

FORM 5

THE PATENTS ACT, 1970
(39 of 1970)

&

The Patents Rules, 2003
DECLARATION AS TO INVENTORSHIP
[See section 10(6); rule 13(6)]

We Dr.Haritha Thotakura, Dr.K. Srihari Rao, **Mr. Mullapudi Rama Krishna**, Sunitha Ravi and Lanka Padmalatha hereby declare that the true and first inventors of the invention disclosed in the complete specification filed in pursuance of our application dated 27-08-2020

Dated this 27th day of August, 2020

Signature

Haritha T

Dr.Haritha Thotakura

INDIAN

D/o T.Mohan Rao

Associate Professor, ECE Department,

Prasad V Potluri Siddhartha Institute of Technology,

Kanuru, Vijayawada-520007, Andhra Pradesh, India.

Signature

K. Sri Hari Rao

Dr K. Srihari Rao

INDIAN

S/o K. Pulla Rao

Professor & HOD,

NRI Institute of Technology,

Visadala, Guntur - 522438

Andhra Pradesh, India

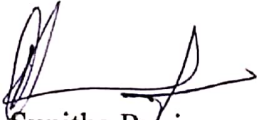
Signature



Mr Mullapudi Rama Krishna
INDIAN

S/o M. Madhusudhana Rao
Associate Professor & HOD, ECE Department ,
Andhra Loyola Institute of Engineering and Technology, Vijayawada-8
Andhra Pradesh, India.

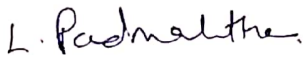
Signature



Sunitha Ravi
INDIAN

D/o Ravi Sivaramaih
Professor , ECE Department,
NRI Institute of Technology,
Pothavarappadu, Nunna Rd, Agiripalli - 521212
Andhra Pradesh, India.

Signature



Lanka Padmalatha
INDIAN
D/o Lanka Raja Gopal Rao
Assistant Professor , ECE Department,
Gudlalleru Engineering College.
Gudlalleru - 521356,
Andhra Pradesh, India.

To

The Controller of Patents,
The Patent Office, At CHENNAI.

(54) Title of the invention : RECOGNIZING HUMAN FACIAL EMOTION AND DETECTION UTILIZING DEEP LEARNING

<p>(51) International classification</p> <p>(31) Priority Document No</p> <p>(32) Priority Date</p> <p>(33) Name of priority country</p> <p>(86) International Application No Filing Date</p> <p>(87) International Publication No</p> <p>(61) Patent of Addition to Application Number Filing Date</p> <p>(62) Divisional to Application Number Filing Date</p>	<p>:G06K0009620000, G06K0009000000, G06N0003040000, G06N0003080000, G06N0020000000</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>: NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p>	<p>(71)Name of Applicant :</p> <p>1)Dr. G S PRADEEP GHANTASALA Address of Applicant :Associate Professor, Department of Computer Science & Engineering, Chitkara University Institute of Engineering & Technology, Chitkara University, Punjab-140603, INDIA Punjab India</p> <p>2)VENKATARAO MADDUMALA</p> <p>3)Dr. LOKAIAH PULLAGURA</p> <p>4)Dr. RAJENDRA BABU CHIKKALA</p> <p>5)Dr. SAMBASIVA NAYAK</p> <p>6)SREENIVASA RAO KAKUMANU</p> <p>7)MADHUSUDHAN RAO DONTA</p> <p>8)Dr. K. R.R. MOHAN RAO</p> <p>9)Dr. RATNABABU PILLI</p> <p>10)Dr. SHOBANA GORINTLA</p> <p>(72)Name of Inventor :</p> <p>1)Dr. G S PRADEEP GHANTASALA</p> <p>2)VENKATARAO MADDUMALA</p> <p>3)Dr. LOKAIAH PULLAGURA</p> <p>4)Dr. RAJENDRA BABU CHIKKALA</p> <p>5)Dr. SAMBASIVA NAYAK</p> <p>6)SREENIVASA RAO KAKUMANU</p> <p>7)MADHUSUDHAN RAO DONTA</p> <p>8)Dr. K. R.R. MOHAN RAO</p> <p>9)Dr. RATNABABU PILLI</p> <p>10)Dr. SHOBANA GORINTLA</p>
--	---	---

(57) Abstract :

Feelings are a major piece of human correspondence. Detecting and recognizing human emotion is a big challenge in computer vision and artificial intelligence. Though there are methods to identify expressions using machine learning and Artificial Intelligence techniques, here we use deep learning and image classification method to recognize expressions and classify the expressions according to the images. With the remarkable success of Deep Learning the different types of architecture techniques are exploited to achieve a better performance. We give an extensive learning of Facial appearance recognition with Deep Learning techniques which incorporates diverse Neural Network Algorithms utilized with various datasets and its productivity result.

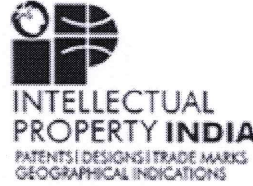
No. of Pages : 10 No. of Claims : 6



Office of the Controller General of Patents, Designs & Trade Marks
Department of Industrial Policy & Promotion,
Ministry of Commerce & Industry,
Government of India

सत्यमेव जयते

(<http://ipindia.nic.in/index.htm>)



(<http://ipindia.nic.in/index.htm>)

Application Details

APPLICATION NUMBER	202141033365
APPLICATION TYPE	ORDINARY APPLICATION
DATE OF FILING	24/07/2021
APPLICANT NAME	1 . Dr. VUPPANAPALLI SHANMUKHA RAO 2 . Dr. A. SRINIVASA RAO 3 . KOSURU SIVRAMA KRISHNA
TITLE OF INVENTION	A SYSTEM TO ENHANCE SECURITY AND PREVENT LOSS OF DATA WHILE ROUTING IN FSO MANET
FIELD OF INVENTION	COMMUNICATION
E-MAIL (As Per Record)	ipr@akshipassociates.com
ADDITIONAL-EMAIL (As Per Record)	akshipassociates@gmail.com
E-MAIL (UPDATED Online)	
PRIORITY DATE	
REQUEST FOR EXAMINATION DATE	--
PUBLICATION DATE (U/S 11A)	30/07/2021

Application Status

APPLICATION STATUS	Awaiting Request for Examination
--------------------	---

[View Documents](#)



Office of the Controller General of Patents, Designs & Trade Marks
 Department of Industrial Policy & Promotion,
 Ministry of Commerce & Industry,
 Government of India

(<http://ipindia.nic.in/index.htm>)



(<http://ipindia.nic.in/index.htm>)

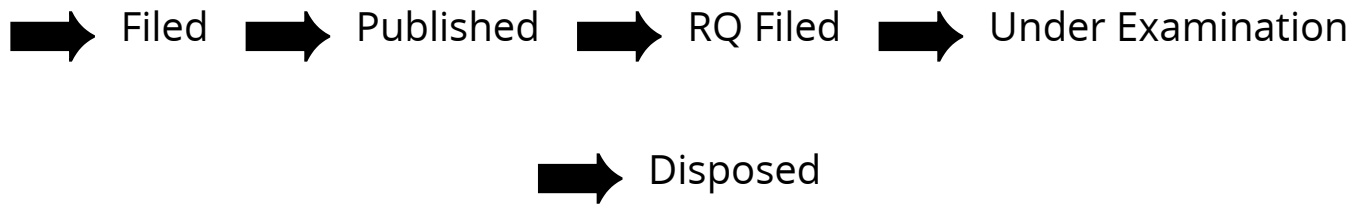
Application Details

APPLICATION NUMBER	201841031037
APPLICATION TYPE	ORDINARY APPLICATION
DATE OF FILING	20/08/2018
APPLICANT NAME	DR. RATNA KISHORE DASARI
TITLE OF INVENTION	NOVEL JOINT CHAINING GRAPH MODEL FOR HUMAN POSE ESTIMATION ON 2D ACTION VIDEOS AND FACIAL POSE STIMATION ON 3D IMAGES
FIELD OF INVENTION	COMPUTER SCIENCE
E-MAIL (As Per Record)	
ADDITIONAL-EMAIL (As Per Record)	ratna.dasari@gmail.com
E-MAIL (UPDATED Online)	
PRIORITY DATE	
REQUEST FOR EXAMINATION DATE	--
PUBLICATION DATE (U/S 11A)	21/02/2020

Application Status

APPLICATION STATUS	Awaiting Request for Examination
--------------------	---

[View Documents](#)



In case of any discrepancy in status, kindly contact ipo-helpdesk@nic.in



Controller General of Patents, Designs and Trademarks
Department of Industrial Policy and Promotion
Ministry of Commerce and Industry

Application Details

APPLICATION NUMBER	202011018787
APPLICATION TYPE	ORDINARY APPLICATION
DATE OF FILING	02/05/2020
APPLICANT NAME	1 . DR. K. RAMKUMAR (ASSOCIATE DEAN (E & T) & PROFESSOR) 2 . DR. S. A. KALAISELVAN (PROFESSOR) 3 . MRS. P. MANJULA (ASSISTANT PROFESSOR) 4 . DR. R. RAJA (ASSOCIATE PROFESSOR) 5 . DR. G. GOVINDA RAJULU (PROFESSOR) 6 . PAKALAPATI MANIKYA PRASUNA (ASSOCIATE PROFESSOR)
TITLE OF INVENTION	MDD-DATA SYSTEMATIC ANALYSIS: METADATA AND DATASET DISCOVERY UNING SYSTEMATIC ANALYSIS
FIELD OF INVENTION	COMPUTER SCIENCE
E-MAIL (As Per Record)	ramkumar1975@gmail.com
ADDITIONAL-EMAIL (As Per Record)	dr.bksarkar2003@yahoo.com
E-MAIL (UPDATED Online)	
PRIORITY DATE	
REQUEST FOR EXAMINATION DATE	--
PUBLICATION DATE (U/S 11A)	19/06/2020

Application Status

[View Documents](#)

पेटेंट कार्यालय
शासकीय जर्नल

**OFFICIAL JOURNAL
OF
THE PATENT OFFICE**

निर्गमन सं. 25/2020
ISSUE NO. 25/2020

शुक्रवार
FRIDAY

दिनांक: 19/06/2020
DATE: 19/06/2020

पेटेंट कार्यालय का एक प्रकाशन
PUBLICATION OF THE PATENT OFFICE

(54) Title of the invention : MDD-DATA SYSTEMATIC ANALYSIS: METADATA AND DATASET DISCOVERY UNING SYSTEMATIC ANALYSIS

<p>(51) International classification</p> <p>(31) Priority Document No</p> <p>(32) Priority Date</p> <p>(33) Name of priority country</p> <p>(86) International Application No Filing Date</p> <p>(87) International Publication No</p> <p>(61) Patent of Addition to Application Number Filing Date</p> <p>(62) Divisional to Application Number Filing Date</p>	<p>:G06Q0010060000, G06F0016250000, G06F0016160000, G06F0008300000, G06F0008100000</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p> <p>:NA</p>	<p>(71)Name of Applicant :</p> <p>1)DR. K. RAMKUMAR (ASSOCIATE DEAN (E & T) & PROFESSOR) Address of Applicant :DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. SRM UNIVERSITY, DELHI NCR, HARYANA-131029, INDIA. E-MAIL: ramkumar1975@gmail.com Haryana India</p> <p>2)DR. S. A. KALAISELVAN (PROFESSOR)</p> <p>3)MRS. P. MANJULA (ASSISTANT PROFESSOR)</p> <p>4)DR. R. RAJA (ASSOCIATE PROFESSOR)</p> <p>5)DR. G. GOVINDA RAJULU (PROFESSOR)</p> <p>6)PAKALAPATI MANIKYA PRASUNA (ASSOCIATE PROFESSOR)</p> <p>(72)Name of Inventor :</p> <p>1)DR. K. RAMKUMAR (ASSOCIATE DEAN (E & T) & PROFESSOR)</p> <p>2)DR. S. A. KALAISELVAN (PROFESSOR)</p> <p>3)MRS. P. MANJULA (ASSISTANT PROFESSOR)</p> <p>4)DR. R. RAJA (ASSOCIATE PROFESSOR)</p> <p>5)DR. G. GOVINDA RAJULU (PROFESSOR)</p> <p>6)PAKALAPATI MANIKYA PRASUNA (ASSOCIATE PROFESSOR)</p>
--	--	---

