Azerbaijan Journal of High Performance Computing, Vol 4, Issue 1, 2021, pp. 3-14
https://doi.org/10.32010/26166127.2021.4.1.3.14

AzJHPC
Azerbaijan Journal of High Performance Computing

*Correspondence:
Zdzislaw Polkowski, WSG University, Bydgoszcz, Poland, zdzislaw.polkowski@byd.pl

Provisioning Large-Scaled Data with Parameterized Query Plans: A Case Study

Zdzislaw Polkowski[1], Sambit Kumar Mishra[2]

[1]WSG University, Bydgoszcz, Poland, zdzislaw.polkowski@byd.pl
[2]Gandhi Institute for Education and Technology, Baniatangi, Bhubaneswar, affiliated to Biju Patnaik University of Technology, Rourkela, Odisha, India, sambitmishra@gietbbsr.com

Abstract

In a general scenario, the approaches linked to the innovation of large-scaled data seem ordinary; the informational measures of such aspects can differ based on the applications as these are associated with different attributes that may support high data volumes high data quality. Accordingly, the challenges can be identified with an assurance of high-level protection and data transformation with enhanced operation quality. Based on large-scale data applications in different virtual servers, it is clear that the information can be measured by enlisting the sources linked to sensors networked and provisioned by the analysts. Therefore, it is very much essential to track the relevance and issues with enormous information. While aiming towards knowledge extraction, applying large-scaled data may involve the analytical aspects to predict future events. Accordingly, the soft computing approach can be implemented in such cases to carry out the analysis. During the analysis of large-scale data, it is essential to abide by the rules associated with security measures because preserving sensitive information is the biggest challenge while dealing with large-scale data. As high risk is observed in such data analysis, security measures can be enhanced by having provisioned with authentication and authorization. Indeed, the major obstacles linked to the techniques while analyzing the data are prohibited during security and scalability. The integral methods towards application on data possess a better impact on scalability. It is observed that the faster scaling factor of data on the processor embeds some processing elements to the system. Therefore, it is required to address the challenges linked to processors correlating with process visualization and scalability.

Keyword: Scalability, Meta-heuristic, Gradient Value, Supervised learning, Parameterized query

1. Introduction

It can now be seen that the terms Big Data and Large Scale Data have become commonplace. Big Data is an expression that is becoming more and more popular all

over the world. Mainly analysts use them in their work, but they also arouse interest among ordinary people. As a work tool, it is a source of many useful data and information, and in society, it causes reluctance and fear of excessive surveillance by corporations using this technology. Big Data describes the tendency to search for, retrieve, collect and process available data. It is a method of gathering information from various sources and then analyzing and using it for your purposes. Therefore, an essential thing in Big Data is processing information and the practical use of conclusions drawn from it and not the mere collection of data. Big Data is a term used for such data sets that are simultaneously characterized by the large volume, diversity, real-time streaming inflow, volatility, complexity, as well as requiring the use of innovative technologies, tools and information methods in order to extract new and useful knowledge from them. See figure 1

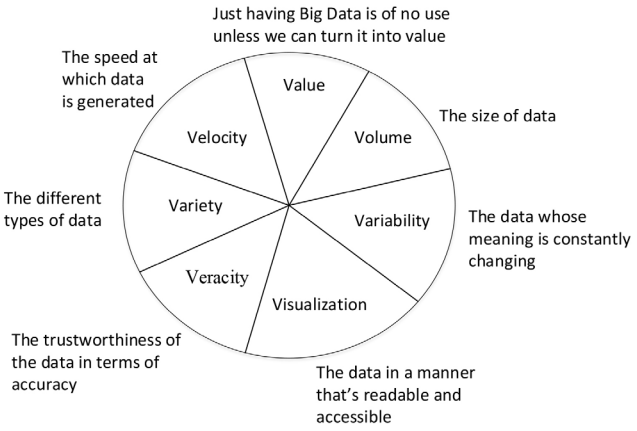

Fig. 1. Current Status and Future Prospect of IoT

Although Large Scale Data is not a standard term, it can be associated with data that grows to enormous sizes over time and is stored in conventional data warehouse solutions. Note that both are large, but the need for all data at one time (a single sample space for analysis) makes the difference to the solution needed to process them. The initial step in such a situation can be associated with a Hadoop solution, and subsequently, it can be linked with a Netezza, Teradata, or Oracle store.

