

ULRR

Towards automatic data cleansing and classification of valid historical data an incremental approach based on MDD

Item Type	Meetings and Proceedings
Authors	O'Shea, Enda;Khan, Rafflesia;Breathnach, Ciara;Margaria, Tiziana
Citation	2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 1914-1923
Publisher	Institute of Electrical and Electronics Engineers
Download date	2026-03-14 08:41:14
Item License	https://creativecommons.org/licenses/by-nc-sa/4.0/
Link to Item	https://doi.org/10.34961/researchrepository-ul.23735850

Towards Automatic Data Cleansing and Classification of Valid Historical Data An Incremental Approach Based on MDD

Enda O’Shea*, Rafflesia Khan*, Ciara Breathnach*[†], Tiziana Margaria*[†]

*University of Limerick, Limerick, Ireland - {name.surname@ul.ie}

[†]Lero: The Irish Software Research Centre

Abstract—The project *Death and Burial Data: Ireland 1864-1922 (DBDIrl)* examines the relationship between historical death registration data and burial data to explore the history of power in Ireland from 1864 to 1922. Its core Big Data arises from historical records from a variety of heterogeneous sources, some aspects are pre-digitized and machine readable. A huge data set (over 4 million records in each source) and its slow manual enrichment (ca 7,000 records processed so far) pose issues of quality, scalability, and creates the need for a quality assurance technology that is accessible to non-programmers. An important goal for the researcher community is to produce a reusable, high-level quality assurance tool for the ingested data that is domain specific (historic data), highly portable across data sources, thus independent of storage technology.

This paper outlines the step-wise design of the finer granular digital format, aimed for storage and digital archiving, and the design and test of two generations of the techniques, used in the first two data ingestion and cleaning phases.

The first small scale phase was exploratory, based on metadata enrichment transcription to Excel, and conducted in parallel with the design of the final digital format and the discovery of all the domain-specific rules and constraints for the syntax and semantic validity of individual entries. Excel embedded quality checks or database-specific techniques are not adequate due to the technology independence requirement. This first phase produced a Java parser with an embedded data cleaning and evaluation classifier, continuously improved and refined as insights grew.

The next, larger scale phase uses a bespoke Historian Web Application that embeds the Java validator from the parser, as well as a new Boolean classifier for valid and complete data assurance built using a Model-Driven Development technique that we also describe. This solution enforces property constraints directly at data capture time, removing the need for additional parsing and cleaning stages. The new classifier is built in an easy to use graphical technology, and the ADD-Lib tool it uses is a modern low-code development environment that auto-generates code in a large number of programming languages. It thus meets the technology independence requirement and historians are now able to produce new classifiers themselves without being able to program. We aim to infuse the project with computational and archival thinking in order to produce a robust data set that is FAIR compliant (Free Accessible Inter-operable and Re-useable),

Index Terms—Data Collection, Data Analytics, Model-Driven Development, Historical Data, Data Parsing, Data Cleaning, Ethics, Data Assurance, Data Quality

“Death and Burial Data: Ireland 1864-1922” is a project funded by Irish Research Council Laureate Award IRCLA/2017/32 to Dr. Ciara Breathnach (Department of History - DH), in cooperation with Prof. Tiziana Margaria (Software Systems, Dept of Computer Science and Information Systems - CSIS) at the University of Limerick. 978-1-7281-6251-5/20/\$31.00 ©2020 IEEE

I. INTRODUCTION

“Death and Burial Data: Ireland 1864-1922 (DBDIrl) [1]” is a national, data-driven public-history research project in the digital humanities that uses historical death registration data and other validating data records such as census returns, to explore the history of Ireland from 1864 to 1922. The access to Historical Civil Registration (CR) data provides a unique opportunity to examine the health of a population over a long time-frame. It also provides an insight into how people responded to and engaged with local government agencies. These official historical records are Big Data: approximately 4.3 million individual CR records just for the death registration with over 1TB of associated images, which in the absence of more complete metadata is analogue in its current guise. The project will use modern Machine Learning (ML) algorithms to determine patterns of underlying social constructs, increasing our understanding of how Irish people at that time lived, progressed and ultimately died. The lasting IT value of the project is to set up an end-to-end data processing and data management platform for Digital Humanities. To be accessible to non-programmers, like most domain experts in the digital humanities, we use a Model-Driven Development (MDD) framework that embeds advanced and explainable ML capabilities, and in which Big Data can be collected, cleaned, validated, verified and processed. This platform will enable users to generate substantial knowledge and critical findings.

