VOL 4 (2020) NO 4

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Large Dataset Classification Using Parallel Processing Concept

Mohammad Aljanabi[a,b], Hind Ra'ad Ebraheem[a], Zahraa Faiz Hussain[a], Mohd Farhan Md Fudzee[c], Shahreen Kasim[c], Mohd Arfian Ismail[d], Dwiny Meidelfi[e], Aldo Erianda[e]

[a]*Department of computer science, Alsalam university college, Baghdad, Iraq*
E-mail: mohammad.cs88@gmail.com; hindraad81@gmail.com; msc_zahraa@yahoo.com

[b]*Department of computer science, Aliraqia university/college of education, Iraq*
E-mail: mohammad.cs88@gmail.com

[c]*Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia (UTHM), Malaysia*

[d]*Faculty of Computing, College of Computing and Applied Sciences, University Malaysia Pahang, Pahang, Malaysia*
E-mail: arfian@ump.edu.my

[e]*Department of Information Technology, Politeknik Negeri Padang, Sumatera Barat, Indonesia*
E-mail: dwinymeidelfi@pnp.ac.id, aldo_pnp@pnp.ac.id

***Abstract*— Much attention has been paid to large data technologies in the past few years mainly due to its capability to impact business analytics and data mining practices, as well as the possibility of influencing an ambit of a highly effective decision-making tools. With the current increase in the number of modern applications (including social media and other web-based and healthcare applications) which generates high data in different forms and volume, the processing of such huge data volume is becoming a challenge with the conventional data processing tools. This has resulted in the emergence of big data analytics which also comes with many challenges. This paper introduced the use of principal components analysis (PCA) for data size reduction, followed by SVM parallelization. The proposed scheme in this study was executed on the Spark platform and the experimental findings revealed the capability of the proposed scheme to reduce the classifiers' classification time without much influence on the classification accuracy of the classifier.**

***Keywords*— Large dataset; Parallel SVMs; PCA; Apache Spark.**

I. INTRODUCTION

The volume of data generated by most application from different fields of life on a daily basis has reached the petabyte scale. This rate of data generation has caused a shift in the data-processing techniques. Big data is a term used to describe a large collection of datasets which may be difficult to process using the conventional data processing tools. Several studies have been dedicated to the storage, handling, and retrieval of information from big data [1].

The existing approach of data processing using a single PC has become obsolete with the massive volume of data generated on a daily basis. This has necessitated the use of stronger computing platforms such as parallelism and cloud computing to process such big data. Data processing using a distributed system involves splitting the huge data into smaller tasks which can be easily handled using one or more computers which run parallelly and communicate with one another via message passing [2]

The major concept of data parallelism is to fragment a large dataset D into smaller data subsets (D1, D2,…, Dn). Each of these subsets may or may not have a duplicate data sample. The next step involves the implementation of a data mining framework in a given number of machines (or nodes) which will execute the task individually on each subset. Lastly, the results from all the individual machines are combined using one combination criterion to generate the overall output [3]. The main contribution in this study is the parallel implementation of the PCA with SVM on Spark.

A study by [4] first proposed an SVM-based performance evaluation scheme for used in analyzing parallel computing frameworks in terms of their performance. They also presented the outcome of a set of analysis using the suggested analytical performance model and comparatively evaluated MARS and Spark using representative workloads