While being associated with storage and the processing of large-scale heterogeneous data, the complete database system and Hadoop can have a better solution. As in the practical situation, many difficulties have been observed during large-scale heterogeneous data with conventional database operations by the researchers. Hadoop has been proved as most appropriate towards processing large-scaled heterogeneous data with much more enhanced performance.

Sometimes, peculiar challenges have been observed during the analysis of large-scale heterogeneous data due to the diversification of data sets, particularly mining data sets. The reason is that the current techniques may not always adhere to adequate time while associated with high dimensional data. Of course, some techniques have

tried to make it possible by accumulating largely sized semi-structured and unstructured data in a specific time frame. Now the main challenge is to analyze these data with observation to obtain better knowledge. Also, it is required to give more attention towards designing storage systems provisioning the guarantees on the outcomes. It is observed that while the size of large-scaled data increases, it can be accessible to the concerned data frameworks following the interpersonal communication strategies. Sometimes the data frameworks are not efficiently proportioned to changes or enhancement. Also, the instant online applications associated with similar data sets are consistently dynamic. In such situations, it can be complicated to focus on distinctive configurations and complete data frameworks. Accordingly, provisioning large-scale data processing in a virtual platform is also equally important because of the highly efficient services linked to databases. Focusing on the evaluation and regeneration of query plans, there should be the provision to execute the query plans within the traditional query-based system initially and a virtual database system. It is understood that the functionalities associated with the virtual database system can store the data and partition automatically with the proper distribution. For example, while a query is being processed, the system should accept the query in structured form and generate the optimized query execution plans. After that, the query plans can be contained by identifying the associated network and deployed towards the virtual storage with optimal feasibility. The intermediate results can be pipelined and monitored based on the scheduling criteria.

2. Review of literature

As mentioned in (Than, 2015), to accumulate the large-scaled data that are exploding in an unprecedented manner, the technologies linked to virtualization, sensing devices, along social media applications should be enhanced. As per research, estimation shows that since the year-2020, more than 30 billion devices have been connected. So these large-scaled data possess remarkable potential in terms of business values in different sectors. See figure 2. below:

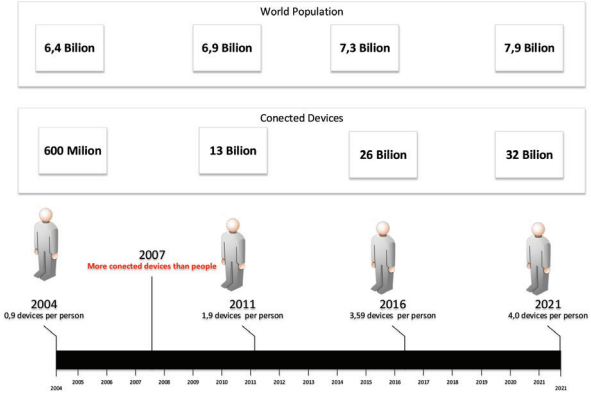

Fig. 2. Current Status and Future Prospect of IoT

In (Raghupathi, W., & Raghupathi, V., 2014), authors have focused on the potenti-

ality of large-scale data linked to business values in different sectors, mainly hospitals, transportation, energy management, and financial services. In work, they presented figure 3.

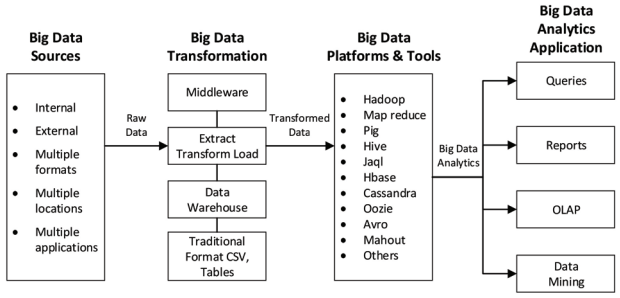

Fig. 3. An applied conceptual architecture of big data analytics

Qiu et al. (2016) in their study prioritized machine learning application on large-scale data, specifically in the application of signal processing. They have identified typical issues, i.e., Types of data, high speed of data, and data with low values. They have also applied different learning techniques and tried to identify the fundamental problems to obtain the desired solution. See figure 4.