(57) Abstract :

My Invention MDD-Data Systematic Analysis is A system may receive a request to derive an output double variable from a source double variable. The request may include predefine proposed logic to derive the output double variable from the source double variable. The Invention may then compare the predefine proposed logic to existing logic to determine the predefine proposed logic is new. In response to the predefine proposed logic being new, the system may generate transformation code configured to execute the proposed logic. The system may further schedule the transformation code for execution at a predetermined time, and then execute the transformation code to generate data for the output double variable. It is realized here that finding the appropriate datasets for a given analysis or experiment can be one of the most challenging aspects of a data science, data mining, cloud science invention. The enterprise management system includes transaction/analytic applications and an archiving system in which data object lifecycles are pre-computed when the data object is created by the transaction application or analytic application. Having pre-computed the data lifecycle via the transaction/analytic applications, an archiving system need not re-determine whether the criteria for archiving are met. When the archiving system is initiated, the archiving system may identify the data objects having lifecycle dates that match the current date and archive them directly. The initial work package defines at least one hypothesis associated with a given data problem, and is generated in accordance with one or more phases of an automated data analytics lifecycle. A plurality of datasets is identified. One or more datasets in the plurality of datasets that are relevant to the at least one hypothesis are discovered. The at least one hypothesis is tested using at least a portion of the one or more discovered datasets. The method comprises the following steps. An initial work package is obtained. The initial work package defines at least one hypothesis associated with a given data problem, and is generated in accordance with one or more phases of an automated data analytics lifecycle.

No. of Pages : 25 No. of Claims : 8

FORM 1 THE PATENTS ACT 1970(39 of 1970) & The Patents Rules, 2003 APPLICATION FOR GRANT OF PATENT (See sections 7, 54 & 135 and rule 20(1))	<u>(FOR OFFICE USE ONLY)</u> Application No: Filing Date Amount of Fees Paid: CBR NO: Signature:
1. APPLICANTS REFERENCE / IDENTIFICATION NO. (AS ALLOTTED BY OFFICE)	

2. TYPE OF APPLICATION [Please tick (✓) at the appropriate category]

Ordinary (✓)		Convention ()		PCT-NP ()	
Divisional ()	Patent of Addition ()	Divisional ()	Patent of Addition ()	Divisional ()	Patent of Addition ()

3A. APPLICANT(S):

Name	Country of Residence	Nationality	Address
DR. K. RAMKUMAR (ASSOCIATE DEAN (E & T) & PROFESSOR)	INDIA	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. SRM UNIVERSITY, DELHI NCR, HARYANA- 131029, INDIA. E-MAIL: ramkumar1975@gmail.com
DR. S. A. KALAISELVAN (PROFESSOR)	INDIA	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD, TELANGANA- 500014, INDIA. E-MAIL: kalaiselvanresearch@gmail.com
MRS. P. MANJULA (ASSISTANT PROFESSOR)	INDIA	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. SRM UNIVERSITY, DELHI NCR, HARYANA- 131029, INDIA. E-MAIL : gpmanjula28@gmail.com
DR. R. RAJA (ASSOCIATE PROFESSOR)	INDIA	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD, TELANGANA- 500014, INDIA. E-MAIL: nsraja1984@gmail.com
DR. G. GOVINDA RAJULU (PROFESSOR)	INDIA	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD, TELANGANA- 500014, INDIA. E-MAIL: rajlug7@gmail.com
PAKALAPATI MANIKYA PRASUNA (ASSOCIATE	INDIA	AN INDIAN NATIONAL	DEPT. OF COMPUTER SCIENCE AND ENGINEERING. ANDHRA LAYOLA INSTITUTE OF

PROFESSOR)			ENGINEERING AND TECHNOLOGY, VIJAYAWADA-520008 ANDHRAPRADESH, INDIA E-Mail: prasunamanikya@gmail.com
------------	--	--	--

3B. CATEGORY OF APPLICANT [Please tick (✓) at the appropriate category]

Natural Person (✓)	Other than Natural Person ()		
	Small Entity ()	Startup ()	Others ()

4. INVENTOR(S): [Please tick (✓) at the appropriate category]

Are all the inventor(s) same as the applicant(s) named above?	Yes (✓)	No ()
---	-----------	--------

If "No", furnish the details of the inventor(s) N.A

Name	Nationality	Country of Residence	Address
NA	NA	NA	NA

5. TITLE OF THE INVENTION:

MDD-Data Systematic Analysis: METADATA AND DATASET DISCOVERY UNING SYSTEMATIC ANALYSIS

6. AUTHORISED REGISTERED PATENT AGENT(S)		NA
ADDITIONAL PATENT AGENTS	NA	

7. ADDRESS FOR SERVICE OF APPLICANT/ PATENT AGENT(S) IN INDIA DR. K. RAMKUMAR (ASSOCIATE DEAN (E & T) & PROFESSOR) Address: DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. SRM UNIVERSITY, DELHI NCR, HARYANA-131029, INDIA. E-MAIL: ramkumar1975@gmail.com	Mobile No.: 8059794469 E-mail: dr.bksarkar2003@yahoo.in
--	---

8. IN CASE OF APPLICATION CLAIMING PRIORITY OF APPLICATION FILED IN CONVENTION COUNTRY, PARTICULARS OF CONVENTION APPLICATION: N.A

Country	App. Number	Filing Date	Name of the Applicant	Title of the Invention	IPC (as classified in the convention country)

NA	NA	NA	NA	NA	NA
----	----	----	----	----	----

9. IN CASE OF PCT NATIONAL PHASE APPLICATION, PARTICULARS OF INTERNATIONAL APPLICATION FILED UNDER PATENT CO-OPERATION TREATY (PCT):

International application number	International filing date as allotted by the receiving office.
NA	NA

10. IN CASE OF DIVISIONAL APPLICATION FILED UNDER SECTION 16, PARTICULARS OF ORIGINAL (FIRST) APPLICATION: N.A

Original (first) application number	Date of filing of Original (first) application
N.A.	N.A.

11. IN CASE OF PATENT OF ADDITION FILED UNDER SECTION 54, PARTICULARS OF MAIN APPLICATION OR PATENT: N.A

Main application / patent Number	Date of filing of main application
N.A.	N.A.

12. DECLARATIONS:

(i) Declaration by the Inventor:

(In case the applicant is an assignee: the inventor(s) may sign herein below or the applicant may upload the assignment or enclose the assignment with this application for patent or send the assignment by post/electronic transmission duly authenticated within the prescribed period).

We, the above named inventor is the true & first inventor for this invention and declare that the applicant herein is my assignee or legal representative:

NA

(ii) Declaration by the applicant/s in the convention country:

(In case the applicant in India is different than the applicant in the convention country: the applicant in the convention country may sign herein below or applicant in India may upload the assignment from the applicant in the convention country or enclose the said assignment with this application for patent or send the assignment by post/electronic transmission duly authenticated within the prescribed period)

I/~~We~~, the applicant(s) in the convention country declare that the applicant(s) herein is/~~are my/our~~ assignee or legal representative. : **N.A.**

(iii) Declaration by the applicants:

We, the applicants hereby declare that: -

1. We are in possession of the above-mentioned invention.
2. The **Complete Specification** relating to the invention is filed with this application.

3. The invention as disclosed in the specification uses the biological material from India and the necessary permission from the competent authority shall be submitted by us before the grant of patent to us: **N.A.**
4. There is no lawful ground of objection to the grant of the patent to me/ us.
5. We are the assignees or legal representatives of true and first inventors:
6. The application or each of the applications, particulars of which are given in Para – 8 was the first application in convention country/countries in respect of our invention: **N.A.**
7. We claim the priority from the above mentioned application filed in convention country/countries and state that no application for protection in respect of the invention had been made in a convention country before that date by us or by any person from which we derive the title: **YES**
8. Our application in India is based on international application under Patent Cooperation Treaty (PCT) as mentioned in Para-9: **N.A.**
9. The application is divided out of our application particulars of which is given in Para-10 and prays that this application may be treated as deemed to have been filed on N.A. Under sec.16 of the Act: **N.A.**
10. The said invention is an improvement in or modification of the invention particulars of which are given in Para-11: **N.A.**

FOLLOWING ARE THE ATTACHMENTS WITH THE APPLICATION

(a) Form 2

Item	Details	Fee	Remarks
Complete specification) #	No. of pages:		
Claim(s)	No. of claims: No. of pages:		
Abstract	No. of pages:		
Drawing(s)	No. of drawings: No. of pages:		

In case of a complete specification, if the applicant desires to adopt the drawings filed with his provisional specification as the drawings or part of the drawings for the complete specification under rule 13(4), the number of such pages filed with the provisional specification are required to be mentioned here.

1. Complete specification (in conformation with the international application)/as amended before the International Preliminary Examination Authority (IPEA), as applicable (2 copies) **N.A**
2. Sequence listing in electronic form **N.A**
3. Drawings (in conformation with the international application)/as amended before the International Preliminary Examination Authority (IPEA), as applicable (2 copies) **N.A**

4. Form 1, 2, 26-1750Rs.
5. Statement and Undertaking on Form-3
6. Declaration of Inventor ship on Form-5
7. Request for Publication Form9 -2750.Rs
8. Form18 Examination Request, 4400.Rs
9. Other Form according to need: 880,1600,2500,4000, ...
10. Power of Authority
11. Other from 4 to 31 According to needed can fill.

Total fee Rs /- **in Cash/ Banker's Cheque /Bank Draft bearing No.....**

Date.....on..... Bank.

We hereby declare that to the best of my knowledge, information and belief the facts and matters stated herein are correct and I request that a patent may be granted to me for the said invention.

