Swampiness!

Murky Healthcare Swamp evolve with Machine Learning!

Official Deadline : ???
Personal Deadline : ???

Andreas Francois VERMEULEN
University of St Andrews
School of Computer Science
St Andrews, Scotland,
United Kingdom
Email: afv@st-andrews.ac.uk

Dr Juliana Küster Filipe BOWLES
University of St Andrews
School of Computer Science
St Andrews, Scotland,
United Kingdom
Email: jkfb@st-andrews.ac.uk

Dr Vladimir JANJIC
University of St Andrews
School of Computer Science
St Andrews, Scotland,
United Kingdom
Email: vj32@st-andrews.ac.uk

***Abstract*—Data scientists are regularly faced with a mammoth task of converting raw health data from *uncurated data lakes* (or *data swamps*) into a structured format for *curated data lakes*, which can then further can undergo transformation into highly-structured *data vault*. This data vault serves as perfect format for performing the required data analytics for healthcare insights. This current process is, for the most part, performed manually and incurs significant costs in terms of man power. In this paper, we present a novel, semi-automated technique for extracting important meta-data from data swamps, allowing tracking of lineage and provenance of data in them and thus forming curated data lakes. Our technique is based on evolutionary multi-agent systems (EMAS), supported by reinforcement-learning algorithms. We also introduce a novel *coefficient of swampiness* to quantify the "purity" of data in data swamps/lakes, and we demonstrate, on examples of data sets from Smart Patient-Centric Healthcare domain, that we can reduce the swampiness of data swamps by XX%. We also demonstrate reduction in required effort for curating data lakes by XXX% and reduction in cost by XXX %.**

***Keywords*-Data Lake; Big Data; Grid Computing; Deep Learning; Evolutionary Programming; Evolutionary Algorithms; Meta-programming; evolutionary multi-agent system; Smart Patient-Centric Healthcare Systems;**

I. INTRODUCTION

With the increased digitisation of the Smart Patient-Centric Healthcare and emergence of internet-of-things systems, we are simply overwhelmed with data of various shapes and coming from various diverse healthcare sources. It is well recognised that the existing *data lakes*, which represent a standard for storing and managing data for Smart Patient-Centric Healthcare across Europe, cannot cope with increased volume, velocity, variety, variability, veracity, visualisation, and value of the data to healthcare insights transformation [1]. The combinatorial explosion [2] in processing complexity of a data lake at scale is resulting in ineffective and inefficient processing of data from the data lakes, resulting in failure to deliver the right information at the right time for good healthcare.

The natural response to the issue is to deploy more data engineers and data scientists to counteract the data deluge. This is, however, very expensive and there is already a recognised shortage of data scientists, which will only further increase with further growth in the volume of data to be processed [3].

One of the solutions to this problem is to increase *automation* of the whole process of managing data, from retrieval and storing into data lake to the final processing and analytics' insights. To allow for this, *data vaults* have emerged as an accepted standard of format of the data for data processing. However, due to variety of sources from which data is collected into lakes, the data in data lakes is very often unstructured (*data swamp*), with minimal or no available metadata to derive its format and structure. This means that significant, time consuming and expensive pre-processing might be required to extract information about the data that is required for its efficient processing.

This paper deals with the first step in structuring the data from data swamps to data vaults. Namely, we deal with *curating* data swamps, extracting the most essential meta-data (lineage and provenance) from data, thus obtaining *curated data lake* on which additional transformations can be applied to obtain a healthcare data vault. Our technique for metadata extraction is based on evolutionary computing multi-agent systems (EMASs) and uses reinforcement-learning to XX. In addition, we introduce a precise measure of the degree of structure of the data in data swamps (*coefficient of swampiness*), which allows us to characterise how "stained" the data is. We demonstrate, on an example use case from Smart Patient-Centric Healthcare domain, that we are able to *automatically* reduce swampiness of the data by XX%. Furthermore, by automating the process of data swamp curating, we were able to improve productiveness by XX% and reduce the cost in developing data analytics by XX%. The work we have done is, of course, not specific to the Smart Patient-Centric Healthcare data and can readily

be applied to data from other domains.