DBDIrl has a reliable, steady source of data from the General Register Office (GRO) of Ireland and will soon receive a data dump from the National Archives of Ireland comprising the census returns for 1901 and 1911 respectively (approximately 7 million individual level records). It also uses other data resources, scattered amongst a wide range of distributed heterogeneous sources. Their heterogeneity makes it difficult to impose uniform standards, security, access control, monitoring and overall governance. To this extent, DBDIrl develops *efficient and flexible access mechanisms to high quality data* from historical records from heterogeneous sources and makes them accessible to a wider range of researchers through adequate user interfaces. The DBDIrl framework will be extensible, so as to incorporate as many resources as possible while providing services to standardise and format the data sets to allow interoperability across all supported types.

In this paper, we detail the data access, cleansing and experience of MDD adoption in our work with the Historical Death Registration data provided by the Irish GRO, in particular, two iterations of the process of its digitization and cleaning prior to storage, management and use. We introduce efficient and generalized methods to restore and maintain the overall integrity of the original data entries, making them amenable to meaningful ML analysis. The platform enables domain experts to perform their own analyses and maintenance in a zero to low-code, programming-less modelling environment. Through our data engineering efforts, we aim to provide a wealth of efficient ways of analysing and understanding death data, micro-histories, and underlying reasons for cause of death in Ireland, from the introduction of civil registration in 1864 until the foundation of the Irish Free State in 1922.

Challenges in setting up a Knowledge Discovery process [1] within a historical data management framework are abundant. DBDIrl's Big Data comes from heterogeneous data sources which need to be collated, filtered, cleaned, integrated, organised, sketched, processed, verified and stored. Each of these processes has its own complexity. The main challenge concerns the CR records: they are handwritten entries in physical registers, which require skilled interpretation of the handwriting, knowledge of the outdated terminology and conventions, followed by their digitization, processing and validation with minimal error. Seemingly minor discrepancies or incorrect transcription of a data record can harm the research outcomes, with lineage being distorted or potentially broken. Hence, the core aspect of the DBDIrl project is the development of a complete data management and analysis platform, that is easy to change and can evolve as knowledge, practices and technologies improve over time.

The choice to adopt an MDD approach provides non-computing experts from the humanities, like the humanities scholars on the project, the ability to control and manage their data efficiently while at the same providing services that ease the use of ML techniques, embedded in the DBDIrl platform in a service oriented fashion. We use Domain-Specific Languages (DSL) [2] combined with a Service oriented approach [3], components needed to model, implement, and provision the necessary IT capabilities. Services can provide a level of abstraction, while encapsulating all required domain logic that is meaningful to the humanities user, which shields users from the underlying implementation and platform details to focus more on functionality and usability. For example, the *DBDIrl Data Cleansing DSL* encapsulates the collection of classifiers that implement the analysis of data and the decisions about their syntactic or semantic correctness. These decision services consist internally of a finite set of constraints [4] that characterise the data properties as Reduced Ordered Binary Decision Diagrams (ROBDDs) [5], that domain experts do not need to know. Domain experts see a collection of modelling elements phrased in their own language, like *Validate Name*, *Validate Cause of Death*, that are easy to understand and use correctly when setting up a specific data cleansing process. Through these DSLs, domain experts can perform operations

on their data without need of expert IT assistance nor extensive training in programming, computer science or information systems. Several recent low-code and zero-code application design tools provide such services. The DBDIrl platform uses the DIME tool [6] for process modelling and the ADD-Lib tool [7] to implement the cleansing of "knowledge". DIME is a general purpose Integrated Modelling Environment for Web Application development, connected with a modern DevOps pipeline that brings models to run. ADD-Lib [7] is a Java framework for the design and composition of Decision Diagrams across various mathematical domains, used to implement our classifiers for data cleansing. Through these environments, an end-to-end data processing framework is currently under development to realise the full pipeline and provide ease of use to future domain experts.