Date: 2.5.2020

DR. K. RAMKUMAR (ASSOCIATE DEAN (E & T) & PROFESSOR)

DR. S. A. KALAISELVAN (PROFESSOR)

MRS. P. MANJULA (ASSISTANT PROFESSOR)

DR. R. RAJA (ASSOCIATE PROFESSOR)

DR. G. GOVINDA RAJULU (PROFESSOR)

PAKALAPATI MANIKYA PRASUNA (ASSOCIATE PROFESSOR)

To,
The Controller of Patent,
The Patent Office, at

FORM 2
THE PATENT ACT 1970 &
 The Patents Rules, 2003
COMPLETE SPECIFICATION
 (See section 10 and rule 13)

1. TITLE OF THE INVENTION:

MDD-Data Systematic Analysis: METADATA AND DATASET DISCOVERY UNING SYSTEMATIC ANALYSIS

Name	Nationality	Address
DR. K. RAMKUMAR (ASSOCIATE DEAN (E & T) & PROFESSOR)	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. SRM UNIVERSITY, DELHI NCR, HARYANA-131029, INDIA. E-MAIL: ramkumar1975@gmail.com
DR. S. A. KALAISELVAN (PROFESSOR)	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD, TELANGANA- 500014, INDIA. E-MAIL: kalaiselvanresearch@gmail.com
MRS. P. MANJULA (ASSISTANT PROFESSOR)	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. SRM UNIVERSITY, DELHI NCR, HARYANA-131029, INDIA. E-MAIL : gpmanjula28@gmail.com
DR. R. RAJA (ASSOCIATE PROFESSOR)	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD, TELANGANA- 500014, INDIA. E-MAIL: nsraja1984@gmail.com
DR. G. GOVINDA RAJULU (PROFESSOR)	AN INDIAN NATIONAL	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING. ST. MARTIN'S ENGINEERING COLLEGE, KOMPALLY, SECUNDERABAD, TELANGANA- 500014, INDIA. E-MAIL: rajulug7@gmail.com
PAKALAPATI MANIKYA PRASUNA (ASSOCIATE PROFESSOR)	AN INDIAN NATIONAL	DEPT. OF COMPUTER SCIENCE AND ENGINEERING. ANDHRA LAYOLA INSTITUTE OF ENGINEERING AND TECHNOLOGY, VIJAYAWADA-520008 ANDHRAPRADESH, INDIA E-Mail: prasunamanikya@gmail.com

REAMBLE TO THE DESCRIPTION

PROVISIONAL	COMPLETE
The following specification describes the	The following specification Invention. Particularly describes the invention and the manner in which it is to be performed.

FIELD OF THE INVENTION

My Invention “**MDD-Data Systematic Analysis**” is related to (Computer Engineering) Meta data, data science, cloud science and data set techniques for discovering datasets for use in data systematic analytics.

BACKGROUND OF THE INVENTION

The invention relates generally to computerized analytics, and more specifically, to integrating, deploying, and monitoring of analytics solutions. In attempting to model complex systems and perform predicting, planning and budgeting for municipal development, a number of factors must be considered. In order to develop scenarios to seek optimized solutions for various constraints, such as time, location, and funding, various analytic tools can be used. Asset optimization at a city-level can also improve cost-to-performance ratios of large asset bases like water systems, grid systems, and transportation infrastructures such as rail and bus lines. To better leverage existing and planned infrastructure, planners need robust models and access to a variety of analytics.

As a result, the massive amounts of data in big data sets may be stored in numerous different data storage formats in various locations to service diverse application parameters and use case parameters. Data variables resulting from complex data transformations (e.g., model scores, risk metrics, etc.) may be central to deriving valuable insight from data driven operation pipelines. Many of the various data storage formats use transformations to convert input data into output variables. These transformations are typically hard coded into systems. As a result, retroactively determining the evolution of individual variables may be difficult, as retracing the layers of transformations for a given variable may be difficult and time consuming. Some of the output data may also contain and/or be derived from personally identifying information. Access to such data may be restricted and layers of derivation may make tracking such data difficult. Furthermore, duplicative output data is frequently generated. Duplicative output data may be generated using processing and storage resources, but the duplicative data may be difficult to detect and prevent.

Data science typically refers to the science that incorporates various disciplines including, but not limited to, data engineering, mathematics, statistics, computing, and domain-specific expertise. A data scientist thus is one who practices some or all aspects of data science in attempting to solve complex data problems. Data analytics is one aspect of data science. Conventional data analytics solutions are becoming more and more limited due to the increasing sizes and varieties of data

sets that such solutions are applied against. For example, such limitations include the lack of ability to determine which datasets among the increasing sizes and varieties of data sets are relevant to solutions of any given complex data problems.

OBJECTIVE OF THE INVENTION

1. The objective of the invention is to A system may receive a request to derive an output double variable from a source double variable.
2. The other objective of the invention is to The request may include predefine proposed logic to derive the output double variable from the source double variable.
3. The other objective of the invention is to The compare predefine proposed logic to existing logic to determine the predefine proposed logic is new.
4. The other objective of the invention is to The system may further schedule the transformation code for execution at a predetermined time.
5. The other objective of the invention is to execute the transformation code to generate data for the output double variable.
6. The other objective of the invention is to finding the appropriate datasets for a given analysis or experiment can be one of the most challenging aspects of a data science, data mining, cloud science invention.
7. The other objective of the invention is to Having pre-computed the data lifecycle via the transaction/analytic.
8. The other objective of the invention is to A plurality of datasets is identified.
9. The other objective of the invention is to One or more datasets in the plurality of datasets that are relevant to the at least one hypothesis is discovered.
10. The other objective of the invention is to work package defines at least one hypothesis associated with a given data problem, and is generated in accordance with one or more phases of an automated data analytics lifecycle.

SUMMARY OF THE INVENTION

A system, method, and computer readable medium (collectively, the “system”) is disclosed for managing data transformation and derivation in a big data environment. The system may receive a request to derive an output variable from a source variable. The request may include proposed logic to derive the output variable from the source variable. The system may then compare the proposed logic to existing logic to determine the proposed logic is new. In response to the proposed logic being new, the system may generate transformation code configured to execute the proposed logic. The system may further schedule the

transformation code for execution at a predetermined time, and then execute the transformation code to generate data for the output variable.

In various embodiments, the system may generate metadata that describes the proposed logic, and it may also look up metadata that describes the existing logic in a metadata store. The system may then compare the metadata of the proposed logic to the metadata of the existing logic. In various embodiments, the request to derive the output variable may include a requested execution time to execute the transformation code. The system may also store metadata describing the proposed logic in a metadata store in response to the proposed logic being new. The transformation platform may run on a distributed file system. The system may further store the data generated for the output variable in a big data management system. At least one of the output variable, a location of the output variable, or a copy of the output variable may be returned in response to executing the transformation code.

The forgoing features and elements may be combined in various combinations without exclusivity, unless expressly indicated herein otherwise. These features and elements as well as the operation of the disclosed embodiments will become more apparent in light of the following description and accompanying drawings.

The computer program product is provided which comprises a processor-readable storage medium having encoded therein executable code of one or more software programs. The one or more software programs when executed by one or more processing elements of a computing system implement steps of the above-described method. An apparatus comprises a memory and a processor operatively coupled to the memory and configured to perform steps of the above-described method.

Advantageously, illustrative embodiments described herein enable business users and data scientists to leverage methodologies that catalog and describe datasets to support hypothesis tests within a work package that is created to automate a data analytics lifecycle. It is realized here that finding the appropriate datasets for a given analysis or experiment can be one of the most challenging aspects of a data science project. Illustrative embodiments overcome these and other challenges. These and other features and advantages of the invention will become more readily apparent from the accompanying drawings and the following detailed description.

The detailed description of various embodiments herein makes reference to the accompanying drawings and pictures, which show various embodiments by way of illustration. While these various embodiments are described in sufficient detail

to enable those skilled in the art to practice the disclosure, it should be understood that other embodiments may be realized and that logical and mechanical changes may be made without departing from the spirit and scope of the disclosure. Thus, the detailed description herein is presented for purposes of illustration only and not of limitation. For example, the steps recited in any of the method or process descriptions may be executed in any order and are not limited to the order presented. Moreover, any of the functions or steps may be outsourced to or performed by one or more third parties. Furthermore, any reference to singular includes plural embodiments, and any reference to more than one component may include a singular embodiment.

As used herein, “big data” may refer to partially or fully structured, semi-structured, or unstructured data sets including hundreds of thousands of columns and records. A big data set may be compiled, for example, from a history of purchase transactions over time, from web registrations, from social media, from records of charge (ROC), from summaries of charges (SOC), from internal data, and/or from other suitable sources. Big data sets may be compiled with or without descriptive metadata such as column types, counts, percentiles, and/or other interpretive-aid data points. The big data sets may be stored in various big-data storage formats containing millions of records (i.e., rows) and numerous variables (i.e., columns) for each record.

BRIEF DESCRIPTION OF THE DIAGRAM

FIG. 1: illustrates an exemplary system for storing, reading, and writing big data sets, in accordance with various embodiments.

FIG. 2: illustrates an exemplary big data management system supporting a unified, virtualized interface for multiple data storage formats, in accordance with various embodiments.

FIG. 3 illustrates a data analytics lifecycle automation and provisioning system, in accordance with one embodiment of the invention.

FIG. 3-A: illustrates an exemplary data flow from source data to output data, in accordance with various embodiments.

FIG. 4: illustrates an exemplary logic map for converting source variables to output variables, in accordance with various embodiments.

FIG. 4-A: illustrates a data analytics lifecycle automation and provisioning methodology, in accordance with one embodiment of the invention.

FIG. 5: illustrates an exemplary system for evaluating, generating, scheduling, and/or executing transformation logic, in accordance with various embodiments.

FIG. 5-A illustrates a dataset discovery engine and methodology, in accordance with one embodiment of the invention.

DESCRIPTION OF THE INVENTION

FIG. 1: a distributed file system (DFS) 100 is shown, in accordance with various embodiments. DFS 100 comprises a distributed computing cluster 102 configured for parallel processing and storage. Distributed computing cluster 102 may comprise a plurality of nodes 104 in electronic communication with each of the other nodes, as well as a control node 106. Processing tasks may be split among the nodes of distributed computing cluster 102 to improve throughput and enhance storage capacity. Distributed computing cluster may be, for example, a Hadoop® cluster configured to process and store big data sets with some of nodes 104 comprising a distributed storage system and some of nodes 104 comprising a distributed processing system. In that regard, distributed computing cluster 102 may be configured to support a Hadoop® distributed file system (HDFS) as specified by the Apache Software Foundation at <http://hadoop.apache.org/docs/>.

In various embodiments, nodes 104, control node 106, and client 110 may comprise any devices capable of receiving and/or processing an electronic message via network 112 and/or network 114. For example, nodes 104 may take the form of a computer or processor, or a set of computers/processors, such as a system of rack-mounted servers. However, other types of computing units or systems may be used, including laptops, notebooks, hand held computers, personal digital assistants, cellular phones, smart phones (e.g., iPhone®, BlackBerry®, Android®, etc.) tablets, wearables (e.g., smart watches and smart glasses), or any other device capable of receiving data over the network.

