Elastic net feature selected multivariate discriminant mapreduce classification

Arunadevi Nakkiran[1], Vidyaa Thulasiraman[2]
[1]Department of Computer Science, Periyar University, Salem, India
[2]Department of Computer Science, Govt Arts and Science College for Women, Bargur, India

Article Info

Article history:

Received Sep 4, 2021
Revised Feb 1, 2022
Accepted Feb 10, 2022

Keywords:

Big stream data
Elastic-net regularization
Feature selection
Kernelized multivariate fisher discriminant
MapR classifier
Prediction accuracy

ABSTRACT

Analyzing the big stream data and other valuable information is a significant task. Several conventional methods are designed to analyze the big stream data. But the scheduling accuracy and time complexity is a significant issue. To resolve, an elastic-net kernelized multivariate discriminant map reduce classification (EKMDMC) is introduced with the novelty of elastic-net regularization-based feature selection and kernelized multivariate fisher Discriminant MapReduce classifier. Initially, the EKMDMC technique executes the feature selection to improve the prediction accuracy using the Elastic-Net regularization method. Elastic-Net regularization method selects relevant features such as central processing unit (CPU) time, memory and bandwidth, energy based on regression function. After selecting relevant features, kernelized multivariate fisher discriminant mapr classifier is used to schedule the tasks to optimize the processing unit. Kernel function is used to find higher similarity of stream data tasks and mean of available classes. Experimental evaluation of proposed EKMDMC technique provides better performance in terms of resource aware predictive scheduling efficiency, false positive rate, scheduling time and memory consumption.

Corresponding Author:

Arunadevi Nakkiran
Department of Computer Science, Periyar University
Salem, Tamilnadu, India
Email: haseenaa@gmail.com

1. INTRODUCTION

In the era of big data, organizations have started to use big data stream computing as it has several advantages and risks from real-time big data. Big data stream computing hence has evolved as a mean in several applications including social networks, trading, video surveillance, and fraud identification and so on. Several research works have been incepted with both opportunities and challenges. Elastic online scheduling framework for big data streaming applications (E-Stream) by Sun *et al*. [1] with the objective of reducing the system response time and application fairness. But less focus was made on predictive scheduling accuracy. A novel predictive scheduling framework was designed by Li *et al*. [2] with the objective of ensuring fast and stream data processing. However, the time complexity in predictive scheduling remained unsolved. Given the significant nature of big data and big data analytics, critical analysis pertaining to big data challenges was presented by Sivarajah *et al*. [3]. According to Fernandes *et al.* [4] the finding of analysis from a metallurgic company was presented. Multivariate Gaussian function was used by Toit [5] to monitor critical variables. A fast and efficient distributed stream processing framework was presented by Choi *et al.* [6]. Bowden *et al.* [7] the design and prototype implementation of enabling predictive maintenance of industrial equipment was designed. Taxonomy, review, and future directions related to scheduling in distributed stream processing systems were designed Liu and Buyya [8]. A resource aware dynamic data stream model was designed by

Gautam *et al*. [9]. Yet another work by Usama *et al.* [10] focused on the problems and challenges in big data. To overcome the above issues such as higher predictive scheduling accuracy and minimum the time complexity, less false positive rate. In this paper, an efficient technique called elastic-net kernelized multivariate discriminant map reduce classification (EKMDMC) is introduced. The novel contributions of the proposed method include the following.

- For pre-processing, Elastic-net regularization is a regression method is introduced to accurately estimate the relationship between dependent and independent variables to avoid overfitting of model, according to the least absolute shrinkage and selection operator (LASSO) predictive model. Elastic-Net Regularization is a regression method applied to not only estimate the relationship among dependent and one or more independent variables, but also to avoid overfitting of model on training data.
- For classification, Elastic-net kernelized multivariate discriminant mapreduce classification is presented for reducing scheduling time by using minimum resources. Resource efficient processing unit prediction is performed via Kernelized Multivariate fisher discriminant mapreduce classifier (KMFDMC) with relevant features such as CPU time, Bandwidth Utilization, Memory Consumption and energy for scheduling tasks.
- The proposed technique is implemented in Python and tested with resource aware predictive scheduling efficiency, false positive rate, scheduling time and memory consumption for varying number of stream data.

The rest of paper is organized as shown in: A related work is presented in section 2. Design and implementation details of proposed technique are presented in section 3. A detailed discussion is presented in section 4 and concludes in section 4.

