Journal of Scientific & Industrial Research
Vol. 81, December 2022, pp. 1310-1316
DOI: 10.56042/jsir.v81i12.54992

Feature Influence Based ETL for Efficient Big Data Management

M Vijayalakshmi* & R I Minu

SRM Institute of Science and Technology, Kattankulathur, Chengalpat 603 203, Tamil Nadu, India

Received 14 September 2021; revised 10 November 2022; accepted 11 November 2022

The increased volume of big data introduces various challenges for its maintenance and analysis. There exist various approaches to the problem, but they fail to achieve the expected results. To improve the big data management performance, an efficient real time feature influence analysis based Extraction, Transform, and Loading (ETL) framework is presented in this article. The model fetches the big data and analyses the features to find noisy records by preprocessing the data set. Further, the method performs feature extraction and applies feature influence analysis to various data nodes and the data present in the data nodes. The method estimates Feature Specific Informative Influence (FSII) and Feature Specific Supportive Influence (FSSI). The value of FSII and FSSI are measured with the support of a data dictionary. The class ontology belongs to various classes of data. The value of FSII is measured according to the presence of a concrete feature on a tuple towards any data node, whereas the value of FSSI is measured based on the appearance of supportive features on any data point towards the data node. Using these measures, the method computes the Node Centric Transformation Score (NCTS). Based on the value of NCTS the method performs map reduction and merging of data nodes. The NCTS_FIA method achieves higher performance in the ETL process. By adapting feature influence analysis in big data management, the ETL performance is improved with the least amount of time complexity.

Keywords: Cloud, FSII, FSSI, FIA, NCTS, Ontology

Introduction

The change in the informatics world in this era has increased the volume of data. The organizations maintain a variety of data about the users, customers, and employees. There is no difference among the organizations, but the kind of data differs between them. However, the structure of the database schema has great differences. Unlike traditional relational data, modern organizations maintain various unrelated data in chunks. But the volume and size of data have no bounds. The big data is the one that is being represented and has no bound for the dimension and no restriction for the volume of data. For example, consider a medical organization that maintains a variety of data regarding patients, including their general address, professional details, normal clinical history, clinical data, treatment history, advanced medical reports, images, etc. Such data forms the big data when the number of patients and users increases. Whatever the data may be, the area of application makes the difference.

Data analytics is the hot topic being discussed in recent times. The data analytics process analyses the available data to get some intelligence to support other processes. For example, consider a pharmaceutical industry that wants to fix a price or launch a product targeting a specific disease. If the organization combines the medical data in the form of big data by merging or clustering the data into groups, it can generate intelligence about how the sale differs according to the economic status of patients, how it differs with the cure rate, and how it differs with the locality of patients. Such intelligence would help the organization to decide on the launch of a product, market the product in a specific region, and fix the product price. Similarly, the outcome of data analytics would be applicable to any part of business, and so on. The same can be adapted to medical solutions in finding the optimal solution or treatment for a disease. In other words, it would be used to generate recommendations for specific treatment.

How data analytics can be applied to big data is, by clustering the data and combining various relational data to support the analysis. To perform this, ETL (Extraction, Transformation, and Loading) is performed. The organization would have big data, but the representation, storage, and format may differ from each other. But to solve the problem, the data must be converted from their heterogeneous form into

*Author for Correspondence
E-mail: vijayalam@srmist.edu.in

a homogeneous one. To perform this, the data from various nodes must be analysed and grouped to form a single entity by clustering and merging them. There are a number of approaches available to perform this, including K-means clustering, which groups the data according to the distance between the data points but is not suitable for high-dimensional big data. Also, dictionary-based approaches are used in a few methods but suffer from higher overlapping and false indexing. Also, decision tree-based approaches are used in this problem, but they are suitable when the number of decision conditions is less than six. Similarly, there are a number of approaches available in the literature, but the key issue is the accuracy of indexing. When the number of dimensions grows, it challenges the process of clustering and grouping. However, the data must be organized in an efficient manner, which directly affects the performance of intelligence generation.

Toward this end, an efficient feature influence analysis-based approach is presented. The method includes the data dictionary and ontology files to get the representation of the data given. According to them, the method applies Feature Influence Analysis (FIA) on various data features and their classes to find the class of data. By identifying the influence of any feature under any class, the data has been applied with map-reduce to perform grouping and merging. The detailed approach has been clearly discussed in the next part.