In various embodiments, client 110 may submit requests to control node 106. Control node 106 may distribute the tasks among nodes 104 for processing to complete the job intelligently. Control node 106 may thus limit network traffic and enhance the speed at which incoming data is processed. In that regard, client 110 may be a separate machine from distributed computing cluster 102 in electronic communication with distributed computing cluster 102 via network 112. A network may be any suitable electronic link capable of carrying communication between two or more computing devices. For example, network 112 may be local area network using TCP/IP communication or wide area network using communication over the Internet. Nodes 104 and control

node 106 may similarly be in communication with one another over network 114. Network 114 may be an internal network isolated from the Internet and client 110, or, network 114 may comprise an external connection to enable direct electronic communication with client 110 and the internet.

A network may be unsecure. Thus, communication over the network may utilize data encryption. Encryption may be performed by way of any of the techniques now available in the art or which may become available—e.g., Twofish, RSA, El Gamal, Schorr signature, DSA, PGP, PKI, GPG (GnuPG), and symmetric and asymmetric cryptography systems. In various embodiments, DFS 100 may process hundreds of thousands of records from a single data source. DFS 100 may also ingest data from hundreds of data sources. The data may be processed through data transformations to generate output variables from input variables. In that regard, input variables may be mapped to output variables by applying data transformations to the input variables and intermediate variables generated from the input values. Nodes 104 may process the data in parallel to expedite the processing. Furthermore, the transformation and intake of data as disclosed below may be carried out in memory on nodes 104. For example, in response to receiving a source data file of 100,000 records, a system with 100 nodes 104 may distribute the task of processing 1,000 records to each node 104 for batch processing. Each node 104 may then process the stream of 1,000 records while maintaining the resultant data in memory until the batch is complete for batch processing jobs. The results may be written, augmented, logged, and written to disk for subsequent retrieval. The results may be written to disks using various big data storage formats.

FIG. 2: an exemplary architecture of a big data management system (BDMS) 200 is shown, in accordance with various embodiments. BDMS 200 may be similar to or identical to DFS 100 of FIG. 1, for example. DFS 202 may serve as the physical storage medium for the various data storage formats 201 of DFS 202. A non-relational database 204 may be maintained on DFS 202. For example, non-relational database 204 may comprise an HBase™ storage format that provides random, real time read and/or write access to data, as described and made available by the Apache Software Foundation at <http://hbase.apache.org/>.

In various embodiments, a search platform 206 may be maintained on DFS 202. Search platform 206 may provide distributed indexing and load balancing to support fast and reliable search results. For example, search platform 206 may comprise a Solr® search platform as described and made available by the Apache Software Foundation at <http://lucene.apache.org/solr/>. In various embodiments, a data warehouse 214 such as Hive® may be maintained on DFS 202. The data

warehouse 214 may support data summarization, query, and analysis of warehoused data. For example, data warehouse 214 may be a Hive® data warehouse built on Hadoop® infrastructure. A data analysis framework 210 may also be built on DFS 202 to provide data analysis tools on the distributed system. Data analysis framework 210 may include an analysis runtime environment and an interface syntax such similar to those offered in the Pig platform as described and made available by the Apache Software Foundation at <https://pig.apache.org/>.

In various embodiments, a cluster computing engine 212 for high-speed, large-scale data processing may also be built on DFS 202. For example, cluster computing engine 212 may comprise an Apache Spark™ computing framework running on DFS 202. DFS 202 may further support a MapReduce layer 216 for processing big data sets in a parallel, distributed manner to produce records for data storage formats 201. For example, MapReduce layer 216 may be a Hadoop® MapReduce framework distributed with the Hadoop® HDFS as specified by the Apache Software Foundation at <http://hadoop.apache.org/docs/>. The cluster computing engine 212 and MapReduce layer 216 may ingest data for processing, transformation, and storage in data storage formats 201 using the distributed processing and storage capabilities of DFS 202.

In various embodiments, DFS 202 may also support a table and storage management layer 208 such as, for example, an HCatalog installation. Table and storage management layer 208 may provide an interface for reading and writing data for multiple related storage formats. Continuing with the above example, an HCatalog installation may provide an interface for one or more of the interrelated technologies described above such as, for example, Hive®, Pig, Spark®, and Hadoop® MapReduce.

In various embodiments, DFS 202 may also include various other data storage formats 218. Other data storage formats 218 may have various interface languages with varying syntax to read and/or write data. In fact, each of the above disclosed storage formats may vary in query syntax and interface techniques. Virtualized database structure 220 may provide a uniform, integrated user experience by offering users a single interface point for the various different data storage formats 201 maintained on DFS 202. Virtualized database structure 220 may be a software and/or hardware layer that makes the underlying data storage formats 201 transparent to client 222 by providing variables on request. Client 222 may request and access data by requesting variables from virtualized database structure 220. Virtualized database

structure 220 may then access the variables using the various interfaces of the various data storage formats 201 and return the variables to client 222.

In various embodiments, the data stored using various of the above disclosed data storage formats 201 may be stored across data storage formats 201 and accessed at a single point through virtualized database structure 220. The variables accessible through virtualized database structure 220 may be similar to a column in a table of a traditional RDBMS. That is, the variables identify data fields available in the various data storage formats 201. In various embodiments, variables may be stored in a single one of the data storage formats 201 or replicated across numerous data storage formats 201 to support different access characteristics. Virtualized database structure 220 may comprise a catalog of the various variables available in the various data storage formats 201. The cataloged variables enable BDMS 200 to identify and locate variables stored across different data storage formats 201 on DFS 202. Variables may be stored in at least one storage format on DFS 202 and may be replicated to multiple storage formats on DFS 202. The catalog of virtualized database structure 220 may thus track the location of a variable available in multiple storage formats.

The variables may be cataloged as they are ingested and stored using data storage formats 201. The catalog may track the location of variables by identifying the storage format, the table, and/or the variable name for each variable available through virtualized database structure 220. The catalog may also include metadata describing what the variables are and where the variables came from such as data type, original source variables, timestamp, access restrictions, sensitivity of the data, and/or other descriptive metadata. For example, internal data and/or personally identifying information (PII) may be flagged as sensitive data subject to access restrictions by metadata corresponding to the variable containing the internal data and/or PII. Metadata may be copied from the data storage formats 201 or generated separately for virtualized database structure 220.

In various embodiments, virtualized database structure 220 may provide a single, unified, and virtualized data storage format that catalogues accessible variables and provides a single access point for records stored on data storage formats 201. Client 222 (which may operate using similar hardware and software to client may access data stored in various data storage formats 201 via the virtualized database structure 220. In that regard, virtualized database structure 220 may be a single access point for data stored across the various data storage formats 201 on DFS 202.

In various embodiments, virtualized database structure 220 may store and maintain the catalog of variables including locations and descriptive metadata, but virtualized database structure 220 may not store the actual data contained in each variable. The data that fills the variables may be stored on DFS 202 using data storage formats 201. Virtualized database structure 220 may enable read and write access to the data stored in data storage formats 201 without a client system having knowledge of the underlying data storage formats 201. The data stored in data storage formats 201 may be generated and/or ingested by applying a series of transformations to input data using DFS 100. The transformations may comprise a series of logical steps to alter some or all of the source data.

FIG. 3-A: a flow chart 300 for transforming source data 302 into output 310 is shown, in accordance with various embodiments. Source data 302 may comprise a one or more raw data files such as, for example, a delimited flat file, an XML file, a database file, a table, or any other structured, semi-structured or unstructured data format. Source data 302 may include a plurality of records with each record containing data. The data in the records may be separated into fields with each field being a source variable. Source data may have transformations 304 applied in the form of logic 306.

In various embodiments, logic 306 may be a series of ordered processing steps to modify source data and generate intermediate variable values and/or output variable values. For example, logic 306 may include data formatting steps such as stripping white space and truncating numbers to a predetermined length. Logic 306 may also include evaluation steps that execute an action against the data or generate a transformed value 308 for an intermediate variable or output variable in response to evaluation of a logical statement against a value of the source variable. Transformed values 308 may be augmented and written into an output 310 such as a load file for loading into a big data storage format. For example, logical steps may identify and copy a zip code from a complete mailing address and write the zip code value into a zip code variable.

FIG.4: a logic map 400 is shown in a graphical form depicting transformations 304 applied to source data 302 at a variable (e.g., column of a table) level, in accordance with various embodiments. A user may request an output variable by using a graphical tool to generate logic maps for the output variable. A user may also write a program that interfaces with a BDMS 200 to read and write data according to the transformations. As shown in logic map 400, source variable 1 is mapped directly to output 1 by a transformation 410. The transformation 410 may modify the data in source variable 1 or preserve the

original data in source variable 1 for writing into an output file. Thus, output 1 may originate from source variable 1 and transformation 410.

In various embodiments, output variables 408 may originate from multiple source variables 402. For example, as illustrated, source variable 2 is transformed into derived variable 1, source variable 3 is transformed into derived variable 2, derived variable 1 and derived variable 2 are both transformed into derived variable 5, and derived variable 5 is transformed into output 2. Thus, output variables 408 are derived from source variables 402 and derived variables 404 by applying transformations. The source variables 402, derived variables 404, and transformations 410 applied to the source variables 402 may be used to compare the origin of output variables and determine whether the output variables are duplicative of existing output variables.

FIG. 5: system 500 for transforming data in a big data environment is shown, in accordance with various embodiments. System 500 may facilitate user 502 requests for output variables, output tables, and/or output files by generating the requested output. User 502 may submit a request to transformation platform 504. The request may be in the form of a text query and/or a submission from a graphical tool. The request may contain proposed logic for transforming source data into output data.

In various embodiments, transformation platform 504 may be a software and/or hardware system configured to perform logic evaluation 505, code generation 506, and schedule and execute 508 the resulting code. In response to receiving a request for an output variable from user 502, transformation platform 504 of system 500 may evaluate the proposed logic that the user submitted to generate the output variable. Logic evaluation 505 may include comparing the proposed logic to existing logic to determine whether the logic is duplicative.

In various embodiments, the logic may be prepared for comparison in a deterministic manner to enable one to one comparison between logic. For example, transformation platform 504 may generate metadata describing the requested transformation to derive an output variable. Transformation platform 504 may access metadata describing existing transformations in metadata store 514. Storing metadata describing transformations may enable logic comparison without manually evaluating each existing transformation in response to each request for a new output variable. The metadata for the requested transformation may be compared to the metadata describing existing transformations.

In various embodiments, transformation platform 504 may deny the request for a new transformation in response to the system detecting that a data transformation exists with the existing logic of the existing data transformation matching the proposed logic. Instead, transformation platform 504 may return the location of the existing transformation results, a copy of the existing transformation results, and/or the actual existing transformation results. In that regard, transformation platform 504 may reduce processing and storage space allotted to duplicative transformation tasks.

In various embodiments, transformation platform 504 may move on to code generation 506, if the proposed transformation passes logic evaluation 505. Transformation platform 504 may dynamically generate code in response to the proposed logic and/or transformation being new (i.e., not matching the existing logic or existing transformations). The machine generated code produced by transformation platform 504 may be an executable code segment that processes source data to produce the requested output in response to execution. Transformation platform 504 may automatically generate the code to perform the proposed logical steps received from user 502 and produce the requested output variable.