Key contributions of this work are as follows:

- 1) An ad hoc intelligent parser that automatically transforms the original coarse granular representations, created in the manual data capture in spreadsheets, to finest granular data items ready for queries by the researchers.
- 2) Data cleaning applied to spreadsheets, with potential errors highlighted for further revision where processing could not determine an appropriate automatic fix.
- 3) Development of a set of Boolean classifiers that embed explainable ML technology. Using the ADD-Lib MDD tool, the classifiers' code is generated from the understandable models and integrated in a new Web Application for scalable and efficient online data entry.
- 4) This study is the first exploration of potential patterns in causation of death within the Irish populace and the potential underlying systematic variables that may be of influence. A high threshold of data quality and assurance must be met for modern ML techniques to be applicable.

The paper is organized as follows: Section II describes related work in the fields of historical data, Big Data, and MDD. Section III describes our data collection and ethics management processes. Section IV explains the data parsing and cleaning along with the challenges faced, with Section V showing the results of these processes. Section VI describes the model-driven approach for data classification. Section VII describes the classification results obtained with the ADD-Lib based classifier and Section VIII concludes the paper and highlights some future work.

II. RELATED WORK

In recent years, the study of population history has been used for detailed studies in various research concerning population progression patterns [8], the evolution of ecosystems [9], the effects of aging [10], global healthcare [10], [11], and impact of social change [12]. Marti-Henneberg, J. et al [8], proposed a methodology based on the homogenization of data and administrative units to compare and analyze long term patterns of population concentration over a period of 125 years in Spain. According to [9], "Historic data on biodiversity provide the context for present observations and allow studying long-term changes in marine populations".

The World Health Organization (WHO) understands the importance of continued research in population factors such as ageing, through the Study on Global AGEing and Adult Health (SAGE) [13], health [14], and cause of death analysis such as those pertaining to maternal death cases [15]. DBDIrl researches similar fields, using data of long past generations, which may provide patterns and knowledge to assist with identifying early indicators of such populace characteristics. Other similar research analysed the worldwide availability of cause-of-death statistics [16], outlining the importance of civil registration data in providing a means to develop comprehensive and detailed life-event data analysis to deepen understandings of life courses, survival patterns, population change, mortality risks and in measuring health trends as demonstrated in [17].

According to [18], Model-Driven Engineering (MDE) is considered as one of the most popular approaches in software abstraction as stated by [19], “Models allow sharing a common vision and knowledge among technical and non-technical stakeholders, facilitating and promoting the communication among them.” The eXtreme Model-Driven Development (XMDD) [20] approach is a formal methods backed low-code approach that joins together several traditional programming trends like service orientation, aspect orientation, data management, model-driven development, agility, eXtreme programming, generative programming, and full code generation [20], [21]. XMDD is best adopted within a Continuous Model-Driven Engineering paradigm that allows for quicker design, development, and deployment of purpose-specific complex software systems employing Big Data through the use of ad hoc meta-models.

Previous works on XMDD such as [5] and [22] demonstrate the use of MDE in robotics, and SkyViz [13] is a model-driven approach to automate the translation of user objectives for visualizing the results of Big Data analytics into a set of most suitable, concrete visualizations. Breuker Dominic [23] introduces a ML based MDE approach for analyzing Big Data by defining a domain specific modeling language for probabilistic modeling.

These approaches, from their respective fields, provide the foundations for our development of a DBDIrl framework that delivers significant knowledge-discovery capabilities through MDD processes and service orientation. From a CR perspective, it starts by working to assure the quality of the Big Data already obtained through the capture and ingestion of the civil registration records.

III. DATA COLLECTION AND ETHICS

This section outlines the properties of a CR record, the workflows and structures incorporated in the Data Collection and Ingestion Phase, the challenges encountered as well as the ethical responsibilities in dealing with this restricted access data set of personal data.