2. RESEARCH METHOD

According to Dehkordi and Zamanifar [11] a deadline aware scheduling framework was designed for minimizing latency and utilization cost. Yet another graphic processing units (GPU) enabled online stream data processing was designed by Chen *et al.* [12]. Modified first-fit based run time aware data stream scheduling strategy was designed by Sun *et al.* [13]. Dual channel pipeline parallel data processing model was designed in [14]. Yet another method based on double level hybrid genetic algorithm and ant colony optimization was presented by Xu *et al.* [15] to address dynamic simultaneous scheduling problem. A relatively novel intelligent model was designed in [16]. A review of complexity of managing bit data was presented in [17]. However, the scheduling process was not considered. To address this issue, by Gil *et al*. [18] a flexible resource-constrained project scheduling issue with competency differences was presented. A comprehensive approach based on novel deep learning models was presented in [19]. A novel priority-aware streaming media multi path data scheduler mechanism was designed in [20] for multimedia Multipathing services. Integrated support for similarity queries in a parallel Big Data management system was introduced in [21]. Haery based query system called Hadoop query (Haery) was developed in [22] to process the high dimensional data. Machine learning approach was introduced in [23]for reality awareness and optimization in cloud. Two decision tree classification methods were introduced in [24] for automatically find priority rules to solve the resource constrained project scheduling problem (RCPSP). A [25], [26] centralized 3-dimensional radio resources (namely, time, frequency, and power) allocation and scheduling approach for control-plane and [27] user-plane (C-/U-plane) separation architectures for fifth generation mobile networks.

3. RESULTS AND DISCUSSION

This big data stream refers to term used in representing huge amounts of data where continuous data stream is processed for extracting real-time insights. Such large voluminous data appears in different formats that cannot be processed with traditional methods. In this work, elastic-net kernelized multivariate discriminant MapReduce classification (EKMDMC) is presented to perform feature selection and resource aware predictive scheduling for big data stream.

Figure 1 shows architecture diagram of proposed EKMDMC technique. The input is obtained from the big dataset ‘D_b’. Consider ‘m’ number of processing units ‘p_1, p_2,p_3, … p_m’ that process ‘n’ number of data streams ‘sd$_1$, sd$_2$, sd$_3$, … sd$_n$’. Initially, Elastic-net regularization is applied to perform feature selection for selecting relevant features such as central processing unit (CPU), bandwidth, memory and energy. After selecting relevant features, resource efficient processing unit is determined by applying kernelized multivariate fisher discriminant MapReduce classifier. Finally, stream data task scheduling is carried out with higher accuracy. The different processes involved in design of the EKMDMC technique are described in the forthcoming sections.

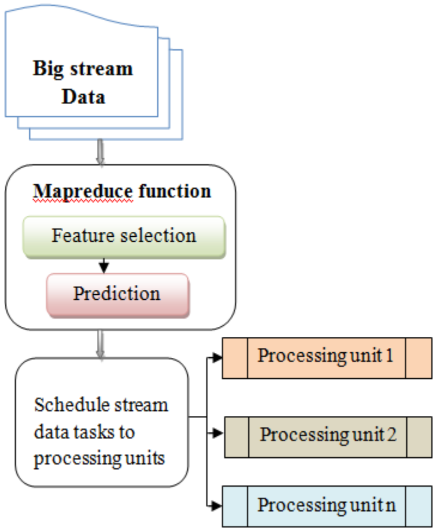

Figure 1. Architecture diagram of proposed EKMDMC technique

3.1. Elastic-net regularization based feature selection

The first process of Feature selection is performed here by applying Elastic-Net regularization. Elastic-Net regularization is a regression method applied to check the overfit present in the training data. EKMDMC technique uses the Elastic-Net regularization method to solve the issue based on LASSO Predictive model. Here, single variable from a group of highly relevant variables, and rejecting highly irrelevant variables. This is performed by adding regularization term to given equation. Besides, Elastic-Net Regularization is applied for both parameter estimation (i.e. prediction of average data processing time) and feature selection, where more relevant features are selected among group of features for performing predictive scheduling. The elastic-net method is defined as (1),

$$\rho = \arg\min(\|y - \alpha F\|^2 + \mathrm{P}_2\|\alpha\|^2 + \mathrm{P}_1\|\alpha\|_1) \tag{1}$$

From (1), 'F' representing a feature set '$\{F1, F2, F3, \ldots . Fn\}$' and regularization term of '$\|\alpha\|$', 'P1' and 'P2' parameters controlling the importance of regularization term with value between '0' and '1', apredicted output 'y' is determined using regression coefficient 'ρ'. The regression coefficient returns with zero for irrelevant features and one for relevant features. Using (1), relevant features such as central processing unit (CPU) time, memory and bandwidth, energy are selected for predicting average processing time of stream data. CPU time here, refers to time consumed in accomplishing task and as shown in (2).

$$t_{cpu} = t_{ct}(\,\mathrm{sd_i}) \tag{2}$$

From (2), the CPU time of processing unit 't_{cpu}' refers to stream data 'sdi' task completion time 'tct'. One of the main characteristics of processing unit is memory that refers to storage space utilized by processing unit to complete certain task. The memory utilization of processing unit is expressed as (3).