In various embodiments, after code generation 506, transformation platform 504 may schedule and execute 508 the automatically generated code. Transformation platform 504 may receive the code and determine when the code can and/or should run. Transformation platform 504 may analyze existing transformation tasks that are scheduled and available processing power on DFS 100 to execute the task to determine when code should execute. User 502 may submit a desired execution schedule with the request for an output. For example, the user may specify that the output variable should be run daily, weekly, monthly, hourly, one time when available, one time immediately, etc. Output data 510 is then produced by execution of the dynamically generated code.

In various embodiments, output data 510 may be stored in a one or more data storage formats of BDMS 200, as disclosed above. Transformation platform 504 may also generate metadata for storage in metadata store 514. The metadata may describe the newly generated output data 510 and/or the transformation used to generate the output data 510. Put another way, metadata may describe what the new output variables are and where the output variables came from. For example, metadata may include a data type, original source variables, logic used to generate the variables, timestamp, access restrictions, sensitivity of the data, and/or other descriptive metadata. The metadata may be used in logic evaluation 505, for example, to identify duplicative transformations

and output variables. The metadata may also be used by BDMS 200 as disclosed above to locate data in various data storage formats.

In various embodiments, existing data 516 may also be processed for analysis 518 resulting in metadata generation and updates. In that regard, metadata store 514 may be maintained to keep metadata up-to-date and accurate. Metadata store 514 may thus be a central location to reference metadata for transformations and existing data. BDMS 200 may use metadata store 514 to identify and locate existing data as disclosed in greater detail above.

FIG. 3 depicts a data analytics lifecycle automation and provisioning system 300 that allows a data scientist 301 (or some other user or users, e.g., business user) to design and generate a provisioned system 320 that can be used to analyze and otherwise process data associated with a given complex data problem. As shown, system 300 includes a graphical user interface 302, a discovery module 304, a data preparation module 306, a model planning module 308, a model building module 310, a results communication module 312, an operationalizing module 314, and one or more work packages 316. Note that the components of system 300 in FIG. 3 may be implemented on a single computing system, or one or more components of system 300 may be implemented in a distributed computing system, e.g.,

The graphical user interface (GUI) 302 is the interface(s) through which the data scientist 301 interacts (e.g., enters data, responses, queries to one or more modules, and receives data, results, and other output generated by one or more modules) with system 300. It is to be understood that the interface used to interact with system 300 does not necessarily have to be a graphical user interface, but rather could be through command lines or some other form of input/output. As such, embodiments of the invention are not limited to any particular form of user interface.

Note that the six modules of the system 300 respectively correspond to the phases of a data analytics lifecycle (DAL). FIG. 4-A depicts the six phases of a DAL 402, according to one embodiment of the invention, including: a discovery phase 404, a data preparation phase 406, a model planning phase 408, a model building phase 410, a results communication phase 412, and an operationalizing phase 414. Each component of the system 300 assists the data scientist 301 in generating work package 316 that is used to provision the actual analytics system (provisioned system 320) that addresses the given complex data problem.

A description of each DAL phase will now be given with an exemplary problem for which the system 320 is being designed and provisioned. In this example, the

problem is a business problem. More specifically, and by way of example only, the business problem is assumed to be the task of accelerating innovation in a global technology corporation. Three aspects of this problem may be: (a) the tracking of knowledge growth throughout the global employee base of the corporation; (b) ensuring that this knowledge is effectively transferred within the corporation; and (c) effectively converting this knowledge into corporate assets. Developing an analytics system (320 in FIG. 3) that executes on these three aspects more effectively should accelerate innovation, which will thus improve the viability of the corporation. Thus, the task of system 300 is to develop such an analytics system. Of course, it is to be understood that this corporate innovation acceleration problem is just one of a myriad of examples of complex data problems that system 300 using DAL 402 can be used to address.

Discovery Phase 404 (Performed by Module 304 in System 300):

In the discovery phase, the data scientist develops an initial analytic plan. The analytic plan lays the foundation for all of the work in the analytic project being developed to address the business problem. That is, the analytic plan assists the data scientist 301 in identifying the business problem, a set of hypotheses, the data set, and a preliminary plan for the creation of algorithms that can prove or disprove the hypotheses. By way of example only, in the corporate innovation acceleration problem mentioned above, one hypothesis identified by the user as part of the analytic plan may be that an increase in geographic knowledge transfer in a global corporation improves the speed of idea delivery. This hypothesis paves the way for what data will be needed and what type of analytic methods will likely need to be used.

Data Preparation Phase 406 (Performed by Module 306 in System 300):

As the arrows in DAL 402 indicate, the six phases are iterative and interrelated/interconnected, and as such, one phase can be returned to/from one of the other phases in the process. Also, proceeding to the second phase (406) is often a matter of whether or not the data scientist is ready and comfortable sharing the analytic plan developed in the first phase (404) with his/her peers (this comfort level is reflective of the maturity of the analytic plan—if it is too rough and unformed, it will not be ready to be shared for peer review). If so, then the data preparation phase 406 can begin. That is, once the analytic plan has been delivered and socialized, the next step focuses on the data. In particular, the next step is about conditioning the data. The data must be in an acceptable shape, structure, and quality to enable the subsequent analysis.

Continuing with the corporate innovation acceleration example, assume that the type of data that the analytics project relies on falls into two categories: (i) an “idea submission” data set (essentially a large-scale database containing structured data); and (ii) a globally-distributed set of unstructured documents representing knowledge expansion within the corporation in the form of minutes and notes about innovation/research activities. It is realized that these data sets cannot be analyzed in their raw formats. In addition, it is possible that the data is not of sufficient quality. Furthermore, the data is likely inconsistent.

All of these issues suggest that a separate analytic “sandbox” must be created to run experiments on the data. The “sandbox” here refers to a separate analytics environment used to condition and experiment with the data. This sandbox is realized via data preparation module 306. On average the size of this sandbox should be roughly ten times the size of the data in question. As such, the sandbox preferably has: (i) large bandwidth and sufficient network connections; (ii) a sufficient amount of data including, but not limited to, summary data, structured/unstructured, raw data feeds, call logs, web logs, etc.; and (iii) transformations needed to assess data quality and derive statistically useful measures. Regarding transformations, it is preferred that module 306 transform the data after it is obtained, i.e., ELT (Extract, Load, Transform), as opposed to ETL (Extract, Transform, Load). However, the transformation paradigm can be ETLT (Extract, Transform, Load, transform again), in order to attempt to encapsulate both approaches of ELT and ETL. In either the ELT or ETLT case, this allows analysts to choose to transform the data (to obtain conditioned data) or use the data in its raw form (the original data). Examples of transformation tools that can be available as part of data preparation module 306 include, but are not limited to, Hadoop™ (Apache Software Foundation) for analysis, Alpine Miner™ (Alpine Data Labs) for creating analytic workflows, and R transformations for many general purpose data transformations. Of course, a variety of other tools may be part of module 306.

It is further realized that once the sandbox is created, there are three key activities that allow a data scientist to conclude whether or not the data set(s) he/she is using is sufficient:

Familiarization with the data:

The data scientist 301 lists out all the data sources and determines whether key data is available or more information is needed. This can be done by referring back to the analytic plan developed in phase 404 to determine if one has what is needed, or if more data must be loaded into the sandbox.

Perform data conditioning:

Clean and normalize the data. During this process, the data scientist 301 also discerns what to keep versus what to discard.

Survey and visualize the data:

The data scientist 301 can create overviews, zoom and filter, get details, and begin to create descriptive statistics and evaluate data quality. As will be described below in Section II, the discovery module 304 and/or the data preparation module 306 can operate in conjunction with a dataset discovery system to be described below in the context of FIG. 5-A. In an alternative embodiment, such a dataset discovery system may be implemented as part of the discovery module 304 and/or the data preparation module 306. However, as will be further explained below in the context of FIG. 5-A, such a dataset discovery system is configured to operate in conjunction with one or more of the other modules (302 through 312) of the system 300.

Model Planning Phase 408 (Performed by Module 308 in System 300):

Model planning represents the conversion of the business problem into a data definition and a potential analytic approach. A model contains the initial ideas on how to frame the business problem as an analytic challenge that can be solved quantitatively. There is a strong link between the hypotheses made in phase 404 (discovery phase) and the analytic techniques that will eventually be chosen. Model selection (part of the planning phase) can require iteration and overlap with phase 406 (data preparation). Multiple types of models are applicable to the same business problem. Selection of methods can also vary depending on the experience of the data scientist. In other cases, model selection is more strongly dictated by the problem set.

Described below are a few exemplary algorithms and approaches (but not an exhaustive list) that may be considered by the data scientist 301 in the exemplary accelerated corporate innovation hypothesis given above:

(i) Use Map/Reduce for extracting knowledge from unstructured documents. At the highest level, Map/Reduce imposes a structure on unstructured information by transforming the content into a series of key/value pairs. Map/Reduce can also be used to establish relationships between innovators/researchers discussing the knowledge.

(ii) Natural language processing (NLP) can extract “features” from documents, such as strategic research themes, and can store them into vectors.

(iii) After vectorization, several other techniques could be used:

(a) Clustering (e.g., k-means clustering) can find clusters within the data (e.g., create ‘k’ types of themes from a set of documents).

(b) Classification can be used to place documents into different categories (e.g., university visits, idea submission, internal design meeting).

(c) Regression analysis can focus on the relationship between an outcome and its input variables, and answers the question of what happens when an independent variable changes. Regression analysis can help in predicting outcomes. This could suggest where to apply resources for a given set of ideas.

(d) Graph theory (e.g., social network analysis) is a way to establish relationships between employees who are submitting ideas and/or collaborating on research.

At this point in the DAL 402, the data scientist 301 has generated some hypotheses, described potential data sets, and chosen some potential models for proving or disproving the hypotheses.

Model Building Phase 410 (Performed by Module 310 in System 300):

In the model building phase, the system experimentally runs the one or more models that the data scientist 301 selected in phase 408. The model(s) may be executed on a portion of the original (raw) data, a portion of the conditioned data (transformed in phase 406), or some combination thereof. In this phase, the initial data analytic plan is updated to form a refined data analytic plan.

For example, Map/Reduce algorithm, NLP, clustering, classification, regression analysis and/or graph theory models are executed by module 310 on a test sample of the data identified and conditioned by module 306 in phase 406 (data preparation). Here the data scientist 301 is able to determine whether the models he/she selected are robust enough (which depends on the specific domain of the data problem being addressed) and whether he/she should return to the model planning phase 408. For example, in the corporate innovation acceleration example, some portion of the data sets identified in the earlier phases (e.g., structured idea submissions and unstructured support documents) is processed with the selected models.

Results Communication Phase 412 (Performed by Module 312 in System 300):

In the results communication phase, the results of the model execution of phase 410 are reported to the data scientist 301 (via GUI 302). This phase is also where the analytic plan that was initially developed in phase 404 and fine-tuned through phases 406, 408 and 410 can be output by the system 300 (i.e., as a refined or final analytic plan). The final analytic plan at this point in the DAL 402 may be referred to as a work package (316 in FIG. 3).