Related Works

There are a number of techniques available for the ETL process and intelligence generation. This section details a set of approaches related to the problem. The problem of latency in financial data integration has been handled through an adaptive approach that combines various hybrid ontologies related to financial data sets.[1] The Resiliently Distributed Data Stream for Online Analytical Processing (RDD4OLAP) helps in integrating the data. A comparative study on various ETL processes is presented, which considers different quality metrics and dimensions in measuring the performance of various ETL frameworks.[2] A semi-automatic inter-attribute semantic mapping that groups the attributes according to the semantic relations has been discussed.[3]

A goal modeling technique-based approach is presented towards an effective ETL process that considers quality characteristics.[4] An automatic data generator called Bijoux is presented, which extracts semantic information, transforms the data, and analyses various constraints of the input data.[5] A survey of ETL models is presented, which covers the models of programming, architecture, and security.[6] A two-level data staging scheme for optimized ETL is presented, which processes various types of data and handles slowly changing feature values.[7] Towards scientific data integration, a hybrid lazy ETL scheme is presented that integrates the data in an incremental manner that works according to the user's queries.[8] A two-phased map-reduce model named Chabok is presented, which uses a star scheme of different warehouses. The method computes the distributive measures to perform map reduction.[9]

A workflow with the focus on supporting ETL and generating semantic relationships over integrated data of a heterogeneous nature is presented by Bansal & Kagemann.[10] Abbes & Gargouri[11] presented a characteristic-based data integration model with ontology which migrates the data to Mongo DB, generates ontology, and converts it to a global one.[11] Similarly, a semantic model is presented to access multimedia big data, which recognizes the multimedia features by representing the concept and linguistic properties to fill the gaps on semantic classes.[12] An efficient anonymization scheme is presented as Pentaho Data Integration (PDI), which integrates anonymization and reidentification. The model provides protection from multiple threats.[13] A block chain-based ETL is presented, which clusters the data in such a way to support easy querying.[14]

A combination of parallelization and a shared cache memory-based approach is presented that works based on a distributed data warehouse.[15] An effective CPU-GPU heterogeneous query plan system is presented that handles the overhead to support higher performance.[16] A real-time streaming model is discussed to receive incoming streams and join according to the requirement.[17] The synchronization issue in live streams is approached with cache of physical RDBMS.[18] A column-access-aware in-stream data cache (CAIDC) is used in streaming relational data. A stream cube incremental model is discussed, to perform data collection, processing and decision making.[19] A 3-phase model is presented to support data transportation in IoT environment to handle big data.[20] A conceptual model is discussed to handle invariant dimensional data and stereotypes. Further, Data on Demand ETL (DOD-ETL) model is proposed, which pipelines on demand stream and memory cache to support effective portioning of data.[21,22] An event based model is sketched; to handle

spatial data obtained from various devices.[23] A Dynamic Multi-Variant Relational ETL model (DMVR-ETL) is presented that considers the different relationships among the data submitted to perform ETL.[24]

A real-time model is presented to perform ETL on unstructured data which is being received from multiple sources. The method pipelines the data stream and reduces the frequency of disk access.[25] A service-oriented implementation for ETL development is proposed to handle coupling issues in data applications.[26] A python based programming tool named pygrametl is presented, which provides detailed functionality on programming ETL with python language.[27] A collaborative web-based ETL model is presented, which allows the user in sharing data through pipelines and execute on different centers. A dedicated user interface is designed to support ETL.[28] Application of ETL in mining text related to employee reviews is discussed and various machine learning algorithms are evaluated with different data sets.[29] All the approaches suffer to achieve higher performance in the extract, transform, and load processes in high-dimensional data environments.

Research Gap

From the above literature survey, we identified that the methods fail to consider the influence of various features in transforming the data toward perfect learning and effective big data management. The data securities in big data transportation with IoT devices are not considered. The methods discussed in the literature survey consider different features and measures when performing big data management. The existing approaches use semantic features and ontology in the ETL process. The methods are suitable for labeled data and known data with similar schema, but not for unknown schema. Also, the security of data transportation with IoT devices is not handled properly.