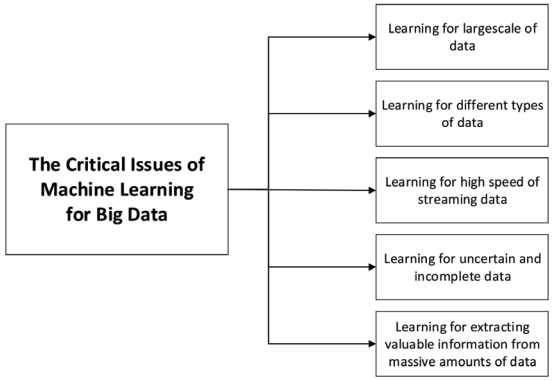

Fig. 4 The critical issues of machine learning for big data

Research by Xue-wen et al. (2014) has focused on the difficulties associated with deep learning. Also, they focused on the different aspects of machine learning tried to solve the challenges linked to large-scale data.

Singh et al. (2015) in their work concentrated on horizontal as well as vertical scaling platforms. Their study has also focused on the advantages and disadvantages of various platforms with scalability, I/O performance, fault tolerance, real-time processing, and iterative task support.

Tang et al. (2016) in their work prioritized learning accuracy as well as computation time. They tried to minimize the computation time as well as enhance the learning ac-

curacy. Their work concentrated on the Reservoir sampling method to split the large data sets into smaller datasets.

Various machine learning techniques and their application in smaller sets have been discussed in (Gruber et al., 2015). They focused on inverse probability weights to obtain the parameters and observed that the suitable applications linked with learning mechanisms could be more accurate on large-scaled data along with minimization of computation time.

Yang et al. (2015) in their work focused on automatic transfer learning techniques associated with large-scale data. They tried to enhance the supervised learning mechanism by enabling automatic knowledge transfer from the source domains to the target domain.

Research by Bachmann et al. (2017) has concentrated on collaborative filtering mechanisms that are quite sufficient to focus on the relevant online data or information to users based on their specified criteria. They have also observed that different transactions accumulated regularly may be verified as a continual process to enhance learning performance.

Research by Wu et al. (2016) prioritized the translation of machine instruction as the computational measures linked to machine translation systems are quite sophisticated and expensive, to manage such a situation with the large-scaled data, they also prefer deep learning concepts.

Saeedi et al. (2016) in their research focused on the development of sensor equipped wearable devices. In fact, they have also prioritized automated reconfigured wearable systems to observe the changes in activity recognition. The main constraint in such a situation can be the heterogeneity of the data, as the sensor data can be associated with different heterogeneous signals.

In their research Tarasyev et al. (2021) focused on analytical concepts of data based on sources of official statistics. In fact, they have considered different strategies towards developing the Kurgan region and enhancing economic conditions.

3. Provisioning the data in the virtual storage system

The concerned layer in the virtual system usually supports the peripherals along with the scheduling mechanisms. Of course, there should be the provision of extreme data availability along with consistent security measures that can be accessed publicly in the virtual platform. In fact, there should be the facilitation of the high valued data with proper optimization criteria maintaining the atomicity, consistency, isolation, and durability. Sometimes the higher-level abstraction during managing data can focus on an independent evaluation of query or more dependent on query linked to virtual database and storage. In such a situation, there should be the provision to execute intensive queries deploying the query execution plans and implementing a specified approach in the system, as mentioned in figure 5.

4. Analysis and refinement of data

In common application, the large-scaled data in social aspects are associated with monitoring, predicting, and mutual communication. Therefore, it is required to analyze and filter data with proper prediction and requisite findings in such a situation. Of