Operationalizing Phase 414 (Performed by Module 314 in System 300):

Operationalizing refers to the process of actually provisioning computing resources (physical and/or virtualized) to generate the system that will be deployed to handle the analytics project in accordance with the final analytic plan, e.g., system 320 in FIG. 3. This may involve provisioning VMs and LUNs as well as other virtual and physical assets that are part of cloud infrastructure. The provisioned system will then analyze subsequent data that is obtained for the given complex data problem.

Given the detailed description of the data analytics lifecycle phases above, we now make some observations and introduce some other features and advantages of the system.

Assume that the data scientist 301 is at a later phase in the process but then realizes that he/she forgot to include some data in the discovery phase 404 that is needed to complete the analysis. Advantageously, the interrelated and iterative nature of DAL 402 and the flexibility of the system used to automate the DAL (system 300) provide the data scientist with the ability to return to the discovery phase, correct the error, and return to a subsequent stage with the results for each stage affected by the change being automatically updated.

During the model building phase 410, it is not known what resources are going to be needed, which have a specific cost, and definition of what would be included (amount of storage, number of VMs, the analytics tools needed, etc.). Being able to know the approximate cost and configuration needed would be very useful for the process of tuning the model based on cost or configuration constraints. Thus, during each phase of the DAL 402, the data scientist 301 is presented (at GUI 301) with an inventory of the current infrastructure, services, and tools needed and their approximate cost as changes are made to the parameters associated with the analysis. This will be further described below in the context of FIG. 5-A. This

allows the data scientist to remove or change the model dynamically based on resource constraints (e.g., cost or VM limits).

Once the analytics work package 316 is defined, provisioning the resources needed to most efficiently support the analysis is important. As such, embodiments of the invention automate and execute work packages for the data scientist by constructing the work package and providing resource and cost estimates throughout the DAL.

Many times, introducing new raw, source data sets into a project can have cascading effects on the size of the analytic sandbox (see data preparation phase 406 above) needed to support the analysis. Embodiments of the invention provide selectable sizing multiples to dynamically provision the system parameters, such as a storage capacity, bandwidth required, and compute power depending on the type of new data involved and its size. For example, these sizing multiples could be used between phases 404 and 406, between 406 and 408, and even between phase 408 and 410. The sizing multiples serve as a mechanism for dynamically provisioning and adjusting the size, capacity, and constraints needed for the analytic sandbox.

By way of example only, assume there is 100 GB worth of innovation data that is to be analyzed. The data preparation module 306 multiplies this value by some constant (e.g., 10 or 20 times) in order to estimate the capacity of the analytic sandbox. That is, the data scientist will take the 100 GB of data and run transformations and other experiments that will require additional amounts of capacity. Therefore, the data preparation module 306 creates a work package specification that states: “allocate 1 TB of sandbox data which has the following features” This aspect of the work package instructs cloud provisioning software to allocate appropriately.

It is also realized that privacy of data is a major concern when mining large amounts or correlating various types of data. Privacy of the individuals needs to be protected while still allowing useful analysis and presentation of the data. Embodiments of the invention provide for masking capabilities in the work package 316, as well as any data presented by the system, for the data scientist, as well as creating contextual views based on the identity of the consumer of the output. This feature is very useful, particularly in a highly regulated data environment.

Further, the privacy/masking techniques associated with the work package 316 and other data can be employed to protect the data from wholesale viewing by the data scientist or an output generated by the work package

execution. Also it is possible to create multiple views of the data based on privacy constraints tied to the context and role of the potential viewer. For example, a mid-level sales manager may have the ability to see consolidated data across the sales areas in the country, but his/her subordinates within the same area would only be allowed to see that specific area's data view as they are not authorized to see data across the country for regulatory (e.g., Security and Exchange Commission) reasons.

As a consequence of the privacy aspect, the data scientist can receive a diagnostic summary stating the resources they have access to for the analytical work they are planning to pursue. While some illustrative privacy/masking techniques have been described above, it is to be understood that alternative privacy protection controls (such as, but not limited to, privacy anonymization) can be employed in system 300.

In addition, the operationalizing module 314 can make predictions of the types of additional technology resources and tools needed to complete the analytics and move into a production environment, based on the type of analytics being undertaken. As a result, the data scientist would be notified early if they needed to request additional tools that would enable them to complete their work. This aspect of system 300 enables the data scientist to initiate funding requests earlier in the DAL, identify people if specific skill sets are needed (such as a Hadoop™ expert in addition to a mathematician), and operationalize the resources before the data modeling stages (e.g., identify this during phase 404 of the DAL, rather than in phase 410) to avoid bottlenecks in the project.

It is further realized that a work package containing a larger sized dataset will contribute to an increased cost, as provisioning will increase. Besides size, other dataset characteristics may impact cost, e.g., perhaps publicly available data is cheaper than sensitive data, which requires an anonymization service. System 300 gives the data scientist insight into which dataset characteristics would be most beneficial to the analytic plan. The system 300 can also perform what-if analysis on different identified dataset and analytic plan scenarios. This can be done in conjunction with the dataset discovery system of FIG. 5-A.

Further, it is realized that the work of all data science projects are not equal. For example, a critical project such as one directed by an officer of the company (e.g., CEO) could require higher priority and take precedence over existing work packages. Also, perhaps the CEO's work package should be executed faster than regular data scientists, thus increasing provisioning. System 300 accounts for the priority levels associated with the data scientists.

Advantageously, system 300 allows a data scientist to know ahead of execution time the execution costs. Additionally, the system is able to dynamically change system parameters as the data scientist begins to refine the data and the analysis without having to start all over again or manually de-provision or increase the provisioned resources. System 300 creates a dynamic work package that includes the parameters needed to move through the analytics lifecycle and include the automation necessary to allow the data scientist to focus on fine tuning the parameters and not on manually changing the infrastructure or data ingest process.

Dataset Discovery System:

We now turn to a description of dataset discovery according to one or more illustrative embodiments of the invention. As mentioned above, dataset discovery may be implemented in the discovery module 304 and/or the data preparation module 306 or any other module in system 300 (and combinations thereof) described above in Section I. Also, dataset discovery may alternatively be implemented as a module separate from the modules of system 300 shown in FIG. 3, e.g., as a dataset discovery module that is operatively coupled to system 300, receiving an initial, intermediate or final work package 316, and providing dataset discovery techniques for further use by system 300. One illustrative example of such a dataset discovery system or module will be described below in the context of FIG. 5-A. Alternatively, dataset discovery techniques as will be described herein can be implemented independent of and separate from system 300, and thus are not intended to be limited to any data analytics lifecycle automation and provisioning system described herein. That is, the techniques may be implemented in a standalone dataset discovery system or in some other computing system that can benefit from advantages of dataset discovery.

For example, during the discovery phase of the data analytics lifecycle, the business user may develop one or more hypotheses that can be explored and tested with data. The next challenge to arise is to determine if data exists that would enable a user to test these ideas and the state of this data, in terms of its quality and availability. Also, after completing the discovery phase of the data analytics lifecycle, business users and data scientists may have a clear idea of the hypotheses they would like to test. However, they may have a limited understanding of the data dependencies they have within an organization. Further, even if an appropriate dataset can be identified to test a particular idea, it can be very difficult to understand the provenance of a dataset, or the relationship of a specific dataset to other, related datasets within an organization. Also, many times datasets are used for a specific project, but although the people performing

the analytics project have a strong background in algorithms, they may not have expertise in data privacy. As such, personal or sensitive information about people may be shared in contexts where it should not be.

Accordingly, embodiments of the invention provide methods techniques that assist users in discovering datasets within and/or outside an enterprise for testing hypotheses whereby such data is accessible and is managed to protect privacy for a given business problem. As mentioned above, these automated techniques for discovering relevant datasets for a given hypothesis are part of the first and second phases of the data analytics lifecycle, which is used for data science projects and includes vetting and translation of a business problem into an analytical challenge that can be tested through quantitative methods. Further, embodiments of the invention enable business analysts and data scientists to leverage a set of methods that catalogue and describe an enterprise's datasets to support hypothesis tests within a work package that is created to automate the data analytics lifecycle.

FIG. 5-A: illustrates a dataset discovery engine and methodology, in accordance with one embodiment of the invention. As shown, dataset discovery engine 500 is operatively coupled to a data analytics lifecycle 510 and a work package 512. It is to be appreciated that the data analytics lifecycle 510 and the work package 512 represent, in one embodiment, data analytics lifecycle automation and provisioning system 300 whereby data analytics lifecycle 510 is automated to generate work package 512. Dataset discovery engine 500 includes a data crawling agent 502, a feature extractor 504, and a data store 506.

Data crawling agent 502, within the context of the work package 512, crawls local and wide area public and private information networks to identify, collect and catalog datasets. These datasets can be comprised of, by way of example only, text, email, images, video, voice mail, and more traditional RDBMS (relational database management system) structures. The data crawling agent 502 write log files cataloguing the data it finds, and also describe these data sources using metadata, such as type of data, size, and topic areas. Such data catalog files (data catalogs) are stored in data store 506.

Thus, for example, the data crawling agent 502 locates, in an enterprise (or outside the enterprise), relevant datasets that satisfy the hypothesis from the work package 512. These datasets can be dynamically located or located from pre-stored datasets. The data crawling agent 502 also filters the located datasets based on what the agent learns from examining and/or analyzing the work package 512. In one example, the agent filters hypothesis-relevant data from non-hypothesis-relevant data. This can be done by identifying associations between datasets in the

data catalogs. Also, provenance data can be generated and used for datasets in the data catalogs.

Feature extractor 504 extracts features from the datasets identified by the data crawling agent 502. By interpreting data attributes written to the log files that describe the enterprise's data elements, it is possible to infer features of these datasets. These features could relate to the presence of conditions or properties related to the data, such as, but not limited to, its rate of change, the presence of information that contains personally identifiable information (PI), and the type of data, both in terms of the structure (such as, for example, unstructured, quasi-structured, semi-structured, and structured) and its potential use cases (for example, revenue information that can be used for a financial analysis byproduct).When the feature extractor 504 determines the presence of private data (PII) in the identified datasets, the engine 500 can prevent use of the private data from the identified datasets. One prevention example is by masking the private data in the identified datasets.

Data store 506 stores the data catalogs generated by the data crawling agent 502 and feature extractor 504. Understanding the datasets and cataloguing them in the manners mentioned above enables an algorithm to run and use these datasets as elements within a superset. In other words, if a user wants to use Dataset A, an algorithm can identify associations of other datasets based on their metadata that are similar (e.g., people who use Dataset A, also use Dataset B). Similarly, Social Network Analysis can be run on the datasets that behave as data elements. Instead of using Social Network Analysis to identify influential people who are nodes within a social network, the data discovery techniques according to one or more embodiments can be used to identify influential datasets in various kinds of decision making, where the datasets themselves act as nodes and their properties (such as, for example, rate of change, influence on other related data sets, or provenance) are edges in a social graph. This shows which datasets influence the creation and growth of other datasets, and can also show which kinds of datasets are used to support which kinds of business decisions. In addition, the data store 506 can show the provenance of certain kinds of datasets and their elements, catalog the quality of specific datasets, and identify a level of trustworthiness of a given dataset.It is to be appreciated that the data stored in data store 506 can then be provided back to the data analytics lifecycle 510 to complete one or more other phases. For example, the hypothesis can be tested using at least a portion of the discovered datasets. The hypothesis can then be refined. The work package 512 can also be updated based on the refined hypothesis to generate a refined work package for use by the one or more phases of the automated data analytics lifecycle.