The specific research contributions of this paper are:

- We propose a novel technique for extracting lineage and provenance meta-data from the existing *completely unstructured* data from data swamps;
- We propose a novel meta-data model for curated data lakes, with application to Smart Patient-Centric Healthcare systems data;
- We introduce a novel metric (*coefficient of swampiness*) for measuring mathematical, how *curated* data in a data swamp is;
- We demonstrate, on an example use case from the three Smart Patient-Centric Healthcare systems data, that we are able to automatically infer lineage and provenance of data and thus reduce its swampiness.

II. Background

Modern data analysts are faced with the prevalent problem of increase in volume of data that needs to be processed and analysed. Bell noted that there is a logarithmic growth not only in volume, but also in velocity, variety, variability and value of the data [1]. As the whole process of data management, from retrieval data to the use in actual analytics, is currently mostly manual, with very little automation, there is an increasing demand for more and more data scientists. There is a noted shortage of 140,000 to 180,000 data scientists with deep analytic skills, as well as 1.5 million managers and analysts (346,000 in Europe alone) to extract useful information and perform effective decisions based on data in data lakes [4], [3]. Forbes predicted that this shortage will increase to almost 3 million by 2020. This shortage is hampering effective processing of big data and thus deriving valuable insight into it and making strategic decisions based on the data. Current data processing methodologies, therefore, need to evolve to address this problem.

A. Data Lake

The term *data lake* was first coined in 2010 by Dixon [5], using an analogy with water - data lake being a large body of data that can be examined and sampled by various users, and into which new data is "flowing in" constantly. Data lakes evolved into massive, easily accessible data repositories for storing big data in a raw format and in flat architecture (as opposed to hierarchical organisation of data in data warehouses), optimised for large-scale data analytics using *schema-on-read* [6] methodology where meta-data is retrieved as the data is retrieved. While schema-on-read increases effectiveness and efficiency of the data lake, it also requires additional preparation of the data, e.g. to extract relevant meta-data from it, before analytics can be applied to it, since data lake can also contain *unstructured data*.

We can observe five dimensions in which data lakes are growing more complex. In Section III, We will precisely quantify the complexity in each of these dimensions as a part of the overall *coefficient of swampiness* of the data lake, which will indicate how "contaminated" the data in the lake is. The dimensions are:

- *Variety* refers to various and more different types of data are stored in data lakes, from textual data in XML form, to images, videos and other kind of binary data. This significantly increases complexity of processing the data, as the data analytic applications need to deal with completely disparate kinds of data.
- *Veracity* refers to noise and abnormalities in the data, as well as determining whether a given data is relevant to the problem that is being solved. A significant part of preparation of data for analytics is ensuring the data lake remains clean an preventing accumulation of "dirty" data, It provides meaningful provenance, reliability, explains the methodology followed in collecting of the data .
- *Validity* refers to how accurate and correctness of the data for its intended use. An significant part of the data processing of the data ensures that consistence, data quality, common definitions and meta-data is up-to-date.
- *Variability* refers to the changes in types of data are stored in data lakes for specific business item. The variance is generated as data evolve over time in format, type and relevance to the business outcome changes.
- *value* refers to the actual business value a specific data items has to the overall processing requirement. The 'true' value of a specific outcome can be measure in per hour savings or simply market share.

Cesspool, Data Swamp and Curated Data Lake: Depending on the degree of cleanliness of the data in the data lake, and the amount of meta-data available, we can distinguish between two different kinds of data lakes - *data swamps* and *curated data lakes*. *Curated data lake* is a data lake where each of the data items has four meta-data parameters completely resolved:

- *unique identifier* which can be used to uniquely identify each item in the lake;
- *lineage of processing pipeline* which records where the data has been obtained from and how it changed in between different stages of the processing pipeline;
- *providence of processing pipeline* which records complete history of any processing action taken on the data item;
- *active processing pipeline* with records for history of the processing to identify previous successful processing patterns for future enhancement of the process of data item.