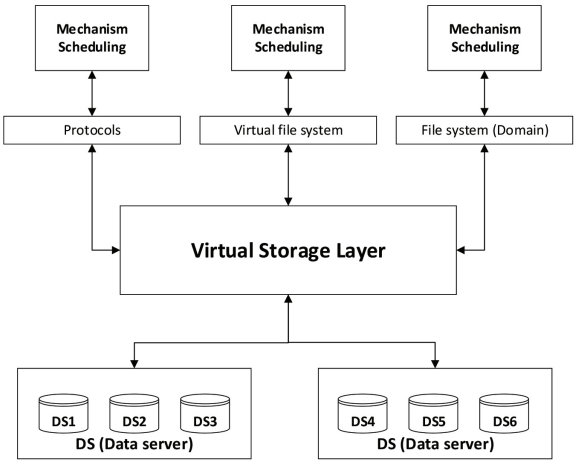

Fig. 5. Mechanisms linked to provision data in virtual storage

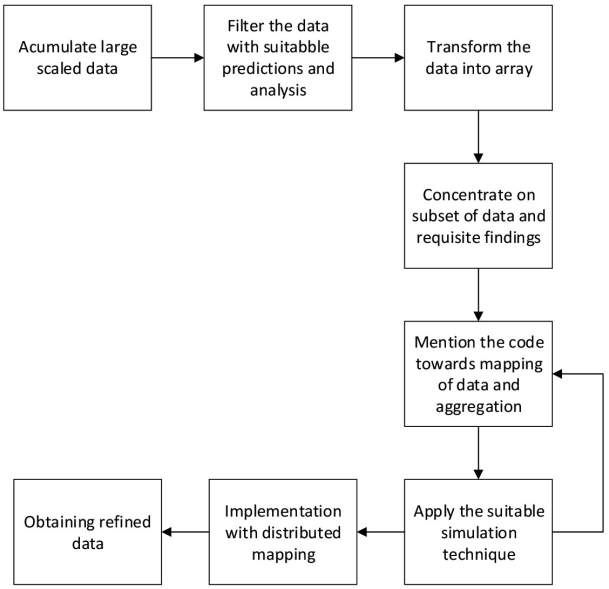

Fig. 6. Representation of refinement of large-scale data

course, the simulation of data along with network categorization is required for optimal findings. Also, large-scale data analysis in many situations can be implemented in an open-source platform comprising distributed computing and mapping systems. While focusing on the real-time distributed data, it is observed that the search engine associated with structured queries initiates and guides in querying data with the help of specified relational databases. As shown in fig 6, the large-scaled data with multidimensional features can be encapsulated and contained as per specific strategic

plans. Also, the inconsistencies and irregularities in each stage of implementation, whether temporal or spatial, should be adequately taken care of.

5. Implementation using Particle swarm optimization

Particle swarm optimization is a meta-heuristic technique that also shares commonalities with other different evolutionary computation techniques. In this case, the system can be first initialized with a population of random solutions. Then the performance applying the search heuristics can be obtained with the optimal solution by updating generations. The particles represented in terms of query plans can be processed in a group to obtain optimal results. Also, accordingly, the group can be used towards updating the position of these particles.

6. Steps to obtain the parameterized value of the particles (query plans, figure 7)

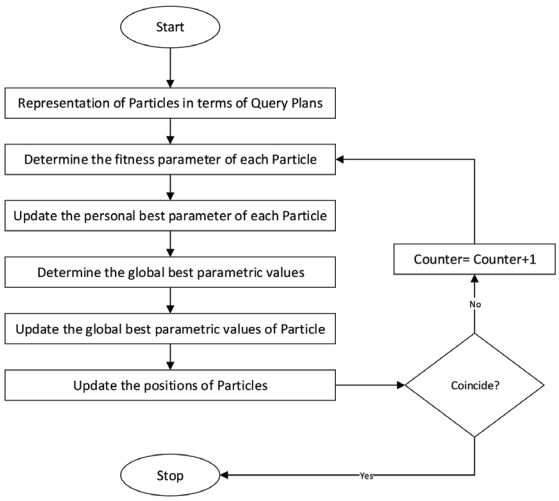

Fig. 7. Stepwise representation of Particles in terms of Query Plans

Step 1: Input the range of variable, V_Range, Particle_Velocity P_Val
Step 2: Obtain the dimension of the parameterized constraints, D
Step 3: Determine the pbest, gbest values, gradient_value
Step 4: for i=1:VarRange
if abs(gbest(i+1)-gbest(i)) < gradient_value
/* maintain the default_value; (1<=default_value<=150)
Set flag=1
Step 5: if V_Range<P_Val
/* Assign the remaining values of particles at random*/

if D<= P_Val

```
 V_Range(min)=ones(D,1)*-100;
 V_Range(max)=ones(D,1)*100;
 V_Range=[V_Range(min),V_Range(max)];
 default_value=150
 else if gradient_value== default_value
 V_Range(min)=ones(D,1)*-100;
 V_Range(max)=ones(D,1)*100;
 V_Range=[V_Range(min),V_Range(max)];
gradient_value=1
 default_value=1
 end
end
```