$$m_{ut} = m_t - m_{ud} \tag{3}$$

From (3), the memory utilization of processing unit 'mut' is the difference between total memory 'mt' and unused space of processing unit 'mud'. Besides CPU time and memory utilization, energy consumption of processing unit is considered for processing stream data. The energy consumption 'Ec' refers to difference between the total energy 'ET' and remaining energy 'ET' of processing unit and given as (4).

$$E_C = E_T - E_R \tag{4}$$

Finally, bandwidth utilization 'bwu' is average rate of data transfer speed of processing unit, that is difference between available bandwidth 'bwt' and unused bandwidth '$bwud$'. The bandwidth utilization of processing unit is computed as shown in (5).

$$bw_u = bw_t - bw_{ud} \quad (5)$$

The elastic-net regression method predicts the processing time of unit. The stream data tasks with lower CPU time and lower task size (i.e. memory) take less processing time. The elastic-net regression method increases the scheduling accuracy by selecting the relevant features of the processing unit.

3.2. Kernelized multivariate fisher discriminant mapreduce classifier

After selecting relevant features, resource efficient processing unit prediction is performed via kernelized multivariate fisher discriminant mapreduce classifier (KMFDMC) with relevant features. MapReduce function includes two phases namely map phase and reduce phase. Here, the streams data are mapped to appropriate processing unit using kernelized multivariate fisher discriminant with relevant features. Next, a summary operation is carried out by providing final output results. Figure 2 illustrates KMFDMC for efficient prediction as well as resource aware scheduling. Let us consider a number of stream data tasks ‘sd1, sd2, sd3, … sdn’ as input. Initially, a number of classes (i.e. processing unit) ‘ p1, p2, p3, … . pn’ are initialized.

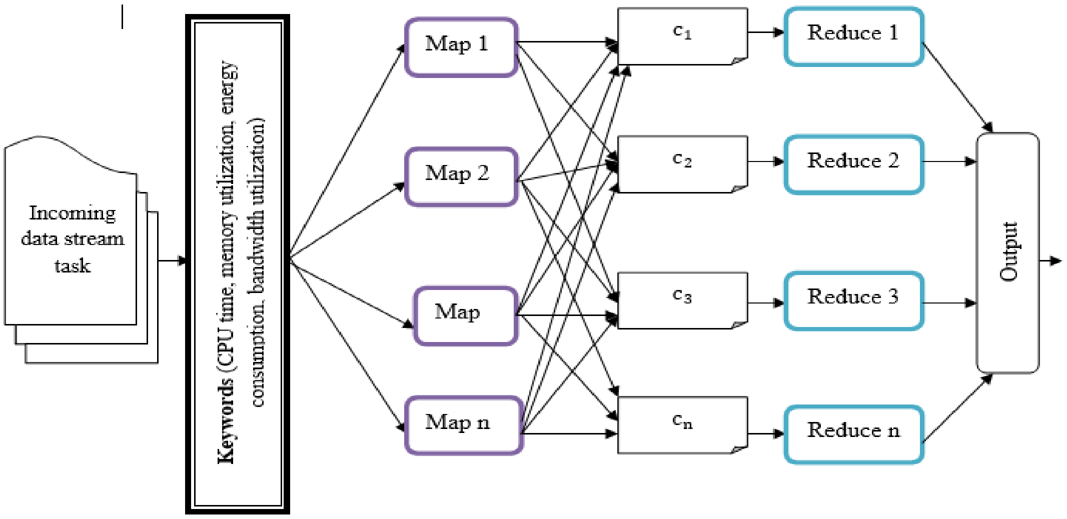
Incoming data stream task
Keywords (CPU time, memory utilization, energy consumption, bandwidth utilization)
Map 1
Map 2
Map
Map n
c1
c2
c3
cn
Reduce 1
Reduce 2
Reduce 3
Reduce n
Output

Figure 2. Flow process of kernelized multivariate fisher discriminant mapreduce classifier (KMFDMC)

Map Phase takes an input and transforms the content into Key-Value pair in which the key forms distinctive keywords combination. Based on combination of keywords, the Kernelized Multivariate Fisher Discriminant predicts the processing unit and performs scheduling process. The KMFDMC uses discriminant vector maps to map different incoming stream data into different classes. Fisher defined as the separation function which is ratio of variance between classes to variance within class is defined as shown in (6).

$$s_n = \frac{\sigma_b}{\sigma_w} = \frac{ws_r\,(b)d}{ws_b(b)d} \quad (6)$$

From (6), the separation function ‘s_n’ refers to ratio of variance between class ‘σ_b’ and variance within the class ‘$\frac{\sigma_b}{\sigma_w}$’. This is obtained using a linear discriminant vector ‘w’ into class based on optimal projection direction ‘d’ with aid of scatter matrix between ‘$s_r\ (b)$’ and within ‘$s_r\ (w)$’classes. Scatter matrix is applied to find whether the processing unit is suitable to handle stream data. Besides scatter matrix, a mean value is initialized for each class. In KMFDMC, a kernel function finds the similarity between the mean of the class and stream data tasks. Here the distance similarity is measured to find the resource efficient processing unit.