If some of this meta-data is not available, then the data lake is a *data swamp*. The goal of this paper is to allow semi-automatic transformation of a data swamp into curated data lake.

The lack of curating a data swamp will result is a cesspool that holds data in storage that is stagnant, corrupt and toxic to the healthcare process.

B. R-A-P-T-O-R/Q-U-B-E

This novel custom build transformation engine is the core controller for the processing capability of our research. The R-A-P-T-O-R/Q-U-B-E consists of two main components:

- Retrieve, Assess, Process, Transfer, Organise, Report (R-A-P-T-O-R) pipeline powered by the Rapid Information Factory (RIF) (Past research project)
- Quantum Universal Bounded Engine (Q-U-B-E) processing entity (New novel Reinforce Deep Learning Engine)

1) Rapid Information Factory (RIF): The Retrieve, Assess, Process, Transfer, Organise, Report (R-A-P-T-O-R) pipeline enables the Rapid Information Factory (RIF) to process data lake by curating the data lake.

- Retrieve - Processing pipeline for raw data transfer into the data lake.Curating the data lake to improve *Swampiness coefficient* by lowering ””variety coefficient”. The pipeline also collects *lineage* and *providence* for the meta-data of the data lake.
- Assess - Assess quality and accuracy of the data within the data lake. Curating the data lake to improve *Swampiness coefficient* by lowering *Validity coefficient*. The pipeline also creates *lineage* and *providence* for the meta-data of the data lake.
- Process - Transform the data into a Time-Person-Object-Location-Event (T-P-O-L-E) data vault to create uniform data structures for upstream data science. The pipeline also creates *lineage* and *providence* for the meta-data of the data lake.
- Transform - Perform human-in-the-loop data science and business intelligence on the data vault. The pipeline also creates *lineage* and *providence* for the meta-data of the data lake. The result is a data warehouse with dimensions and facts created as insights into data vault.
- Organise - Subdivide the data warehouse into appropriate subject specific data marts to form business areas of insight and interest. The pipeline also creates *lineage* and *providence* for the meta-data of the data lake.
- Report - Generate visualisations for the business to report the knowledge insights. The pipeline also creates *lineage* and *providence* for the meta-data of the data lake.

2) Deep Learning Engine:

- Quantum - Smallest data action required to manipulate the data entities using dynamic evolutionary meta-programming.
- Universal - Systematic and methodical set of rules. Novel *Data Crawler* design optimises the *fitness* of the processing patterns.
- Bounded - Rules set is bounded as it has a finite population of Data Crawlers.
- Engine - Data processing appliance that converts data into knowledge and insights. The Rapid Data Factory acts as a processing engine to control the data processing in the data lake.

C. Smart Patient-Centric Healthcare

We used a specific data lake as a hypothesis test case to concentrated the research onto a practical real-world scenario to explain the precise workings of the proposed solution for the automatic curating of the data swamp into a data lake by improving *swampiness coefficient* of the data lake.

The data consists of:

- Patient Details
- Health Care Records
- Treatment
- Facility

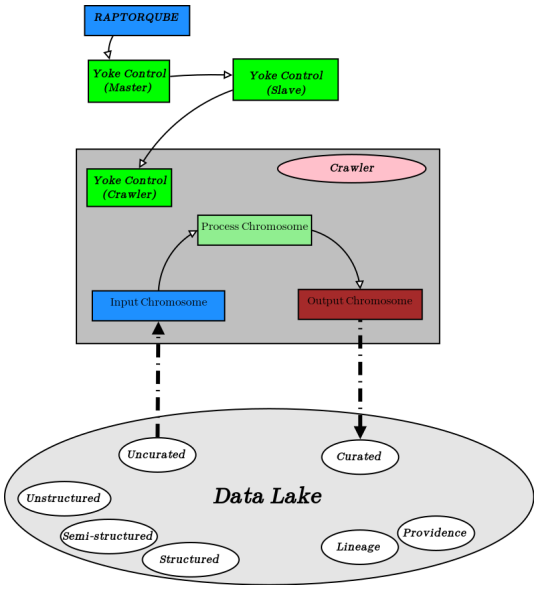
- Location
- Calendar table
- Shapes of geospatial boundaries of area covered by the facility.

