CBD**
	Acc (%)	90.21	88.98	89.46	89.66	96.55	94.82	95.10	95.86	89.42	87.75	88.38	89.00	97.38	95.32	95.76	96.36
	Train-T(s)	0.23	0.104	0.11	0.14	278.16	117.88	99.92	152.46	0.08	0.034	0.05	0.05	2.6	1.54	1.56	1.86
	Sel-T(s)	-	3.44	0.64	253.85	-	3.44	0.64	253.85	-	3.44	0.64	253.85	-	3.44	0.64	253.85
Optdgit	Stor (%)	100	39.24	41.67	60.01	100	39.24	41.67	60.01	100	39.24	41.67	60.00	100	39.24	41.67	60.00
	Acc (%)	73.23	71.4	69.48	72.25	80.96	80.05	79.72	80.97	62.3	62.38	60.78	62.57	96.18	93.17	92.02	94.34
	Train-T (s)	0.33	0.20	0.11	0.19	365.31	218.87	164.96	252.35	0.1	0.06	0.04	0.07	11.29	7.68	5.20	9.70
	Sel-T (s)	-	53.99	1.15	1606.31	-	53.99	1.15	1606.31	-	53.99	1.15	1606.31	-	53.99	1.15	1606.31
Letter	Stor (%)	100	48.75	41.67	60	100	48.75	41.67	60	100	48.75	41.67	60	100	48.75	41.67	60
	Acc (%)	83.53	82.45	82.81	83.42	89.82	90.45	91.10	91.45	82.13	82.48	81.52	81.99	96.59	95.71	95.27	95.83
	Train-T (s)	0.19	0.07	0.06	0.11	74.25	48.99	32.84	45.33	0.07	0.03	0.02	0.04	4.46	2.03	1.83	4.26
	Sel-T (s)	-	4.00	0.44	564.47	-	4.00	0.44	564.47	-	4.00	0.44	564.47	-	4.00	0.44	564.47
PentDigit	Stor (%)	100	40.03	41.66	59.99	100	40.03	41.66	59.99	100	40.03	41.66	59.99	100	40.03	41.66	59.99
	Acc (%)	81.96	81.81	81.40	81.89	94.32	93.07	92.86	93.87	76.78	74.99	75.15	76.12	93.37	92.58	92.10	92.89
	Train-T (s)	4.53	1.36	1.42	1.93	5047.81	2434.1	2354.37	5032.53	0.65	0.25	0.31	0.35	19.06	6.46	9.48	11.89
	Sel-T (s)	-	44.85	2.06	630.38	-	44.85	2.06	630.38	-	44.85	2.06	630.38	-	44.85	2.06	630.38
USPS	Stor (%)	100	41.15	41.67	60.01	100	41.15	41.67	60.01	100	41.15	41.67	60.01	100	41.15	41.67	60.01
	Acc (%)	93.11	93.87	94.04	93.04	96.41	95.18	96.04	96.29	94.96	95.31	95.49	94.75	96.69	96.64	94.98	96.64
	Train-T (s)	20.77	0.64	9.55	5.61	8859.44	535.38	2794.68	2918.04	4.46	0.19	1.68	0.83	275.06	4.95	1.60	43.17
	Sel-T (s)	-	86.19	47.70	3678.21	-	86.19	47.70	60.00	-	86.19	47.70	60.00	-	86.19	47.70	60.00
Twitter	Stor (%)	100	6.93	41.67	60.00	100	6.93	41.67	3678.21	100	6.93	41.67	3678.21	100	6.93	41.67	3678.21

TABLE III. TRAINING SPEED IMPROVEMENT

Datasets	**BayesNet (%)**	**ANN (%)**	**NB (%)**	**RF (%)**
Optdigit	54.78261	57.62151	57.5	40.76923
Letter	39.39394	40.0865	40	31.9752
Pendigit	63.15789	34.0202	57.14286	54.4843
USPS	69.97792	51.77909	61.53846	66.10703
Twitter	96.91863	93.95695	95.73991	98.20039
Average	**64.8462**	**55.49285**	**62.38425**	**58.30723**

REFERENCES

[1] A. A. Akinyelu and A. O. Adewumi, "On the Performance of Cuckoo Search and Bat Algorithms Based Instance Selection Techniques for SVM Speed Optimization with Application to e-Fraud Detection," *KSII Transactions on Internet & Information Systems,* vol. 12, no. 3, 2018.

[2] S. Garcı, I. Triguero, C. J. Carmona, and F. Herrera, "Evolutionary-based selection of generalized instances for imbalanced classification," *Knowledge-Based Systems,* vol. 25, no. 1, pp. 3-12, 2012.

[3] E. Leyva, A. González, and R. Pérez, "Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective," *Pattern Recognition,* vol. 48, no. 4, pp. 1523-1537, 2015/04/01/ 2015.

[4] J. L. Carbonera and M. Abel, "A Density-Based Approach for Instance Selection," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, Italy, 2015, pp. 768-774.

[5] J. L. Carbonera and M. Abel, "Efficient instance selection based on spatial abstraction," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2018, pp. 286-292.

[6] S. Rathee, S. Ratnoo, and J. Ahuja, "Instance Selection Using Multi-objective CHC Evolutionary Algorithm," in Information and Communication Technology for Competitive Strategies, ed: Springer, 2019, pp. 475-484.

[7] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold, "Efficient biased sampling for approximate clustering and outlier detection in large data sets," *IEEE Transactions on Knowledge and Data Engineering,* vol. 15, no. 5, pp. 1170-1187, 2003.

[8] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2016.

[9] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," *Data mining and knowledge discovery,* vol. 6, no. 2, pp. 153-172, April, 2002.

[10] J. L. Carbonera, "An Efficient Approach for Instance Selection," in *Big Data Analytics and Knowledge Discovery*, Cham, 2017, pp. 228-243.

[11] G. S. S. L. Alekhya, E. L. Lydia, and N. Challa, "Big Data Analytics: A Survey," *Journal of Big Data,* vol. 2, no. 21, pp. 1-32, 2015.

[12] J. Chen, C. Zhang, X. Xue, and C.-L. Liu, "Fast instance selection for speeding up support vector machines," *Knowledge-Based Systems,* vol. 45, pp. 1-7, 2013.

[13] N. Panda, E. Y. Chang, and G. Wu, "Concept boundary detection for speeding up SVMs," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 681-688.

[14] K. Bache and M. Lichman. (2013), "UCI machine learning repository". available at: http://archive.ics.uci.edu/ml (accessed 12-May-2017).

[15] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 16, no. 5, pp. 550-554, 1994.