$$k(\text{sdi}, \mu j) = \|\text{sdi} - \mu j\|2 \quad (7)$$

From (7), distance similarity is obtained via kernel function 'k(sdi, μj)'. With resultant value, fisher discriminant analysis identifies the minimum distance between the stream data task 'sdi' and mean of classes 'μj' (i.e. processing unit).

$$f(x) = \arg\min \|\mathrm{sd_i} - \mu_j\|^2 \tag{8}$$

From (8), $f(x)$ denotes an output of fisher discriminant analysis, arg min denotes an argument of minimum function. The minimum distance represents higher similarity between mean and stream data task. It means that specific processing unit is suitable for completing certain stream data task with less resource utilization. Fisher discriminant analysis predicts efficient processing unit for all incoming tasks. After predicting resource efficient processing unit, the stream data tasks are scheduled with corresponding unit.

Algorithm 1 Elastic-Net Kernelized Multivariate Discriminant Map Reduce Classification

```
Input: Number of stream data task $\mathrm{sd_1,sd_2,sd_3,\ldots sd_n}$, processing unit $(p_1,p_2,p_3,\ldots.. p_n)$
Output: Improve resource aware predictive scheduling efficiency
Begin
\\ feature selection
    1. Apply regression $\rho$ to select the features Central Processing Unit (CPU)
       time, memory, bandwidth, energy
    2.  For each processing unit $p_i$
    3.  Calculate $t_{cpu}$, $m_{ut}$, $E_c$, $bw_u$
    4. End for
\\prediction and scheduling
    5.  Initialize number of classes $c_j$
    6. Define class separability function $s_n$
    7.  Define the mean of the class $\mu_j$
    8. For each stream data task $\mathrm{sd_i}$
    9. For each mean of the class $\mu_j$
    10. Measure similarity $k(\mathrm{sd_i},\mu_j)$
    11. Find minimum distance $\arg\min\|\mathrm{sd_i}-\mu_j\|^2$
    12. Predict resource efficient $p_i$
    13. Schedule $\mathrm{sd}_i$ to $p_i$
    14. End for
    15. End for
 End
```

Algorithm 1 describes the Elastic-Net kernelized multivariate discriminant MapReduce classification (EKMDMC) to improve scheduling efficiency by utilizing minimum resources.

4. FIGURES AND TABLES

The experimental evaluation is performed with epileptic seizure recognition dataset https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition. Epileptic seizure recognition dataset is high dimensional dataset includes 11500 instances and 179 attributes. Comparative analysis and discussion is made four different parameters, resource aware predictive scheduling efficiency, false positive rate, scheduling time and memory consumption with number of stream data tasks. Associated tasks are classification and clustering. The dataset characteristics are multivariate and time series. Experimental configuration as shown in Table 1.

Table 1. Tabulation for experimental configuration

Requirements	Specification
Software	Python 3.5
Processor	Intel i3-4130 3.40GHz
RAM	2 GB and above
Operating System	Windows 7, 10

4.1. Impact of resource aware predictive scheduling efficiency

Resource aware predictive scheduling efficiency measures percentage ratio of number of tasks correctly scheduled to resource aware optimized processing unit '$RAOPU_i$ ' to number of data task 'N'. As shown in (9).

$$RPSE = \frac{RAOPU_i}{N} * 100 \qquad (9)$$

From (9), resource aware predictive scheduling efficiency ‘N’ is measured in percentage. Table 2 illustrates the convergence graph of resource aware scheduling efficiency using three different methods, EKMDMC, E-Stream [1] and predictive scheduling framework [2]. With the increase in thenumber of data tasks from 100 to 1000, the convergence graph shows a decreasing trend and then increasing trend is found. Hence, the graph of resource aware scheduling efficiency is neither inversely nor directly proportional to number of tasks. With 100 number of data tasks considered for experimentation. Resource aware predictive scheduling efficiency using EKMDMC, E-Stream [1], and predictive scheduling framework [2] was ‘83.71%’, ‘77.57%’and ‘70.42%’.

Table 2. Tabulation for resource aware scheduling efficiency

Number of Data task	Resource aware scheduling efficiency (%)			
	EKMDMC	E-Stream	Predictive Schedulingframework	3D radio
100	83.71	77.57	70.42	63.38
200	79.25	76.35	67.55	56.75
300	78.55	74.25	63.35	52.45
400	76.35	71.15	61.15	51.15
500	74.25	67.55	64.55	61.55
600	77.15	64.35	60.35	56.35
700	78.35	67.25	63.33	59.41
800	80.45	69.35	61.55	53.75
900	78.15	71.15	64.55	57.95
1000	77.55	69.45	67.35	65.25

Figure 3 shows the comparison of impact of resource aware predictive scheduling efficiency. The resource aware predictive scheduling efficiency using EKMDMC was improved due to the application of Elastic-Net kernelized multivariate discriminant MapReduce classification algorithm. This improved the resource aware scheduling efficiency using EKMDMC by 10% compared to [1], 21% compared to [2] and 36% compared to [25], [26].