A. Civil Registration Records

Three types of Civil Records (CR) were introduced for Roman Catholics in Ireland in 1864: birth, death, and marriage registers. They record a large amount of handwritten data about individuals and their relatives, containing personal elements such as names, dates, occupations, and addresses. DBDIrl focuses primarily on death registration as it enables us to close off a life course in Ireland - a country of high net emigration. This work permits a study of the social determinants of health from gender, life cycle and class perspectives from macro and micro-history vantage points [24]. The GRO provided DBDIrl with a data dump in the form of an Excel spreadsheet containing indices to the volumes of the civil registration of deaths along with high resolution scans of the original register pages. The GRO scans are digital reproductions of the writings originally produced by the registrar at the time of death. Each scan is stored in Tagged Image File Format (TIFF), with every file showing only one page and up to 10 entries, each entry having 11 properties as shown in Table I. Fig.1 shows an example image of a TIFF file (top) along with a sample spreadsheet of a single record (bottom), one record per row, with the metadata. Currently, the GRO register data is partially available online to the public at no cost at <https://www.irishgenealogy.ie>, with search capability by name, year range, civil registration district/office, and life event level.

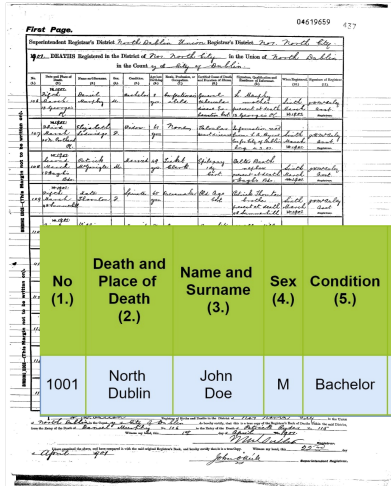
TABLE I
PROPERTIES IN THE ORIGINAL .DOC/ .TIFF FILE

#	Property Name	Description
1	No	Index of record
2	Date and Place of Death	Information regarding the (1) date and (2) place of individual deceased's death
3	Name and Surname	Individual deceased's name
4	Sex	Individual deceased's sex
5	Condition (marital status)	Civil status of the deceased person
6	Age last Birthday	Age of the deceased person (year/month/week/day/hour)
7	Rank, Profession or Occupation of the deceased person	Occupation
8	Certified Cause of Death and Duration of Illness	(1) Cause of death (2) If the person was ill before death, then duration of illness in year, month, week, day and hour
9	Signature Qualification and Residence of Informant	Information of informant
10	Time of Registration	The time of record registration
11	Signature of Register	Information about the signature of the informant who registered the record

B. Data Ethics Maintenance

The data management process is consistently transparent. In accordance with the GRO data agreement, only core project members have access to the entire data dump, which is stored on a non-networked PC. The core principles of data ethics such as transparency, accountability, equality, individual data control and human interest are observed. Data pertaining to students assisting with data input is stored in accordance with General Data Protection Regulation (GDPR). The final data set files are saved in the local repository hosted on the local intranet of the University of Limerick (UL). The Principal Investigator (PI) is responsible for granting access only to team members and student participants, only with official UL email

(A) Sample TIFF file of GRO death Data



(B) A sample spreadsheet entry view.

No (1.)	Death and Place of Death (2.)	Name and Surname (3.)	Sex (4.)	Condition (5.)	Age Last Birthday (6.)	Rank, Profession, or Occupation (7.)	Certified Cause of Death and Duration of Illness (8.)	Signature Qualification and Residence of Informant (9.)	When Registered (10.)	Signature of Registrar (11.)
1001	North Dublin	John Doe	M	Bachelor	22	Car driver	Heart attack	Marry jane	3/7/1901	P.H. Jhon

Fig. 1. Death record of Irish civil registration: the GRO original register page (TIFF file) and a sample spreadsheet structure

addresses and associated credentials. Different levels of access rights are assigned on a file-by-file basis, with traceability features to determine the origin of modifications. GDPR is not applicable to data pertaining to deceased, however for exemplification in publications and presentations we regularly resort to proxy data in place of actual data.