1) Dynamic Programming (DP): Dynamic programming is a methodology for solving a complex problem by breaking it down into a collection of simpler sub-problems, solving each of those sub-problems just once, and storing their solutions.

2) Python Library (raptorqube): The dynamic programming (DP) is my solution is achieved by using a python library called *raptorqube* to generate *data crawlers* using DP. This is ongoing research work.

The *data crawlers* consists of three distinctive processing steps:

- Input Agreement - ingress the data from the data lake into the crawler.
- Process Agreement - performs a state check, reward calculation or curating action.
- Output Agreement - egress the data from crawler to data lake.



A genetic algorithm (GA) assists selection of each of the data crawlers for their fitness to perform a specific function in the overall process. This results in a process that can adapt and evolve by simply combining combinations of input, process and output agreements to formulate candidate using mutation, crossover and selection of the processing agents (data crawlers).

3) Evolutionary multi-agent system (EMAS): The processing of a data lake is performed by a set of finite processing agents generated by dynamic meta-programming generate the specific feature agents requirements. The EMAS forms a *directed acyclic graph* between the resulting data crawlers. The resulting processing paths then guide the Rapid Information Factory (RIF) to curate the data lake.

D. Evolutionary Meta Programming (EMP)

Evolutionary meta programming is a stochastic optimisation technique of the data lake's meta-data to enable the evolutionary meta-programming of a given processing generation for a EMAS. The evolution is achieved through appropriate selection, crossover and mutation of data crawlers to support the optimum pipeline solution. This enables the optimisation of the current EMAS to evolve to a better generation against the solution.

E. Deep Reinforcement Learning (DRL)

Deep Reinforcement Learning (DRL) is a end-to-end reinforcement learning methodology that discovers the complete data lake process via evolving supervised learning without need of human-in-the-loop generating sample labelled data.

The finite impulse recurrent neural network (fiRNN) optimises the finite state machine (FSM) formulated in the directed acyclic graph that models the interaction between the data crawlers in a population.

¡– Include a figure of a DAG –¿

The specific DRL we use formulates a special multi-layer perceptron network that consists of cascaded DRL populations that enables knowledge transfer, selective attention, prediction and exploration between the evolving populations of data crawlers.

The *R-A-P-T-O-R/Q-U-B-E* with the help of the *Rapid Information Factory (RIF)* executes the surviving population of data crawlers to curate the data lake. This creates a micro-processing capability that enables a fine-grain evolution of the processing eco-system.

III. Characterising Swampiness of Data Lakes

In this section, we precisely quantify how "contaminated" the data in data lake is, related to the five dimensions described in Section II-A. We define the *coefficient of swampiness* of a data lake, SC, as:

The "swampiness coefficient" is the sum of the five dimension coefficient (ϵ) weighted by their individual impact (α) factor.

$$SC = \sum_{i=1}^{5} \alpha_i \times \epsilon_i$$

whereas $\epsilon_i, i \in \{1, 2, 3, 4, 5\}$ is a coefficient of the i-th dimension (described below), and α_i is the weight associated to this dimension in the overall formula. Coefficients ϵ_i are defined as:

1) For the *coefficient of variety,* ϵ_1, we first define complexity δ_j of each type of data j stored in the data lake by

   $$\delta_j = nrDataItems_j \cdot complexityFactor_j.$$

   If n is the number of different types of data stored in the data lake, ϵ_1 is then defined as

   $$\epsilon_1 = \sum_{j=1}^{n} \delta_j.$$

2) For the *coefficient of veracity,* ϵ_2 is defined in terms of data items for which we have information about provenance:
   if $nrRecordswithProvenance = 0$ then

   $$\epsilon_2 = nrRecords$$

   else

   $$\epsilon_2 = \frac{nrRecords}{nrRecordswithProvenance}.$$

3) For the *coefficient of validity,* ϵ_3, is defined in terms of the proportion of data items that are valid in the data lake:
   We need:

   $nrRecords$ the volume of the data lake.
   $nrRecordsValid$ the volume that is valid data.

if $nrRecordsValid = 0$ then

$$\epsilon_3 = nrRecords$$

else

$$\epsilon_3 = \frac{nrRecords}{nrRecordsValid}.$$

4) For the *coefficient of variability,* ϵ_5 is defined as the amount of changes required against previously stable and good fitness processing code.
We need:

$nrProcesses$ is the data crawler population.
$nrValidProcesses$ the fit data crawler population after the changes.