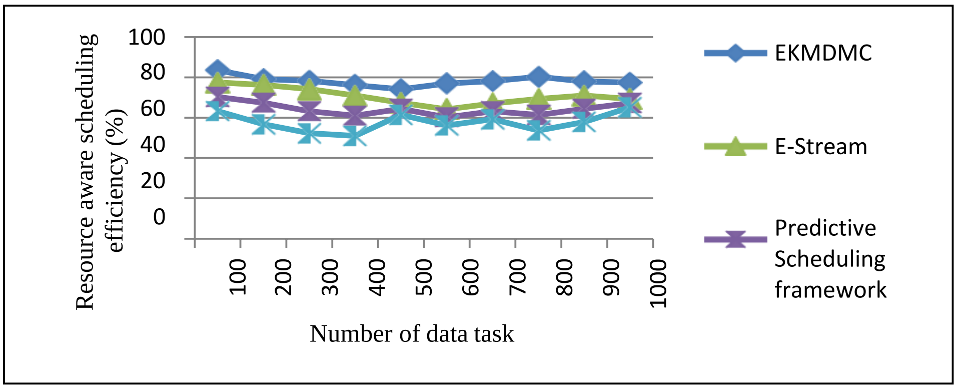

Figure 3. Comparison of resource aware predictive scheduling efficiency

4.2. Impact of false positive rate

False positive rate refers to the percentage ratio of number of tasks incorrectly scheduled to resource aware processing unit ‘$RAOPU_{Incorrect}$’ to number of data task ‘N’, as shown in (10).

$$FPR = \frac{RAOPU_{Incorrect}}{N} * 100 \qquad (10)$$

From (10), false positive rate (FPR) is measured in percentage (%). Lower false positive rate ensures the efficiency of the method. Table 3 illustrates the convergence graph of resource aware scheduling efficiency using three different methods, EKMDMC, E-Stream [1] and predictive scheduling framework

[2]. With the increase in the number of data tasks from 100 to 1000, the convergence graph shows a decreasing trend and then increasing trend is found. FPR using EKMDMC, E-Stream [1], and predictive scheduling framework [2] was ’13.28%’, ‘20.42’ and ‘27.57’.

Table 3. Tabulation for false positive rate

Number of Data Tasks	False positive rate (%) EKMDMC	E-Stream	Predictive Scheduling framework	3D radio
100	13.28	20.42	27.57	34.72
200	14.35	22.25	28.55	34.85
300	16.2	24.55	30.35	36.15
400	18.35	28.15	32.25	36.35
500	20.45	31.35	34.55	37.75
600	22.55	34.55	38.15	42.25
700	29.25	36.25	40.25	44.25
800	32.15	38.35	42.25	46.15
900	34.44	40.25	43.55	47.85
1000	38.25	42.35	44.55	48.75

Figure 4 shows the false positive rate with 1000 different numbers of tasks. With the increase in number of data tasks, the FPR is found to be in the increasing trend. The FPR using EKMDMC, E-Stream [1] and Predictive Scheduling framework [2] was ’13.28%’, ‘20.42’ and ‘27.57’. From that, the false positive rate was reduced in EKMDMC. This is because of application of Kernelized Multivariate Fishe Discriminant. The FPR using EKMDMC is reduced by 26% compared to [1] 35% compared to [2] and 43% when compared to [25], [26].

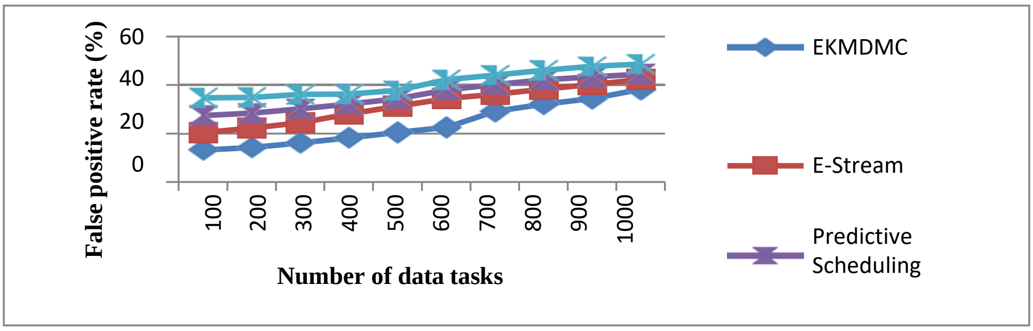

Figure 4. Graphical representations for false positive rate

Impact of scheduling time scheduling time refers to time consumed in scheduling resource aware for single data. It is formulated as shown in (11).

$$ST = N * Time[RAPS] \quad (11)$$

From (11), scheduling time ‘ST’ is measured based on number of data tasks given as input ‘N’ and time consumed in scheduling data tasks ‘$Time[RAPS]$’ for single data in a resource aware manner. It is measured in milliseconds (ms). Table 4 shows the convergence graph of scheduling time measured for 1000 different numbers of data tasks. With increase in the number of data tasks, the time consumed in scheduling also increases due to the increase in the size of data tasks. Therefore, the overall scheduling time using EKMDMC, E-Stream [1] and Predictive Scheduling framework [2] were observed to be ‘0.148ms’, ‘0.242ms’ and ‘0.27ms’.