Step 6: Pre-allocation of variables and fix the maximum velocity and position of particles

```
if length(gradient_value)==1
 P_Val(min) = - gradient_value *ones(V_Range,D);
 P_Val(max) = gradient_value *ones(V_Range,D);
elseif length(gradient_value)==D
 P_Val(min)= min(- gradient_value, P_Val)
P_Val(max) = max(gradient_value, P_Val)
```

Step 7: Initialize the population of particles along with velocities.

formulate the position,post, i.e. based on particle_id(p_id) and dimension(parameterized constraint),D

```
post(1:p_id,1:D) = normalization(rand([p_id,D]),V_Range',1);

 if gradient_value == 1
 Acc_sz = size(P_Value);
 post(1:Acc_sz(1),1:Acc_sz(2)) = P_Value;
 end
```

Step 8: Assign initial gbest value and pbest value and check whether pbest is equivalent to gradient_value

```
 if gradient_value ==1
 [gbest_val,id_1] = max(pbest_val);
 elseif gradient_value ==0
 [gbest_val,id_1] = min(pbest_val);
 elseif gradient_value >=2
 [Acc_sz,id_1] = min((pbest_val-ones(size(pbest_val))* gradient_value).^2);
 gbest_val = pbest_val(id_1);
 end
```

Step 9: Regulate the velocities implementing masking and check the threshold value with gbest_val

```
if gbest_val >= itr_best_val
gbest_val = itr_best_val;
gbest = pbest(id_1,:);
end

else
pbest_val(tmp_i,1) = update(tmp_i,1);
pbest(tmp_i,:) = post(temp_i,:);
[itr_best_val,id_1] = max(pbest_val);
if gbest_val <= itr_best_val
gbest_val = itr_best_val;
gbest = pbest(id_1,:);

end
end
```

Step 10: Check the criteria with constrained optimization

```
if ((gbest_val<= gradient_value) | ((gbestval>= gradient_value)
if flg == 1
Accumulate values associated with gbest_val
```

7. Experimental analysis

Table 1: Position of Particle with optimization time

Sl.No.	Particle Position(gbest)	V_Range	Gradient_Value	Time(Optimization) (m.sec.)
1	4	11	2	0.29
2	7	19	2	0.47
3	11	29	2	0.70
4	19	47	2	0.70
5	20	51	2	0.83

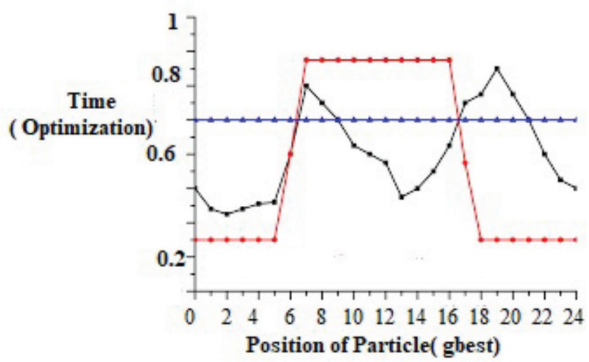

Fig. 8. Particle position (gbest) with Optimization time

Particularly, to process the large-sized queries in a proper signified manner, the mechanisms and strategies are very much essential. Also, it should be more focused on relations linked to different sites towards generating optimal query execution plans. This optimality enhances the number of accumulated query plans while processing queries. Sometimes it may not be feasible to explore all possible query plans in a large search space. In such a situation, the particle swarm optimization technique can be adopted to manage all possible query plans represented as particles provisioning the most cost-effective option towards optimization shown in figure 8. It has also been observed that the optimum time associated with the query plans is directly proportional to the particle's position at a particular gradient value.