First we calculate the $totalProcessFitness$:

$$totalProcessFitness = \sum_{i=1}^{n} ProcessFitness_i$$

where $n = nrProcesses$
Secondly we calculate the $averageProcessFitness$:

$$averageProcessFitness = \frac{totalProcessFitness}{nrProcesses}$$

Thirdly we calculate *coefficient of variability,* ϵ_5:

$$\epsilon_5 = \frac{nrProcesses}{nrValidProcesses} \cdot averageProcessFitness$$

5) For the *coefficient of value,* ϵ_4, is defined in terms of the business value of each individual record multiplied by the number of users of that record: We need:

$nrRecords$ the volume of the data lake.
$nrUsers$ that depends on each data item.
$businessValue$ of each data item.

$$\epsilon_4 = \sum_{j=1}^{n} businessValue_j \times nrUsers_j.$$

where $n = (nrRecords - nrRecordsValid)$ in data lake.

Now we have the complete calculation for the *coefficient of swampiness* SC.

IV. Evolving the swamp

The Deep Reinforcement Learning (DRL) enables the processing of the data swamp into a curated data lake by generating genotype (R-A-P-T-O-R/Q-U-B-E model) for the processing ready for the meta-programming to produce a population of phenotypes (data crawlers) that then is evaluated by a combination of two phase evaluation of the EMAS DAG validity and execution of the valid data crawlers against the data lake.
The "swampiness coefficient" for the resulting data lake is used a the fitness test of a specific population of data crawlers' paths for survival into a next generation of the population of data crawlers.

If "swampiness coefficient" has increases the population suffers a mass extinction effect and is 100% culled.

If "swampiness coefficient" has decreased, the population is a viable population to survive and a process of partial culling and new data crawler generation is started.

We then activate the next generation by determining the specific individual data crawlers survival fitness.
We define the fitness of a specific data crawler (i) by the following formula:
if $nrRecordsIn_i > 0$ and $nrRecordsOut_i > 0$ then:

$$\omega_i = \frac{nrRecordsIn_i \times runTimeInMilliSeconds_i}{nrRecordsOut_i}$$

if $nrRecordsIn_i = 0$ or $nrRecordsOut_i = 0$ then:

$$\omega_i = 0$$

and is 100% culled.
if data crawler (i) is orphaned:

$$\omega_i = 0$$

and is 100% culled.
Secondly we define the specific path (j) of surviving data crawlers' fitness by:

$$\Omega_j = \sum_{i=1}^{nrCrawlersInPath_j} \omega_i$$

This path describes the order of the processing required to enhance the data lake by curating it with better lineage and providence. This actions enable the removal or correction of issues that will negatively impact the successful processing of the data lake.
We determine next generation of the population of data crawlers with a set of evolution rules.

The following rules was assumed to evolve the data lake:

- Four fittest paths of data crawlers are spawned via selection.
- Four new crawlers are spawned via cross-over of two fittest data crawlers at every layer of the R-A-P-T-O-R synchronisation.
- Four new paths of crawlers are spawned via mutation of four fittest path's data crawlers.

V. Exit Criteria for Data Lake Optimisation

The data lake is optimised until the "swampiness coefficient" is stable within 6σ of the median of the "swampiness coefficient" last 100 consecutive surviving populations.

This enables the reinforce deep learning to balance the "swampiness coefficient" against processing cost to maintain the achieved coefficient.