Figure 5 show the graphical representation for scheduling time. However, with the sample of ‘100’ number of data tasks. Therefore, the overall scheduling time using EKMDMC, E-Stream [1] and Predictive Scheduling framework [2] were observed to be ‘0.148ms’, ‘0.242ms’ and ‘0.27ms’. From this analysis, the scheduling time using EKMDMC were lesser than [1], [2]. This is because of reason that with the application of Elastic-net Regularization. The scheduling time using EKMDMC was lesser than 30%, 47%, 52% compared to [1], [2], [25], [26].

Table 4. Tabulation for scheduling time

Number of Data Tasks	Scheduling time (ms)			
	:KMDMC	E-Stream	ctive Scheduling framework	3D Radio
100	0.148	0.242	0.27	0.35
200	0.185	0.255	0.345	0.435
300	0.203	0.305	0.405	0.505
400	0.275	0.325	0.475	0.525
500	0.305	0.375	0.515	0.545
600	0.31	0.405	0.575	0.605
700	0.318	0.465	0.625	0.658
800	0.325	0.525	0.676	0.694
900	0.341	0.574	0.705	0.715
1000	0.375	0.604	0.725	0.760

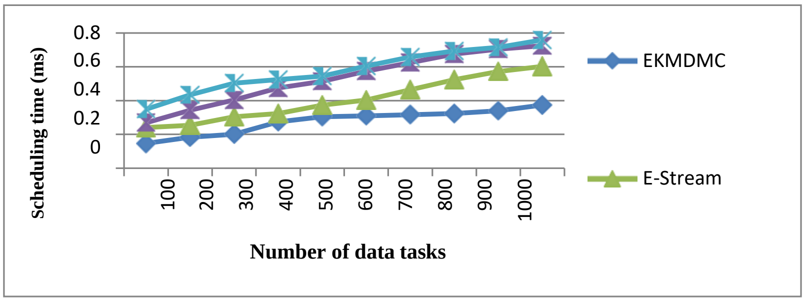

Figure 5. Graphical representations for scheduling time

4.3. Impact of memory consumption

Memory consumption refers to memory consumed in scheduling resource aware for single data. As shown in (12). From (12), memory consumption 'MC' is measured based on number of data tasks given as input 'N' and memory consumed in scheduling data tasks '$Space[RAPS]$' for single data in resource aware manner. It is measured in kilobytes (KB). Table 5 shows the convergence graph of scheduling time measured for 1000 different numbers of data tasks. With increase in the number of data tasks, the time consumed in scheduling also increases due to the increase in the size of data tasks. Hence, from the table it is inferred that the scheduling time is directly proportional to number of data tasks. However, with the sample of '100' number of data tasks. Therefore, the overall scheduling time using EKMDMC, E-Stream [1] and Predictive Scheduling framework [2] were observed to be '69KB', '97 KB' and '124 KB'.

$$MC = N * Space[RAPS] \tag{12}$$

Table 5. Tabulation for memory consumption

Number of Data Tasks	Memory consumption (KB)			
	EKMDMC	E-Stream	Predictive Scheduling framework	3D radio
100	69	97	124	151
200	110	133	164	178
300	113	154	188	203
400	117	183	214	220
500	122	194	223	29
600	130	213	239	27
700	143	224	254	29
800	154	233	284	26
900	173	254	288	310
1000	188	288	319	328

Figure 6 shows the results of memory consumed in scheduling data. From the figure, it is illustrative that the memory consumption is reduced using EKMDMC technique as compared to [1] and [2]. This is because of application of regression method. Therefore, memory consumed in resource aware predictive scheduling using EKMDMC technique is reduced by 32% compared to [1], 42% compared to [2] and 45% compared to [25], [26].