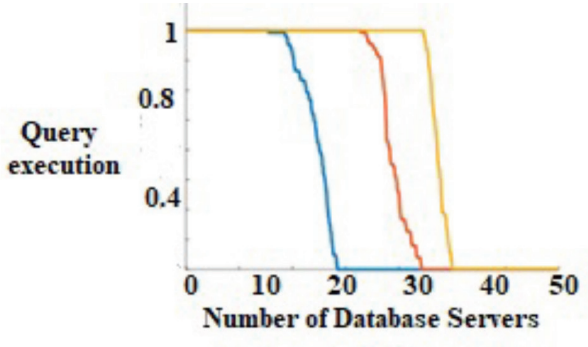

Fig. 8. Particle position (gbest) with Optimization time

Table 2: Database Servers with Query execution

Sl.No.	Number of Database Servers	Gradient_ Value	Query execution(m. Sec.)	V_Range
1	19	2	1	11
2	29	2	1	19
3	34	2	1	29
4	36	2	1	47
5	37	2	1	51

In general, the storage allocation associated with database servers can focus on many manageability problems in database systems. In such a situation, the database systems can be operated incrementally on database servers. In a similar situation, it is also required to focus on the self-managed database systems and analyze the associated computational cost. Figure 9 reflects the overhead of execution of databases associated with database servers. The constancy in query execution can be maintained at a particular gradient value irrespective of the number of database servers.

8. Discussion and future direction

The desired optimal query plans associated with the queries can be retrieved following the optimization criteria based on computational intelligence and parameters like resource availability. The desired query plans can be partial or near-optimal as the

values of cost parameters are not consistent due to the compilation process. Focusing on the parametric query optimization criteria, it is observed that every near parametric query plan is probably optimal, and the probable size of the query plans with the approximate cost parameter tends to non-linear. Accordingly, it can also be thought of optimizing the parameterized query plans dynamically in two ways to minimize query execution times. First of all, there should be the provision towards fulfilling all the statistics linked to the optimization process as well as an optimizer, and secondly, all the statistics should be computed at runtime explicitly.

9. Conclusion

This manuscript has been focused on the large-scale data with parameterized query plans. The routine associated with the application is based on nonlinearity, and the size of parametric queries may not be consistent. To explore the optimization criteria, the parametric queries are processed, and accordingly, each query plan within the parameterized space is approximated with a specified cost value. Also, to obtain better simulation, the particle swarm optimization technique is used to manage all possible query plans provisioned with cost-effective options towards optimization.

References

Bachmann, D. (2017). Contextual Model-Based Collaborative Filtering For Recommender Systems, M.S. thesis, Dept. Elect. Comput. Eng, Univ. Western Ontario, London, ON, Canada.

Chen, X. W., & Lin, X. (2014). Big data deep learning: challenges and perspectives. *IEEE access, 2,* 514-525.

Gruber, S., Logan, R. W., Jarrín, I., Monge, S., & Hernán, M. A. (2015). Ensemble learning of inverse probability weights for marginal structural modeling in large observational datasets. *Statistics in medicine, 34*(1), 106-117.

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing, 2016*(1), 1-16.

Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health information science and systems, 2*(1), 1-10.

Saeedi, R., Ghasemzadeh, H., & Gebremedhin, A. H. (2016, December). Transfer learning algorithms for autonomous reconfiguration of wearable systems. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 563-569). IEEE.

Singh, D., & Reddy, C. K. (2015). A survey on platforms for big data analytics. *Journal of big data, 2*(1), 1-20.

Tang, Y., Xu, Z., & Zhuang, Y. (2016, April). Bayesian network structure learning from big data: A reservoir sampling based ensemble method. In *International Conference on Database Systems for Advanced Applications* (pp. 209-222). Springer, Cham.

Tarasyev, A. M., Vasilev, J., Turygina, V. F., & Panchenko, A. D. (2021, March). Analysis of data on sources of official statistics, development strategy of the Kurgan region. In *AIP Conference Proceedings* (Vol. 2333, No. 1, p. 150009). AIP Publishing LLC.

Than, M. (2015). Billion Devices Will Wirelessly Connect to the Internet of Everything in 2020, ABI Research.

Wu, Y., Schuster, M., Chen, Z., et al. (2016). Google's neural machine translation

system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144.*

Yang, L., Chu, Y., Zhang, J., Xia, L., Wang, Z., & Tan, K. L. (2015, October). Transfer learning over big data. In *2015 Tenth International Conference on Digital Information Management (ICDIM)* (pp. 63-68). IEEE.

Submitted: 13.09.2020
Accepted: 28.04.2021