VI. Evaluation

- Volume
  - 1 Month of data
  - 6 Months of data
  - 1 Year of data
  - 2 Years of data
- Variety
  - 1 Type of data
  - 3 Types of data
  - 5 Types of data
- Veracity
  - 100% error-free data
  - 50% error-free data
  - 10% error-free data
- Validity
  - 100% error-free data
  - 50% error-free data
  - 10% error-free data
- Value
  Use first five prime numbers from a Fibonacci sequence to determine business value impact:

$$F_n = F_{n-2} + F_(n-1) \quad \text{for } n \in \{2, 3, 5, 7, 11, 13\}$$

VII. Related Work

A. Managed Ingestion

The use of managed ingestion into the data lake could improve the "swampiness coefficient" of the data before it enters the data lake.

This creates a "raw data pool" that then needs curating before been allowed into the "curated data lake".

B. Metadata Management

Metadata management supplies governance capabilities for building a catalogue of a data lake. This will assist with the creation of lineage. The meta-data management of a at-scale data lake is a major study field in the data science communities.

C. Security and Data Privacy

Enable, monitor and manage comprehensive data security across the data lake ensures fine grained authorisation for any data been create, read, update, and delete (CRUD) in the data lake.

The implementation of General Data Protection Regulation (GDPR) on 25 May 2018 provides the following data rights:

- The right to be informed - Data Lake needs full lineage and providence to inform data owners of the legal requirements.
- The right of access - Data Lake requires role-based access control (RBAC) design
- The right to rectification - Curate the Data Lake on request.
- The right to erasure - Curate the Data Lake on request.
- The right to restrict processing - Curate the Data Lake within restrictions.
- The right to data portability - Only transfer data with permission.
- The right to object - Keep full audits of data lake.
- Rights in relation to automated decision making and profiling - Curate the Data Lake with autonomous processing.

D. Data Life cycle Management

Data life cycle management (DLM) is a policy-based approach to managing the flow of data throughout its life cycle within data lake.

Seven Data Processes required to achieve full DLM:

1) Data Capture during data acquisition into the data lake from other data sources.
2) Data Maintenance of the data lake by curating the data on a continuous cycle.
3) Data Synthesis uses advance analytics to generate insights.
4) Data usage requires special data governance meta-data with the introduction of GDPR. All processing and data use must be secured, logged and audited.
5) Data Publication is now required to supply full lineage and providence of all data ported to other data custodians.
6) Data Archival need to be designed into the curate process of the data lake.
7) Data Purging must be designed to ensure valid purging of data lake.

E. Data Zoning

The sub-division of a data lake into five smaller ponds of interest results in a specific "swampiness coefficients" for each pond.

The following suggestions are investigated by other research streams.

- Raw Zone in a Data Lake stores the data in its raw form to ensure an untainted base reference of the data lake.
- Structured Zone contains structures that provide a first stage of transformation of the data from the Raw Zone.
- Curated Zone contains data that is often organised in a approved data model which combines similar from a variety of raw sources to form a canonical model.
- Consumer Zone of a Data Lake provides an easy trusted access point for consumers.
- Analytics Zone allows data scientists to analyse and experiment with data in the Lake.

VIII. Conclusion

The conclusion goes here.

ACKNOWLEDGMENT

The authors would like to thank... more thanks here

REFERENCES

[1] G. Bell, T. Hey, and A. Szalay, "Beyond the data deluge," *Science*, vol. 323, no. 5919, pp. 1297–1298, 2009.

[2] L. I. Perlovsky, "Conundrum of combinatorial complexity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 666–670, 1998.

[3] W. Butz, G. Bloom, M. Gross, K. Kelly, A. Kofner, and H. Rippen, "Is there a shortage of scientists and engineers? how would we know?" *Rand Corporation*, 2003.

[4] T. H. Davenport and D. Patil, "Data scientist," *Harvard business review*, vol. 90, no. 10, pp. 70–76, 2012.

[5] J. Dixon, "Pentaho, hadoop, and data lakes," *blog, Oct*, 2010.

[6] T. Deutsch, "Why is schema-on-read so useful," *IBM big data and analytics hub*, 2013.