based on scalability and performance. The outcome of the experiments showed the proposed model to achieve a better accuracy compared to the MLR in terms of execution time prediction. The results also showed the proposed model to offer a resource utilization requirement. Finally, benchmark studies were conducted with MARS and Spark in which MARS presented a better performance compared to Spark in terms of execution speed and throughput due to its large number of GPU threads which are capable of handling higher parallelism. The evaluation also showed Spark to achieve a lower latency compared to MARS with respect to the execution of 4 benchmark functions Another study by [5] presented a Spark-Chi-SVM model the detection of network intrusion. This model used Chi Sq Selector for feature selection to develop an intrusion detection model based on the SVM on Apache Spark Big Data platform. The trained the developed model using KDD'99 dataset. During the experiments, they compared the Chi-SVM classifier with Chi-LR classifier and the results showed Spark Chi-SVM model to perform better in minimizing the training time which is significant for Big Data. An efficient algorithm was presented by[6] using Apache Spark. Here, the performance of the algorithm was evaluated using various performance metrics and the outcome showed the algorithm to be efficient for concept generation and lattice graph construction compared to the current algorithms. There are several algorithms which are currently available for the identification of the formal concepts and construction of the digraph from the established concepts in large datasets. However, these existing algorithms are not efficient for concept generation owing to the iterative nature of the concept generation process. The existing algorithms are executed using distributed frameworks such as MapReduce and Open MP which are not suitable for iterative applications. This has raised the need to devise efficient distributed algorithms which are applicable to both concept lattice digraph construction and formal concept generation in large formal contexts. A study by [7] introduced a method which is depend on the iterative docking of a given set of ligands to generate the training dataset. The ligand-based model is trained to predict the remaining ligands in order to exclude ligands predicted as 'low-scoring'. Upon docking of latter set of ligands, the model will be re-trained, and this is continued until a given level of efficiency is reached, after which the resting ligands are either docked or docked using this model. To make correct prediction periods for the ranking of the predicted ligands, the study employed SVM and conformal prediction while Apache Spark was implemented to parallelize the modeling and the docking.

A study by [8] proposed a parallel PCA combination with SVM (SP-PCA-SVM) based on Spark platform. To solve the problem of high computational time, high memory requirement, and low single detection efficiency of intrusion detection algorithms, this method used PCA for data training and data prediction before introducing a combined Bagging-SVM algorithm and its implementation on the Spark distributed framework. The results of the study showed the proposed method to reduce the training time for a large number of intrusion data to an extent and improved the learning efficiency of the model. The study reported in[9] highlighted the challenges of big data in terms of volume (growing from terabyte to petabyte) and the difficulty of building decision trees using larger dataset. The study also highlighted the inefficiency of data storage in the main memory which brings the need to move them to secondary memory, thereby increasing the computation cost. Having highlighted these problems, the study implemented the C4.5 which is the latest version of the DT algorithm based on the MapReduce model which is a good model for big data parallelism. The algorithm was evaluated on a big dataset of student alcohol consumption and the results showed the algorithm to save time and ensured scalability.

The study by [10] discussed the increasing use of big data concepts in Data Warehouses and the Intelligent of Business for producing better business insights, decisions, and fostering innovation. Hence, the study presented the best techniques for implementation a Big Data using a 3-legged Big Data environment strategy, as well as the problems of Big Data technologies adoption in enterprises. Hence, this study is relevant from both academic practitioner's perspectives

II. Machine Learning Algorithms

Machine learning (ML) is the ability of a computer of automatically learn on the hundreds of examples and experiences without being explicitly programmed. Depends on the given data it builds a logic using various algorithms. ML can be generally distributed into three classes: Supervised Learning, Unsupervised Learning, and Reinforcement Learning [11, 12]. In Supervised Learning, the algorithms can be utilized to foretell the values of output (Regression) or classifying the grade (Classification). There are two forms of supervised algorithms, Parametric and Non-Parametric. In Unsupervised Learning, based on unlabelled data the system work to find the unobserved pattern and significant structure In Reinforcement Learning, in this type the system must contact and make react with aroundable environment for decisions making and find the goals. By grouping returns to rating the prospective or unprospective attitude of the system. Examples include self-driving cars, self-cleaning vacuum cleaners, etc [13, 14]. SVM is a supervised machine-learning algorithm, which can be used for classification or regression problems. It transforms the data and based on these transformations it finds an optimal limit between the possible outputs this method is called the kernel trick [15].

III. Apache Spark

This is a publicly available cluster computing framework it became a top-level project in 2014 by Matei Zaharia and later donated to the Apache Spark Foundation. The framework was construct on HDFS [16]. It needs cluster manager and distributed storage system for completing its task which including the works on expanded processing of data, handing out data to dissociate worker nodes for processing. As a manager, a major node posting and schedules the dispenses tasks to the worker nodes. Its speed up in-memory data engine and developer-friendly API makes it the framework of choice. By comparing to Hadoop MapReduce reads and writes from disk, which goes slow the processing speed the Apache spark was progressed as faster

alternative which saved the data in memory and minimized the read/write cycle. This effects in running the applications 100x faster in memory and 10x faster on disk than Hadoop MapReduce [17].