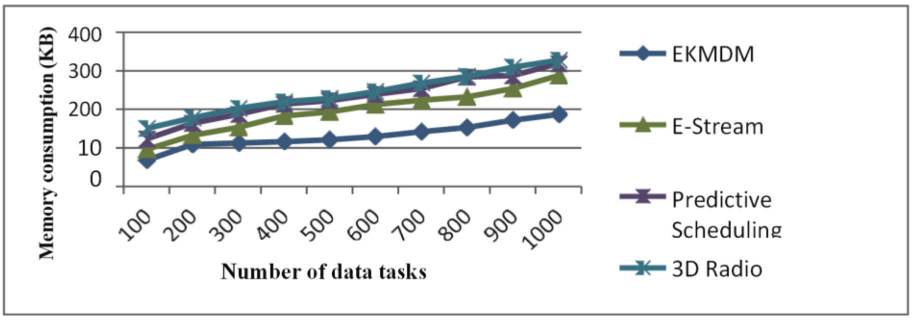

Figure 6. Graphical representations for memory consumption

5. CONCLUSION

This paper presents elastic-net kernelized multivariate discriminate map reduce classification (EKMDMC) technique, which is a resource aware predictive scheduler for big stream data. It selects more relevant feature using KMFDMC to perform resource aware predictive scheduling and ensure that incoming stream data tasks are scheduled. EKMDMC technique reduces the false positive rate by utilizing the Fisher discriminant analysis. Simulation results show that EKMDMC technique provides better performance in terms of false positive rate, scheduling time, Memory consumption and resource aware predictive scheduling efficiency. However, EKMDMC technique considers only limited number of stream data task. In future work, number of data task further increased to evaluate performance of EKMDMC technique. Hence, future work of EKMDMC technique can be proceeded to solve multi-mode resource constrained project scheduling problem (MRCPSP).

REFERENCES

[1] D. Sun, H. Yan, S. Gao, X. Liu, and R. Buyya, "Rethinking elastic online scheduling of big data streaming applications over high-velocity continuous data streams," *Journal of Supercomputing*, vol. 74, no. 2, pp. 615–636, Feb. 2018, doi: 10.1007/s11227-017-2151-2.

[2] T. Li, J. Tang, and J. Xu, "Performance modeling and predictive scheduling for distributed stream data processing," *IEEE Transactions on Big Data*, vol. 2, no. 4, pp. 353–364, Dec. 2016, doi: 10.1109/tbdata.2016.2616148.

[3] U. Sivarajah, M. M. Kamal, Z. Irani, and V. Weerakkody, "Critical analysis of Big Data challenges and analytical methods," *Journal of Business Research*, vol. 70, pp. 263–286, Jan. 2017, doi: 10.1016/j.jbusres.2016.08.001.

[4] M. Fernandes, A. Canito, V. Bolón-Canedo, L. Conceição, I. Praça, and G. Marreiros, "Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry," *International Journal of Information Management*, vol. 46, pp. 252–262, Jun. 2019, doi: 10.1016/j.ijinfomgt.2018.10.006.

[5] J. Du Toit, "Enabling predictive maintenance using semi-supervised learning with Reg-D transformer data," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, no. 3, pp. 6111–6116, 2014, doi: 10.3182/20140824-6-za-1003.00201.

[6] J. H. Choi, J. Park, H. D. Park, and O. G. Min, "DART: Fast and efficient distributed stream processing framework for internet of things," *ETRI Journal*, vol. 39, no. 2, pp. 202–212, Apr. 2017, doi: 10.4218/etrij.17.2816.0109.

[7] D. Bowden *et al.*, "A cloud-to-edge architecture for predictive analytics," in *CEUR Workshop Proceedings*, 2019, vol. 2322.

[8] X. Liu and R. Buyya, "Resource management and scheduling in distributed stream processing systems: A taxonomy, review, and future directions," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–41, May 2020, doi: 10.1145/3355399.

[9] B. Gautam and A. Basava, "Performance prediction of data streams on high-performance architecture," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, p. 2, Dec. 2019, doi: 10.1186/s13673-018-0163-4.

[10] M. Usama, M. Liu, and M. Chen, "Job schedulers for Big data processing in Hadoop environment: testing real-life schedulers using benchmark programs," *Digital Communications and Networks*, vol. 3, no. 4, pp. 260–273, Nov. 2017, doi: 10.1016/j.dcan.2017.07.008.

[11] M. Mortazavi-Dehkordi and K. Zamanifar, "Efficient deadline-aware scheduling for the analysis of Big Data streams in public cloud," *Cluster Computing*, vol. 23, no. 1, pp. 241–263, Mar. 2020, doi: 10.1007/s10586-019-02908-2.

[12] Z. Chen, J. Xu, J. Tang, K. A. Kwiat, C. A. Kamhoua, and C. Wang, "GPU-accelerated high-throughput online stream data processing," *IEEE Transactions on Big Data*, vol. 4, no. 2, pp. 191–202, Jun. 2016, doi: 10.1109/tbdata.2016.2616116.

[13] D. Sun, S. Gao, X. Liu, F. Li, X. Zheng, and R. Buyya, "State and runtime-aware scheduling in elastic stream computing systems," *Future Generation Computer Systems*, vol. 97, pp. 194–209, Aug. 2019, doi: 10.1016/j.future.2019.02.053.