WE CLAIMS

1. My Invention "MDD-Data Systematic Analysis" is A system may receive a request to derive an output double variable from a source double variable. The request may include predefine proposed logic to derive the output double variable from the source double variable. The Invention may then compare the predefine proposed logic to existing logic to determine the predefine proposed logic is new. In response to the predefine proposed logic being new, the system may generate transformation code configured to execute the proposed logic. The system may further schedule the transformation code for execution at a predetermined time, and then execute the transformation code to generate data for the output double variable. It is realized here that finding the appropriate datasets for a given analysis or experiment can be one of the most challenging aspects of a data science, data mining, cloud science invention. The enterprise management system includes transaction/analytic applications and an archiving system in which data object lifecycles are pre-computed when the data object is created by the transaction application or analytic application. Having pre-computed the data lifecycle via the transaction/analytic applications, an archiving system need not re-determine whether the criteria for archiving are met. When the archiving system is initiated, the archiving system may identify the data objects having lifecycle dates that match the current date and archive them directly. The initial work package defines at least one hypothesis associated with a given data problem, and is generated in accordance with one or more phases of an automated data analytics lifecycle. A plurality of datasets is identified. One or more datasets in the plurality of datasets that are relevant to the at least one hypothesis are discovered. The at least one hypothesis is tested using at least a portion of the one or more discovered datasets. The method comprises the following steps. An initial work package is obtained. The initial work package defines at least one hypothesis associated with a given data problem, and is generated in accordance with one or more phases of an automated data analytics lifecycle.

2. According to claim 1# The invention is to A system may receive a request to derive an output double variable from a source double variable.

3. According to claim 1,2# The invention is to The request may include predefine proposed logic to derive the output double variable from the source double variable.

4. According to claim 1,2# The invention is to The compare predefine proposed logic to existing logic to determine the predefine proposed logic is new.

5. According to claim 1,3# The invention is to The system may further schedule the transformation code for execution at a predetermined time.

6. According to claim 1,2,5# The invention is to execute the transformation code to generate data for the output double variable.

7. According to claim 1,2# The invention is to finding the appropriate datasets for a given analysis or experiment can be one of the most challenging aspects of a data science, data mining, cloud science invention.

8. According to claim 1,2,7# The invention is to Having pre-computed the data lifecycle via the transaction/analytic.

9. According to claim 1,2# The invention is to A plurality of datasets is identified.

10. According to claim 1,2,6# The invention is to One or more datasets in the plurality of datasets that are relevant to the at least one hypothesis is discovered.

11. According to claim 1,2,7# The invention is to work package defines at least one hypothesis associated with a given data problem, and is generated in accordance with one or more phases of an automated data analytics lifecycle.

Date: 2/5/2020

DR. K. RAMKUMAR (ASSOCIATE DEAN (E & T) & PROFESSOR)

DR. S. A. KALAISELVAN (PROFESSOR)

MRS. P. MANJULA (ASSISTANT PROFESSOR)

DR. R. RAJA (ASSOCIATE PROFESSOR)

DR. G. GOVINDA RAJULU (PROFESSOR)

PAKALAPATI MANIKYA PRASUNA (ASSOCIATE PROFESSOR)

MDD-Data Systematic Analysis: METADATA AND DATASET DISCOVERY UNING SYSTEMATIC ANALYSIS

ABSTRACT

My Invention “**MDD-Data Systematic Analysis**” is A system may receive a request to derive an output double variable from a source double variable. The request may include predefine proposed logic to derive the output double variable from the source double variable. The Invention may then compare the predefine proposed logic to existing logic to determine the predefine proposed logic is new. In response to the predefine proposed logic being new, the system may generate transformation code configured to execute the proposed logic. The system may further schedule the transformation code for execution at a predetermined time, and then execute the transformation code to generate data for the output double variable. It is realized here that finding the appropriate datasets for a given analysis or experiment can be one of the most challenging aspects of a data science, data mining, cloud science invention. The enterprise management system includes transaction/analytic applications and an archiving system in which data object lifecycles are pre-computed when the data object is created by the transaction application or analytic application. Having pre-computed the data lifecycle via the transaction/analytic applications, an archiving system need not re-determine whether the criteria for archiving are met. When the archiving system is initiated, the archiving system may identify the data objects having lifecycle dates that match the current date and archive them directly. The initial work package defines at least one hypothesis associated with a given data problem, and is generated in accordance with one or more phases of an automated data analytics lifecycle. A plurality of datasets is identified. One or more datasets in the plurality of datasets that are relevant to the at least one hypothesis are discovered. The at least one hypothesis is tested using at least a portion of the one or more discovered datasets. The method comprises the following steps. An initial work package is obtained. The initial work package defines at least one hypothesis associated with a given data problem, and is generated in accordance with one or more phases of an automated data analytics lifecycle.

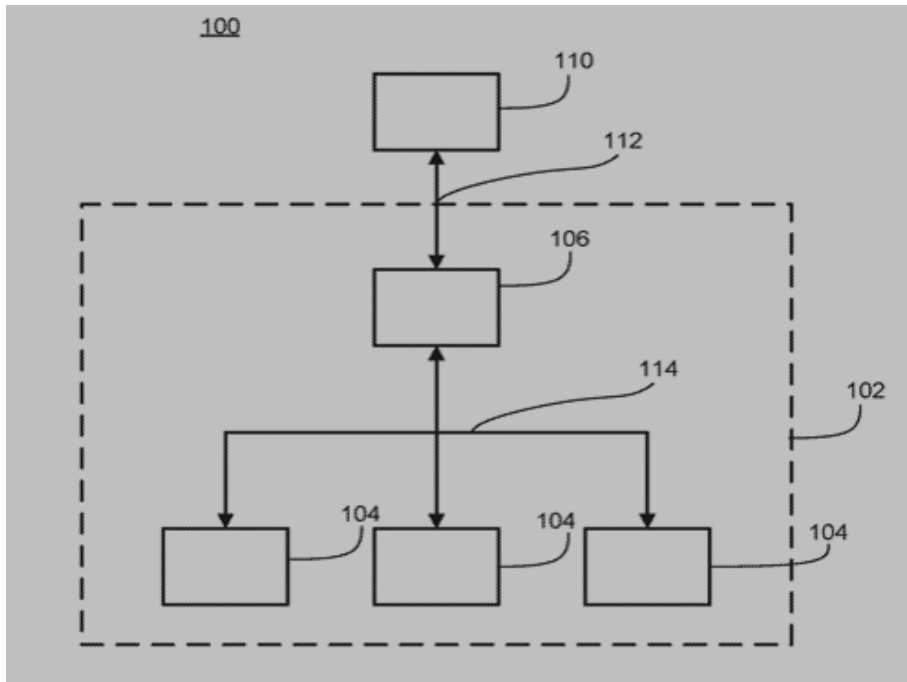


FIG. 1: ILLUSTRATES AN EXEMPLARY SYSTEM FOR STORING, READING, AND WRITING BIG DATA SETS, IN ACCORDANCE WITH VARIOUS EMBODIMENTS;

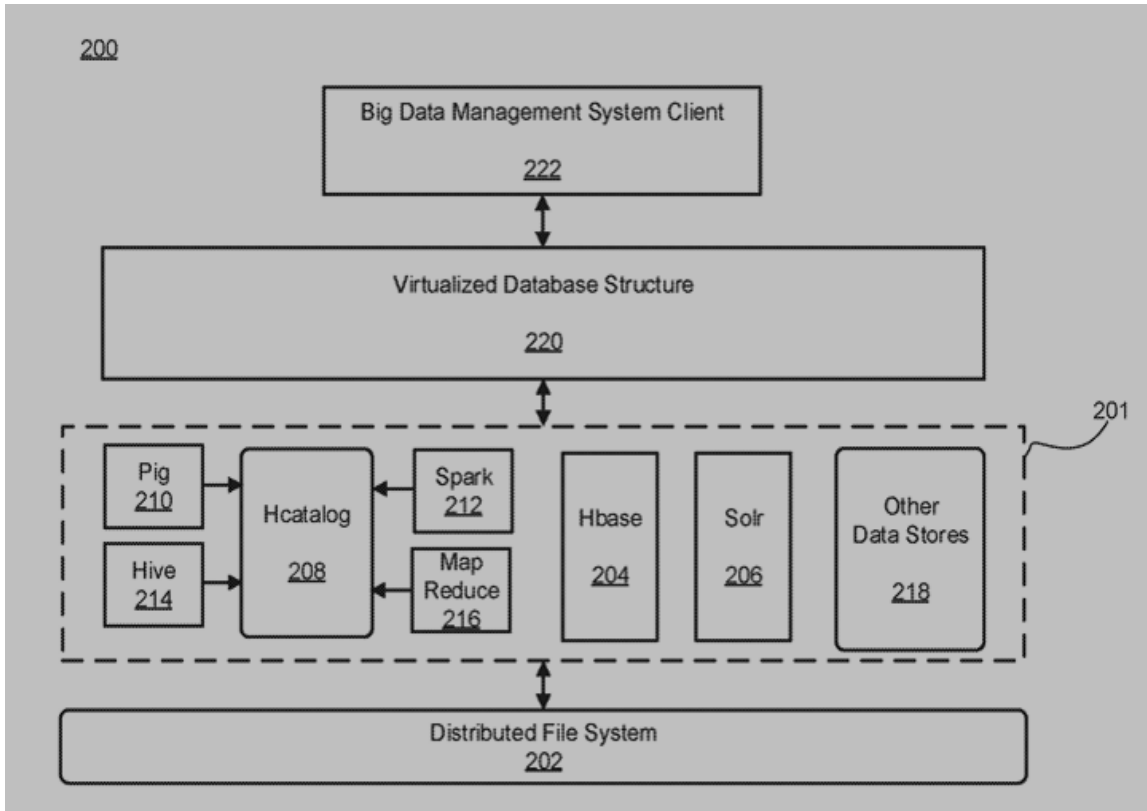


FIG. 2: ILLUSTRATES AN EXEMPLARY BIG DATA MANAGEMENT SYSTEM SUPPORTING A UNIFIED, VIRTUALIZED INTERFACE FOR MULTIPLE DATA STORAGE FORMATS, IN ACCORDANCE WITH VARIOUS EMBODIMENTS;