Objective

Improving the ETL performance in big data management and securing the data transportation in an IoT environment.

Materials and Methods

The proposed feature-based influence analysis-based ETL model works according to the data dictionary, which belongs to various classes of data, and the ontology provided. The method performs preprocessing by applying the Feature Mean Analysis (FMA) algorithm and extracting the features from the big data tuple. Further, the method estimates Feature-Specific Informative Influence (FSII) and Feature-Specific Supportive Influence (FSSI). Using these feature values, the method chooses a data node and performs Map Reduce and merging. Finally, the method computes the value of the Node Centric Transformation Score (NCTS) to perform map, reduce, and merge. The detailed approach is discussed in this section.

The architecture of the NCTS-FIA-based ETL model has been pictured in Fig. 1, which details the pictorial representation and overall work flow of proposed model. The workings of the complete model are described in this part.

FMA Preprocessing

The input big data set given has been read toward the elimination of noise from the data set. Further, the method fetches each tuple T from the data set given and traverses through the features of the data point. If any of the features T_i from the tuple T is identified as incomplete or the feature T_i has empty values, then it has been considered noisy and incomplete. For the feature identified as incomplete or missing, the method applies the Feature Mean Analysis (FMA), which estimates the mean value of the concerned feature according to the feature values of the several data points. An estimated mean value has been placed on the tuple feature. If the feature is alphanumeric, then it has been placed with the maximum occurring feature value. The noise-removed data point has been used for the ETL process.

Consider the sample dataset given in Table 1, which contains the details of COVID-19 being observed by different users with their sample collection. By applying the FMA, algorithm, the method identifies the noisy records (3, 4, 5) with missing features and eliminates them from the set.

The result of FMA preprocessing is presented in Table 2, which is used towards ETL processing.

Fig. 1 — Architecture of proposed NCTS-FIA based ETL framework

Table 1 — Sample Data set

No	Aadhaar number of the patient	Name of the Patient	Name of Sample	COVID result	Sample Collected date	Feedback	Result Date
1	1234	Ravi	CT	Yes	10/12/22	Ok	16/12/22
2	4567	Sankar	RTPCR	Yes	28/11/22	Good	1/12/22
3	6789	Raja	CT	No	—	Good	11/12/22
4	5678	Kumar	Blood	—	—	Good	—
5	3456	Samy	CT	Yes	9/11/22	—	10/11/22
6	7890	Sethu	CT	Yes	5/10/22	Poor	9/10/22

Table 2 — Result of FAM preprocessing

No	Aadhaar number of the patient	Name of the Patient	Name of Sample	COVID result	Sample Collected date	Feedback	Result Date
1	1234	Ravi	CT	Yes	10/12/22	Ok	16/12/22
2	4567	Sankar	RTPCR	Yes	28/11/22	Good	1/12/22
6	7890	Sethu	CT	Yes	5/10/22	Poor	9/10/22

FMA Algorithm:
Given: Data set Ds
Obtain: Noise Removed data set Dds

1 Start
2 Read Ds.
3 //$\in$ −represents A contained in B, $\ni$ −represents A doesnot contained in B
4 Find Feature set Fes = $\sum_{i=1}^{size(Ds)}$ Features $\in$ Ds(i)&&i.Features$\ni$Fes
5 Initialize mean set Ms = size (Fes)
6 For each feature f
7 //$\sum -represents\ summation\ of\ specific\ value$, size (Ds) -represents total data points of data set Ds.
8 Compute feature mean Femean = $\frac{\sum_{i=1}^{size(Ds)} DS(i).f}{size(DS)}$
9 End
10 For each tuple T
11 If T contains $\forall Features \in Fes$ then // means if T contains all features of set Fes.
12 If T (Feature) == Null then
13 T (Feature) = Ms (Feature)
14 End
15 Else
16 Remove the tuple T from the set.
17 Ds =DS $\cap$ T
18 End
19 End
20 Stop

The above-discussed feature means the analysis algorithm finds the noisy record with missing values and adjusts the value with the mean value. The preprocessed data set has been used to perform extraction, transformation, and loading.