[14] L. Li, F. Li, G. Shi, and K. Geng, "An efficient stream data processing model for multiuser cryptographic service," *Journal of Electrical and Computer Engineering*, vol. 2018, pp. 1–10, Jul. 2018, doi: 10.1155/2018/3917827.

[15] W. Xu, S. Guo, X. Li, C. Guo, R. Wu, and Z. Peng, "A dynamic scheduling method for logistics tasks oriented to intelligent manufacturing workshop," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–18, Apr. 2019, doi: 10.1155/2019/7237459.

[16] K. R. K. Kamoona and C. Budayan, "Implementation of genetic algorithm integrated with the deep neural network for estimating at completion simulation," *Advances in Civil Engineering*, vol. 2019, pp. 1–15, May 2019, doi: 10.1155/2019/7081073.

[17] D. Gil, M. Johnsson, H. Mora, and J. Szymański, "Review of the complexity of managing big data of the internet of things," *Complexity*, vol. 2019, pp. 1–12, Feb. 2019, doi: 10.1155/2019/4592902.

[18] J. Chen, S. Tong, H. Xie, Y. Nie, and J. Zhang, "Model and algorithm for human resource-constrained R&D program scheduling optimization," *Discrete Dynamics in Nature and Society*, vol. 2019, pp. 1–13, Apr. 2019, doi: 10.1155/2019/2320632.

[19] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. M. Altowaijri, “Rapid transit systems: smarter urban planning using big data, in-memory computing, deep learning, and GPUs,” *Sustainability*, vol. 11, no. 10, p. 2736, May 2019, doi: 10.3390/su11102736.

[20] W. Lu, D. Yu, M. Huang, and B. Guo, “PO-MPTCP: Priorities-oriented data scheduler for multimedia multipathing services,” *International Journal of Digital Multimedia Broadcasting*, vol. 2018, pp. 1–9, Dec. 2018, doi: 10.1155/2018/1413026.

[21] T. Kim *et al.*, “Similarity query support in big data management systems,” *Information Systems*, vol. 88, p. 101455, Feb. 2020, doi: 10.1016/j.is.2019.101455.

[22] J. Song, H. He, R. Thomas, Y. Bao, and G. Yu, “Haery: A hadoop based query system on accumulative and high-dimensional data model for Big Data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 7, pp. 1362–1377, Jul. 2020, doi: 10.1109/TKDE.2019.2904056.

[23] L. Zhen, Y. Liu, W. Dongsheng, and Z. Wei, “Parameter estimation of software reliability model and prediction based on hybrid wolf pack algorithm and particle swarm optimization,” *IEEE Access*, vol. 8, pp. 29354–29369, 2020, doi: 10.1109/ACCESS.2020.2972826.

[24] P. Roy, G. S. Mahapatra, and K. N. Dey, “Forecasting of software reliability using neighborhood fuzzy particle swarm optimization based novel neural network,” *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1365–1383, Nov. 2019, doi: 10.1109/JAS.2019.1911753.

[25] N. Gupta, A. Sharma, and M. K. Pachariya, “Multi-objective test suite optimization for detection and localization of software faults,” *Journal of King Saud University - Computer and Information Sciences*, Jan. 2020, doi: 10.1016/j.jksuci.2020.01.009.

[26] S. A. Sari and K. M. Mohamad, “Recent research in finding the optimal path by ant colony optimization,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 1015–1023, Apr. 2021, doi: 10.11591/eei.v10i2.2690.

[27] M. A. A. K. Alabajee, N. A. AL-Saati, and T. R. Alreffaee, “Parameter tuning of software effort estimation models using antlion optimization,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 817–828, Jun. 2021, doi: 10.12928/TELKOMNIKA.v19i3.16907.

BIOGRAPHIES OF AUTHORS

Arunadevi Nakkiran iD 8 SC P rreceived MCA and MPhil degrees in Computer Science from Bharathidasan University (2000) and Periyar University (2004) respectively. She is working as a Solution Architect in DXC Technology and also a Research Scholar in Periyar University, Salem. She is working in Microsoft Technologies. Her research include data streaming in Big Data, Artificial Intelligence. Her passionate towards multi-skill always helps in win-win situation. Her passionate interests include the applications of artificial intelligence, evolutionary and heuristic optimization techniques in stream data. She can be contacted at email: haseenaa@gmail.com.

Dr. Vidyaa Thulasiraman iD 8 SC P received MCA. in University of Madras and M.Phil., Ph.D., in Computer Science Mother Teresa Women's University, Kodaikanal. She has almost 23 years of teaching experience and currently serving as Associate Professor and HOD in Government Arts and Science College for Women, Bargur. She has done her research in the field of integrated approach to Network Security Management. She is also in the field of research from 1999 and her research interests include Artificial Intelligence and Data Mining. She has contributed for 30 journals and guiding almost 13 MPhil and 7 PhD candidates. She can be contacted at email: vidyaathulasi@gmail.com.