IV. Research Method

SVM is a robust classification and regression framework which has seen several modifications in recent times. However, these models cannot be used directly for big data handling due to the increase in the memory and time required to train the SVM when the size of the training sample is large.

Furthermore, a single SVM cannot handle large-scale data sets effectively[24]. This has ushered in the practice of parallel SVM algorithm (splitting the task into fragments) to solve the problem of time and memory insufficiency for SVM[25].

The training and prediction of any given training sample X follow these steps. During the training stage, M samples are first extracted from the training set and denoted as TR (i =1,2,...,m); the PCA is used to analyze and extract relevant info from each sample set with the aim of reducing the data dimensionality. Then, the input attribute space will be converted into another attribute space in which the number gained attribute will be less than the number in the original feature space and still portray the original attributes of the most relevant information. Next, the PCA-handled data will be trained as a training set to generate the PCA SVM classifier.

The SVM algorithm was combined with a parallel programming model on the Spark platform to build the PCA-SVM model. The dataset was divided into M pieces by the Spark cluster before applying PCA data processing on the spark cluster and parallelly training SVM until the training is completed and the M pieces of the model are achieved. Each of the generated models was used to predict the tested dataset and their prediction results were later combined through a voting process as depicted in Fig. 1

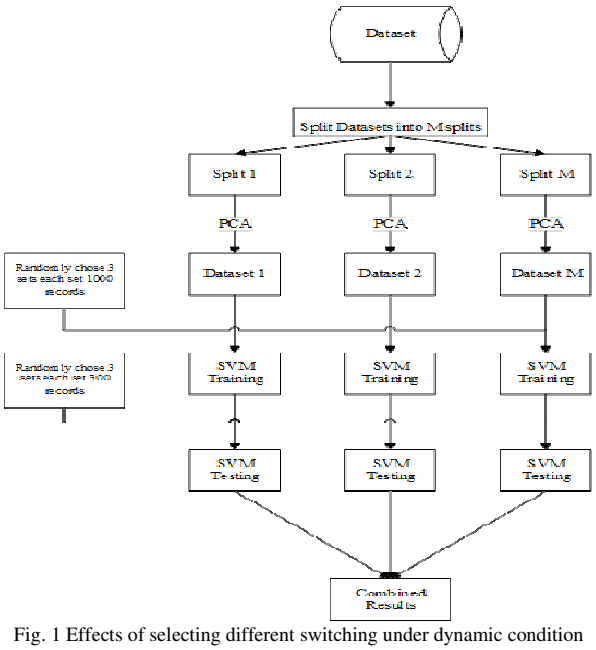

Fig. 1 Effects of selecting different switching under dynamic condition

V. Datasets

A. KDD Cup

The KDD'99 [15] which was first built by Stolfo et al. has been the commonly used data set for evaluating novel intrusion detection systems since 1999. This data was built using the data contained in DARPA'98 IDS evaluation program, a dataset of about 4GB consisting of compressed raw (binary) TCP dump data. KDD Cup is a yearly program regulated by ACM Special Interest Group on Knowledge Discovery and Data Mining. The ACM is the pioneer organization of data miners. This program ensures the availability of the yearly data archives, instructions, and winners for most years[23]. There are five million records in this dataset and each of these datasets contains 41 features used to classify malicious attacks into 4 groups (Probe, DoS, U2R or R2L). Being that KDD Cup '99 dataset was generated via simulation over a virtual network, it cannot reflect real traffic data [18]. Table 1 shows the details of KDDCUP dataset[4]

TABLE I
KDDCUP Dataset

Attack classes	22 types of attacks
Normal	
DoS	smurt, neptune, pod, teardrop, back, land
R2L	phf, ftp-write, imap, multihop, warezclient, warezmaster, spy, guess password
U2R	perl, loadmodule, buffer-overflow, rootkit
Probing	portsweep, ipsweep, satan, nmap