Feature Specific Informative Influence (FSII) Estimation

The feature-specific informative influence measure represents the tuple of influence it has on the informative features. Any data point or tuple would have a number of features, but in order to become a member of a class, it must have specific features that are more concrete. For example, a person must have specific features to belong to a particular gender. Similarly, for a tuple to become a member of a class, it must have a set of features, which we call "informative features." According to the informative features the data point has, the method computes the value of the feature-specific informative influence measure (FSII). It has been measured according to the appearance of class features of an informative nature and the total number of features it has. The estimated value of FSII has been used towards map reduction and merging.

Algorithm:
Given: Data Point D, Data Dictionary Ddict, Ontology O
Obtain: FSII

1 Start
2 Read data point D, data dictionary Ddict, ontology O.
3 Find feature list Flist = $\sum$ Features $\in$ D
4 ////Flist(i) $\in$ Ddict - represents sum of all features present in data dictionary, size (Ddict)-//number of records in data dictionary, $\sum_{i=1}^{size(Flist)}$ Flist(i) $\in$ O&& $Type == Informative$ – //number/of features are informative.
5 Compute FSII = $(\frac{\sum_{i=1}^{size(Flist)} Flist(i)\in Ddict}{size(Ddict)} + \frac{\sum_{i=1}^{size(Flist)} Flist(i)\in O\&\& Type==Informative}{size(O)})/2$
6 Stop

The above-discussed algorithm estimates the feature-specific informative influence measure according to the presence of a feature in the data dictionary of any class, and the ontology belongs to that. The estimated value of FSII has been used in map-reduce and the merging of data sets.

Feature Specific Supportive Influence (FSSI) Estimation

The supportive influence is the measure that represents the support that the data point provides to be a part of that according to the supportive features. Unlike informative features, supportive features are not part of the class, but they still encourage the data point to become part of that. For example, even though the clinical feature age and the demographic feature sex are not part of the cardiac data set, the presence of such features encourages the completeness of the data groups. Accordingly, the method computes the value of FSSI based on the presence of a certain number of supportive features in the data point and the total number of features in the data point. The estimated value of FSSI has been used towards map reduction and merging.

FSSI Estimation Algorithm:

Given: Data Point D, Data Dictionary Ddict, Ontology O

Obtain: FSII

1 Start
2 Read data point D, data dictionary Ddict, ontology O.
3 //$\sum$ Features $\in$ D -- counts the number of features present in data point
4 Find feature list Flist = $\sum$ Features $\in$ D
5 //$\sum_{i=1}^{size(Flist)}$ Flist(i) $\in$ O&& $Type ==$ Supportive -- counts no of them are supportive.
6 Compute FSSI = ($\frac{\sum_{i=1}^{size(Flist)} Flist(i)\in O \&\& Type==Supportive}{size(Flist)}$
7 Stop

The above-discussed algorithm computes the value of feature-specific supportive influence (FSSI) according to the presence of supportive features in the class ontology. Estimated values have been used to perform data merging and map reduction.

NCTS_FIA ETL Process

The proposed approach performs ETL processes according to the Node Centric Transformation Score (NCTS). To achieve this, the method first preprocesses the given data set. As the model maintains various class dictionaries and ontologies dedicated to various classes, the method preprocesses the data set and extracts the features to compute the value of feature-specific informative influence (FSII) and feature-specific supportive influence (FSSI) values towards various classes. Finally, the method computes the value of NCTS for different classes. A class of nodes has been selected with maximum NCTS and computes the average FSSI value for all the tuples in the class, both internally and externally. The method computes influence frequency according to the values from the samples of a specific data node and computes inverse influence frequency with the tuple values of other data nodes. Both of them are measured based on the supportive features themselves. Now, based on the values of both NCTS and IF, the method performs merging and map reduction.

Algorithm:

Given: Data Set Ds, Dictionary set Dset, Ontology Set Os

Obtain: Dictionary set Dset, Ontology set Os.