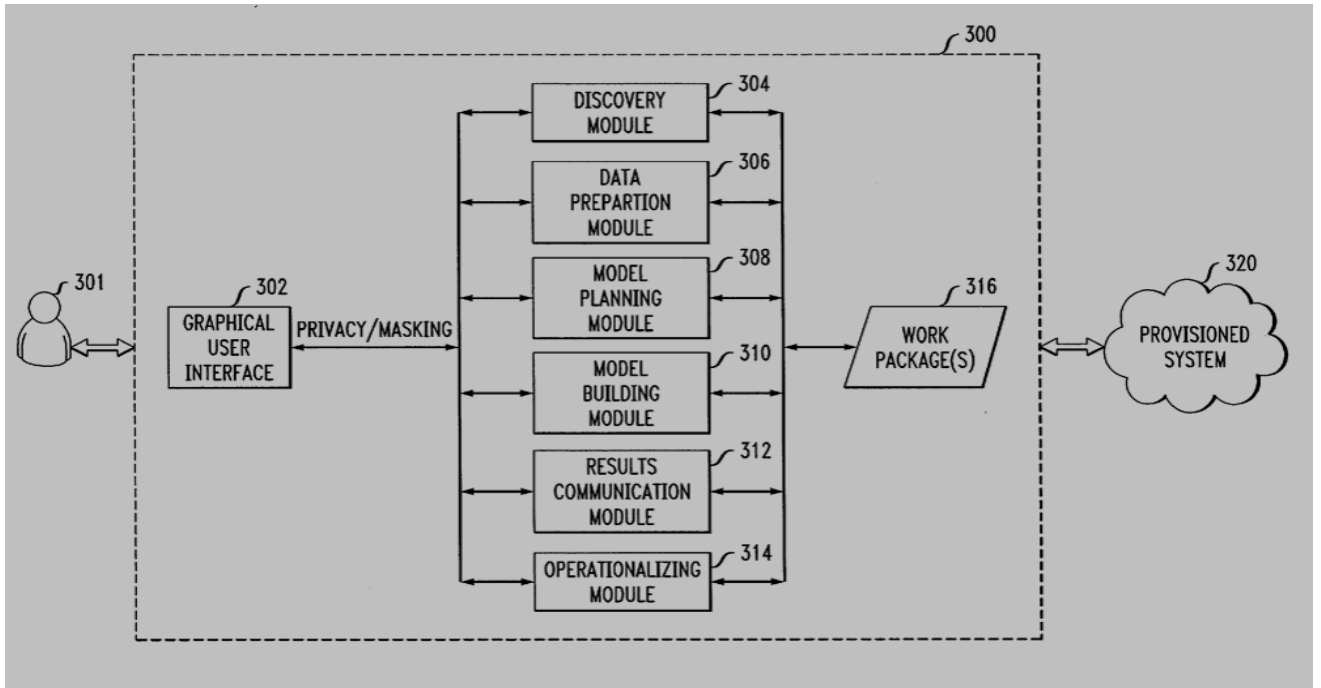


FIG. 3 ILLUSTRATES A DATA ANALYTICS LIFECYCLE AUTOMATION AND PROVISIONING SYSTEM, IN ACCORDANCE WITH ONE EMBODIMENT OF THE INVENTION.

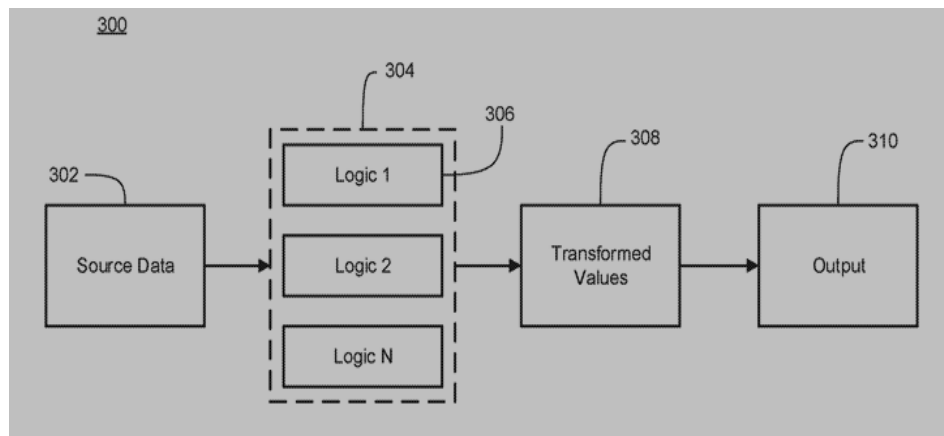


FIG. 3-A: ILLUSTRATES AN EXEMPLARY DATA FLOW FROM SOURCE DATA TO OUTPUT DATA, IN ACCORDANCE WITH VARIOUS EMBODIMENTS;

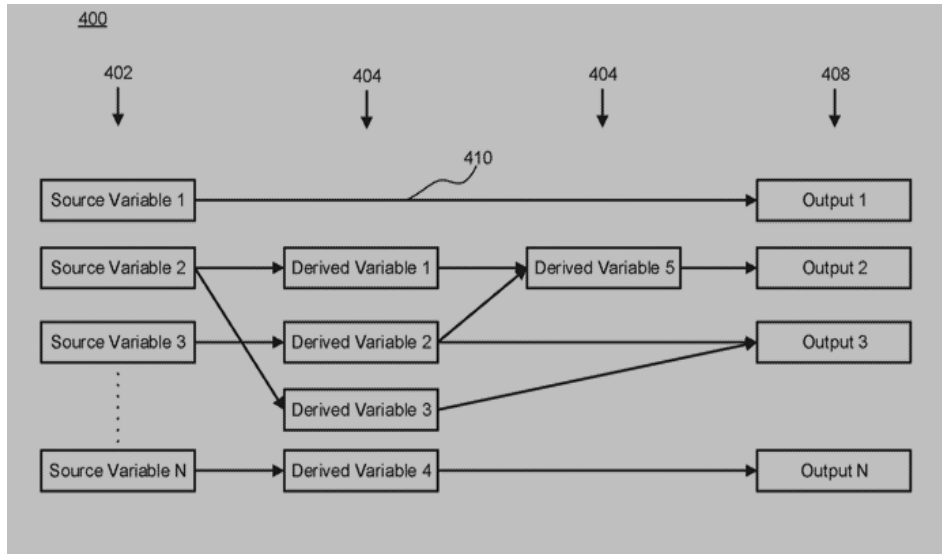


FIG. 4: ILLUSTRATES AN EXEMPLARY LOGIC MAP FOR CONVERTING SOURCE VARIABLES TO OUTPUT VARIABLES, IN ACCORDANCE WITH VARIOUS EMBODIMENTS;

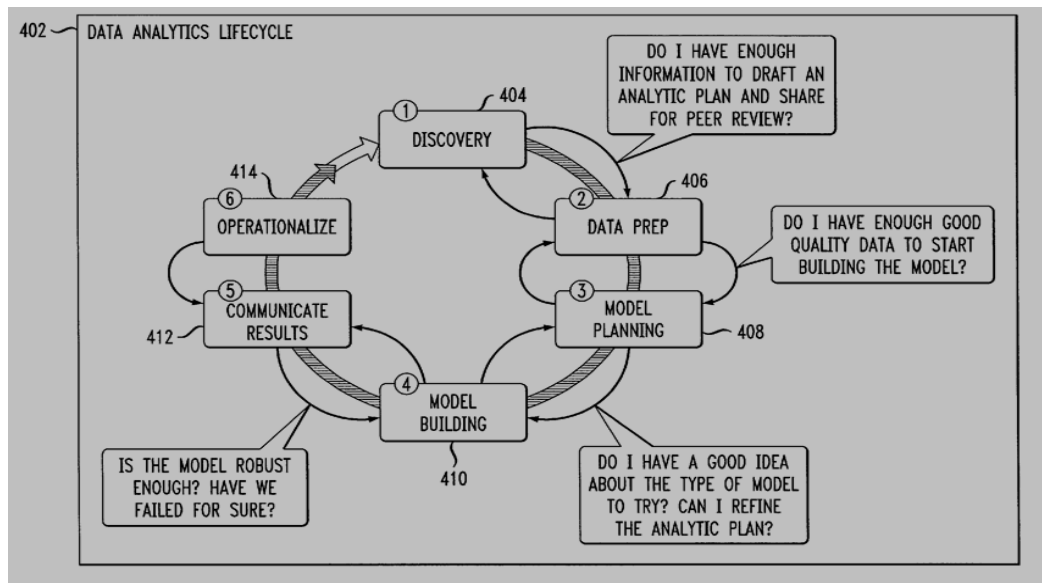


FIG. 4-A: ILLUSTRATES A DATA ANALYTICS LIFECYCLE AUTOMATION AND PROVISIONING METHODOLOGY, IN ACCORDANCE WITH ONE EMBODIMENT OF THE INVENTION.

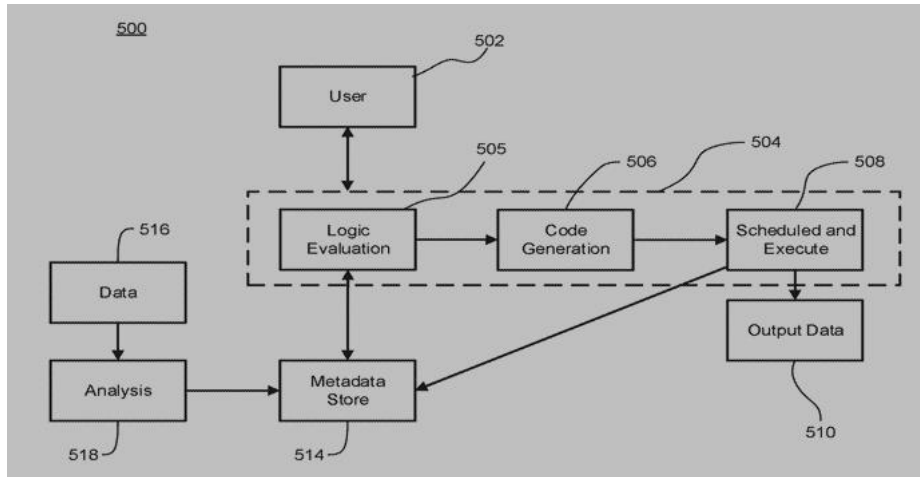


FIG. 5: ILLUSTRATES AN EXEMPLARY SYSTEM FOR EVALUATING, GENERATING, SCHEDULING, AND/OR EXECUTING TRANSFORMATION LOGIC, IN ACCORDANCE WITH VARIOUS EMBODIMENTS;

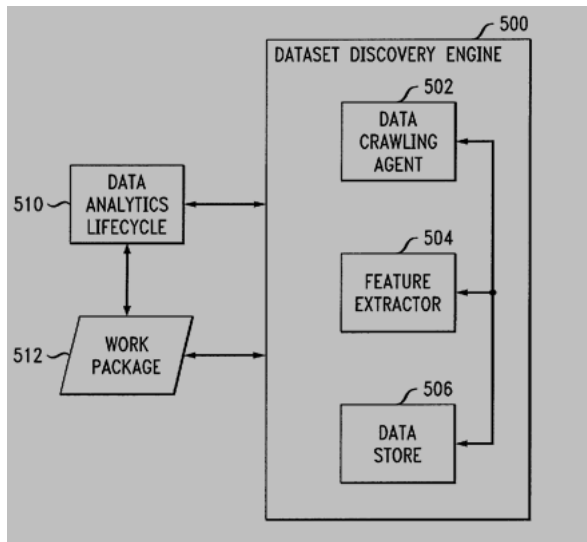


FIG. 5-A ILLUSTRATES A DATASET DISCOVERY ENGINE AND METHODOLOGY, IN ACCORDANCE WITH ONE EMBODIMENT OF THE INVENTION.