B. CICIDS2017 dataset

From the beginning of CICIDS2017 dataset it started enticing researchers for analysis and development of new models and algorithms [19]. Based on the author of CICIDS2017, the dataset spanned over eight different files, containing five days normal and attacks traffic data of Canadian Institute of Cybersecurity. We found the whole shape of a dataset that contains 3119345 instances and 79 features, containing 15 class labels (1 normal + 14 attack labels) [20][21]. Table 2 shows the details of CICIDS dataset[5]

TABLE III
CICIDS2017 Dataset

Attack classes	14 types of attacks
Benign (normal)	
DOS	DDoS, slowloris, Heratbleed, Hulk, GoldenEye, Slowhttptest
PortScan	PortScan
Bot	Bot
Brute-Force	FTP-Patator, SSH-Patator
Web attack	Web attack XSS, web attack SQL injection, web attack brute force
Infiltration	Infiltration

VI. Results And Discussions

This paper used PCA to reduce the size of the sample data, followed by SVM parallelization and scheme implementation on the Spark platform. Randomly three sets chosen for training each set size 1000 records, and three testing set randomly chosen each set size is 300 records.

From the results of the experiments, the classifier achieved good classification accuracy with a minimized computation time. To show the efficiency of our model, the results compared to single SVM and PCA-SVM as shown in Table3

TABLE III
EXPERIMENTAL RESULTS

Model	**Evaluation Metrics**	**KDDCUP99**	**CICIDS2017**
Single SVM	Accuracy	0.98	0.96
	Detection Rate	0.97	0.958
	Training Time	743	856
	Testing Time	15	18
PCA-SVM	FAR	0.08	0.108
	Accuracy	0.99	0.965
	Detection Rate	0.975	0.962
	Training Time	480	504
	Testing Time	10	12
Parallel PCA-SVM	FAR	0.06	0.094
	Accuracy	0.994	0.974
	Detection Rate	0.974	0.969
	Training Time	405	430
	Testing Time	4	6
	FAR	0.05	0.07

VII. CONCLUSIONS

Most real-time application nowadays generates a large volume of data (unstructured) which are meaningless unless preprocessed to extract useful information. Hence, there is a need to devise ways of analyzing these huge datasets to extract the useful information therein. In this work, we used parallel SVM to classify the dataset after dividing it into subset (m subset) before applying PCA to every subset in order to obtain a model. The results (model) of each subset were combined to get the final result (model). Two different datasets were used (KDDCUP and CICIDS 2017) and the developed model achieved 99.4% and 97.4% accuracy, respectively on these datasets. In the future study, this model will be applied to the real big dataset [22].

ACKNOWLEDGMENT

This work is supported by Ministry of Higher Education (MOHE) under Fundamental Research Grant Scheme (FRGS) reference code FRGS/1/2018/ICT04/UTHM/02/3

REFERENCES

[1] F. Yuan, F. Lian, X. Xu, and Z. Ji, "Decision tree algorithm optimization research based on MapReduce," in 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2015, pp. 1010-1013: IEEE.

[2] M. Bochicchio, A. Cuzzocrea, and L. Vaira, "A big data analytics framework for supporting multidimensional mining over big healthcare data," in 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, pp. 508-513: IEEE.

[3] G. Vaishali and V. Kalaivani, "Big data analysis for heart disease detection system using map reduce technique," in International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), 2016, pp. 1-6: IEEE.

[4] W. Fan, Z. Han, and R. Wang, "An evaluation model and benchmark for parallel computing frameworks," Mobile Information Systems, vol. 2018, 2018.

[5] S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on Big Data environment," Journal of Big Data, vol. 5, no. 1, p. 34, 2018.

[6] A. Chaudhary, V. Tiwari, and A. Kumar, "A novel intrusion detection system for ad hoc flooding attack using fuzzy logic in mobile ad hoc networks," in Recent Advances and Innovations in Engineering (ICRAIE), 2014, pp. 1-4: IEEE.

[7] M. K. Khaleel and M. A. Ismail, "Review on Intrusion Detection System Based on the Goal of the Detection System," 2018, vol. 10, no. 6, 2018-11-25 2018.

[8] H. Wang, Y. Xiao, and Y. Long, "Research of intrusion detection algorithm based on parallel SVM on spark," in 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC), 2017, pp. 153-156: IEEE.