1 Start
2 Read Ds, Dset, Os.
3 Ds = Perform Preprocessing
4 For each data tuple T
5 For each data class Dc
6 Compute PSSI = PSSI_Estimation (Dset(Dc),Os(Dc),T)
7 Compute PSII = PSSI_Estimation (Dset(Dc),Os(Dc),T)
8 Compute NCTS = PSSI × PSII
9 End
10 End
11 Class C = Choose the class with maximum NCTS.
12 // below equation counts supportive features of the class
13 Compute Influence Frequency IF = $\frac{\sum_{j=1}^{size(DC)} DC(j)\in SupportiveFeature(i)}{size(Dc)}$ $\frac{i=1}{size(SupportiveFeatures)}$
14 //below equation counts the features present in other classes supportive features.
15 Compute Inverse Influence Frequency IIF = $\frac{\sum_{j=1}^{size(OC)} OC(j)\in SupportiveFeature(i)}{size(Oc)}$ $\frac{i=1}{size(SupportiveFeatures)}$
16 Compute Influence weight Iw = IF × IIF
17 If Iw of the class DC is greater than other classes then
18 MapReduce
19 Choose Supportive features with value IF>Th.
20 Else
21 Perform merging.
22 End
23 Stop

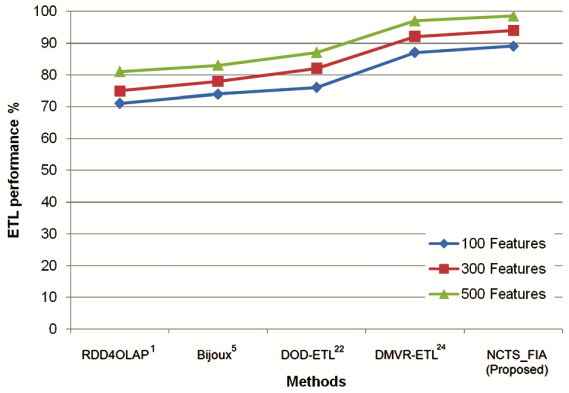

Fig. 2 — ETL performance

Table 3 — Test Bed Details

Factor	Value
Deployment Tool	Microsoft Azure
Number of Data Source	50
Total Features	500
Data Volume	One Million
Data Set	COVID-19 Data Set

The above-discussed algorithm shows how the proposed approach performs map reduction and merging. It has been performed by measuring NCTS, IF, IIF, and IW values. Based on these values, the feature selection is performed with an eye towards map reduction, and with the same, the method performs the merging of data points from different nodes.

Results and Discussion

The proposed NCTS-FIA model is hardcoded in Microsoft azure platform. The method has been validated for its efficiency under different parameters. The evaluation is carried out according to different features, data dictionaries and ontologies.

The test bed considered for evaluating the performance of proposed model is displayed in Table 3. The data set is combined with cardiac, lung, diabetic and clinical data obtained from various patients of COVID-19. According to the data collected, the NCTS-FIA model is evaluated for its performance. The data set is framed up to one million records, which are obtained from different sources such as, American Health Department.

The ETL performance introduced by various methods are measured and plotted in Fig. 2. The proposed NCTS_FIA model triggered higher performance than any other technique at all the test cases.

The value of overlap generated at different cases are measured for various approaches and sketched in Fig. 3. The NCTS_FIA model introduces only least overlap at all test cases.

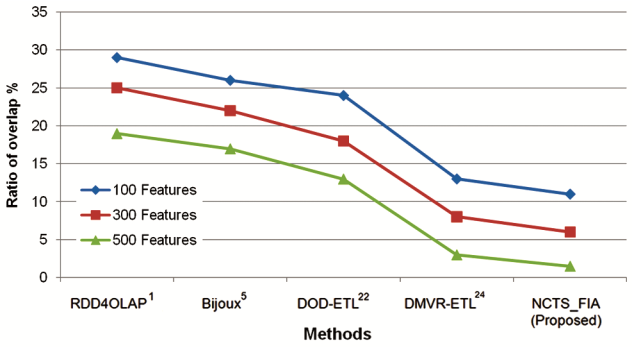

Fig. 3 — Analysis on overlap

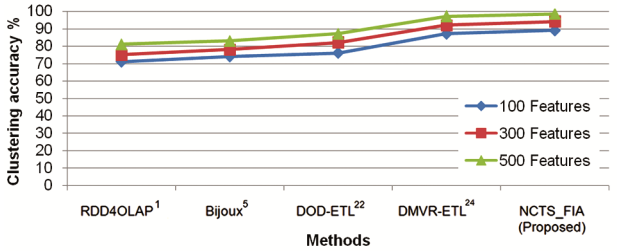

Fig. 4 — Analysis on clustering

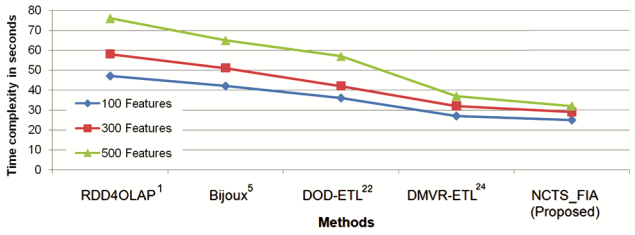

Fig. 5 — Comparison of proposed method with existing ones on Time Complexity

The clustering accuracy introduced by different methods are measured and plotted in Fig. 4. The proposed NCTS_FIA model introduces higher accuracy than others.

The time complexity introduced by different models are measured and displayed in Fig. 5. The proposed NCTS-FIA model shows least value than others.

Conclusions

This paper presented a novel node-centric transition score with a feature influence analysis-based ETL framework. The model performs preprocessing on the data set according to the data dictionary and ontology given. With the preprocessed data set, the method estimates FSSI (Feature Specific Supportive Influence) and FSII (Feature Specific Informative Influence) values for each tuple towards various classes of data nodes. Now, using them, the method computes the value of NCTS to decide which

class the data point belongs to. Further, the method computes the influence frequency (IF) and Inverse Influence Frequency (IIF) to compute the influence weight to decide on the merging or Map Reduce. However, map reduction is performed according to the value of IF for different features of any class. The proposed method produces efficient results on ETL with less time complexity.

References

1 Fikri N, Rida M, Abghour N, Moussaid K & El Omri A, An adaptive and real-time based architecture for financial data integration, Springer, *J Big Data*, **6** (2019) Article number 97, https://doi.org/10.1186/s40537-019-0260-x.

2 Souibgui M & Atigui F, Zammali S, Cherfi S & Yahia S B, Data quality in ETL process: A preliminary study, Elsevier, *Procedia Comput Sci*, **159** (2019) 676–687, https://doi.org/10.1016/j.procs.2019.09.223.

3 Bergamaschi S, Guerra F, Orsini M, Sartori C & Vincini M, A semantic approach to ETL technologies, *J Data Knowl Eng*, **70** (2011) 717–731.

4 Theodorou V, Abelló A, Lehner W & Thiele M, Quality measures for ETL processes: from goals to implementation, *J Concurr Comput Pract Exp*, **28** (2016) 3969–3993.

5 Theodorou V, Jovanovic P, Abelló A & Nakuçi E, Data generator for evaluating ETL process quality, *J Inf Syst*, **63** (2017) 80–100.

6 Mohammed Muddasir N, Raghuveer K & Dayanand R, Study of ETL optimization techniques in big data, *Int J Adv Sci Technol*, **29(5)** (2020), 13194–13209.

7 Liu X, Iftikhar N, Huo H, Nielsen P S & Huo H, Optimizing ETL by a two-level data staging method, *Int J Data Warehous Min*, **12(3)** (2016) 32–50.

8 Kathiravelu P, Sharma A, Galhardas H, Van Roy P & Veiga L, On-demand big data integration: A hybrid ETL approach for reproducible scientific research, *Distrib Parallel Databases*, **37** (2019) 273–295, https://doi.org/10.48550/arXiv.1804.08985

9 Barkhordari M & Niamanesh M, Chabok: a Map-Reduce based method to solve data warehouse problems, *J Big Data*, **5** (2018), Article number 40, https://doi.org/10.1186/s40537-018-0144-5.

10 Bansal S K & Kagemann S, Integrating big data: A semantic extract-transform-load framework, *Computer* (Long. Beach. Calif), **48** (2015) 42–50.

11 Abbes H & Gargouri F, Big data integration: A Mongo DB database and modular ontologies based approach, *Procedia Comput Sci* **96** (2016) 446–455.

12 Rinaldi A M & Russo C, A semantic-based model to represent multimedia big data*, MEDES '18: Proc 10th Int Conf Manag Digit EcoSyst*, (September 2018) 31–38, https://doi.org/10.1145/3281375.3281386.

13 Prasser F, Spengler H, Bild R, Eicher J & Kuhn K A, Privacy-enhancing ETL-processes for biomedical data, *Int J Med Inform*, **126** (2019) 72–81.

14 Galici R, Ordile L, Marchesi M, Pinna A & Tonelli R, Applying the ETL process to blockchain data prospect and findings, *Information*, **11(4)** (2020) 204 https://doi.org/10.3390/info11040204.

15 Faridi Masouleh M, Afshar Kazemi M A, Alborzi M, & Toloie Eshlaghy A, Optimization of ETL process in data warehouse through a combination of parallelization and shared cache memory, *Eng Technol Appl Sci Res*, **6(6) (**2016) 1241–1244, https://doi.org/10.48084/etasr.849.

16 Lee S, Performance analysis of big data ETL process over CPU-GPU heterogeneous architectures, *IEEE Trans Parallel Distrib Syst*, 42–47 (2020), 10.1109/ICDEW53142.2021.00015.

17 Biswas N, Sarkar A & Mondal K C, Efficient incremental loading in ETL processing for real-time data integration, *Innov Syst Softw Eng*, **16** (2019) 53–61.

18 Ma K & Yang B, Column access-aware in-stream data cache with stream processing framework, *J Signal Process Syst,* **86(2)** 191–205 (2017).

19 Zheng T, Chen G, Wang X, Chen C, Wang X & Luo S, Real-time intelligent big data processing: Technology platform and applications, *Sci China Inf Sci*, **62(8)** (2019) 82101.

20 Babar M & Arif F, Real-time data processing scheme using big data analytics in internet of things based smart transportation environment, J *Ambient Intell Hum Comput*, **10(10)** (2019) 4167–4177.

21 Bouali H, Akaichi J & Gaaloul A, Real-time data warehouse loading methodology and architecture: A healthcare use case, *Int J Data Anal Techn Strategies*, **11(4)** (2019) 310–327.

22 Machado G V, Cunha I, Pereira A C M & Oliveira L B, DOD-ETL: Distributed on-demand ETL for near real-time business intelligence, *J Internet Serv Appl*, **10(1)** (2019) Article number 21, https://doi.org/10.1186/s13174-019-0121-z.

23 Rieke M, Bigagli L, Herle S, Jirka S, Kotsev A & Liebig T, Geospatial IoT—The need for event-driven architectures in contemporary spatial data infrastructures, *ISPRS Int J Geo-Inf*, **7(10)** (2018) 385.

24 Manickam V & Rajasekaran Indra M, Dynamic multi-variant relational scheme-based intelligent ETL framework for healthcare management, *Soft Comput* (in press), (2022). https://doi.org/10.1007/s00500-022-06938-8.

25 Mehmood E & Anees T, Distributed real-time ETL architecture for unstructured big data, Knowledge and Information System**, 64** (2022) 3419–3445.

26 Oliveira B, Leite M, Oliveira Ó & Belo O, A service-oriented framework for ETL implementation, in *Progress in Artificial Intelligence. EPIA 2022. Lecture Notes in Computer Science*, vol 13566, edited by G Marreiros, B Martins, A Paiva, B Ribeiro, A Sardinha (Springer, Cham) https://doi.org/10.1007/978-3-031-16474-3_52.

27 Jensen S K, Thomsen C, Pedersen T B & Andersen O, Pygrametl: A powerful programming framework for easy creation and testing of ETL flows, *Transactions on Large-Scale Data- and Knowledge-Centered Systems* (Springer), **12670** (2021) 45–84, https://doi.org/10.1007/978-3-662-63519-3_3.

28 Almeida J R, Coelho L & Oliveira J L, BIcenter: A collaborative Web ETL solution based on a reflective software approach, *SoftwareX*, **16** (2021) (100892), https://doi.org/10.1016/j.softx.2021.100892.

29 Tanasescu L G, Vines A, Bologa A R & Vaida C A, Big data ETL process and its impact on text mining analysisfor employees, reviews, *Appl Sci*, **12** (2022) 7509, https://doi.org/10.3390/app12157509.