C. Data Collection and Ingestion

The Data Ingestion or digitization phase of DBDIrl encountered significant challenges when digitizing CR records, especially how to handle the input of mass amounts of handwritten records. It began with the inspection of the 11 properties of the original CR records, illustrated in Fig. 1 and Table I. Understanding the data set with the aim of supporting the subsequent fine-granular access for research purposes to data elements contained within the properties [25], this led us to restructure them, initially into 24, and subsequently to 63 finer granular properties. This separates the sub-fields with potential individual relevance, in a quest to identify the adequate level of granularity for further use in research.

Data enrichment happens by manual transcription of each individual record from the TIFF files to Excel spreadsheets. Across our already transcribed records, the number of records originally handwritten by a single registrar varies between 200 and as few as 5. From a purely ML perspective, neither Natural Language Processing nor Optical Character Recognition-based text recognition are currently applicable to our sources due in part to the low number of records per registrar, multiple signature entries from differing sources that are unique to a single record, coupled with the wildly varied and often nearly illegible handwriting styles. Effectively, there is not sufficient data from each individual to train a network, and the structure in records with very short texts that are names, dates, addresses is not helpful to form a training corpus. This

applicability may be revised at a later stage, as [26] shows that through the use of a Convolutional Neural Network, aided with statistical models and crowd-sourcing efforts, they achieve up to 65% exact word transcription on selected Vatican archives. However, it should be noted that their results were achieved on large manuscripts written by dedicated scribes, therefore with high calligraphic consistency which our data do not match. In a trial carried out using the Transkribus platform [27] in fact we observed that it too required vast amounts of data from the same scribe, with the structure of the register document also causing difficulties for the recognition process. That is why we stopped that trial and resort now to the current approach. Digitization is now mostly done by DBDIrl project members, enhanced by Hackathons in order to expand the pool of contributors and saving some effort to the project members. The goal is to handle future Data Ingestion operations through the online Historical Web Application developed using MDD processes, which embeds Boolean classifiers to provide the needed quality assurance (see Section VI-A). This web application will also move storage away from Excel workbooks to a secure database structure, providing a centralised, controlled environment for all Historical records of relevance.

D. Challenges

The major challenges we faced during data entry and how we overcame those are described as follows.

Volume of Data: There are approximately 4.3 million individual CR death register records which require vast amounts of time and commitment to enrich and process manually. To counter this large-scale data collection process, a Hackathon event was organised in the Glucksman Library at the University of Limerick (UL) under the title “Death and Burial Data: Ireland 1864 to 1922”. The objectives of the Hackathon were to explain the data, and to instill computational and archival

thinking in humanities teaching and learning, and to help and guide participants in the best practices when digitizing such records. Twenty current Master students of History, three past Master students and several Doctoral scholars and lecturers took part. Each group was tasked with the digitization of 50 GRO records, and for Master students their contribution was part of their continuous assessment. The average number of complete entries was 19, and some contributed over 40.

Handwritten Records: Extracting properties from handwritten documents requires close inspection of the handwriting, skilled interpretation of the handwriting and a good understanding of the domain. For example, *suffocation of spinal cord* may be hard to distinguish from *separation of spinal cord* but an incorrect reading affects the ways in which the data can be analyzed and visualized at a later stage. To resolve the *Cause of Death* property we use the nineteenth century Statistical Nosology [28] that distinguishes between (1) *ZYMOTIC DISEASES*: acute infectious diseases, especially chief fevers and contagious diseases, (2) *SPORADIC DISEASES OF UNCERTAIN OR VARIABLE SEAT*: dropsy, haemorrhage, mortification, abscess, and cancer, and (3) *SPORADIC DISEASES OF SPECIAL SYSTEMS AND ORGANS*: infrequently and irregularly occurring diseases. For the *Name of Registrar* property we used the additional register *Dublin Data: (Incomplete) List of Registrars and Assistant Registrars* to assist in their recognition, as individual signatures are difficult to comprehend.

Ambiguity: The raw civil registration documents present significant ambiguity. The home address of an individual may appear in alternative forms, with no standard terminology even for common words. For example, 'Street' is denoted S., St., St, Street. Chaotic use of abbreviations leads to a host of semantically similar or identical terms with vastly different syntactic versions. Generalisations, such as *Widow* being entered to cover all the available input set (*Widow, Widowed, Widower*) add further ambiguity. These issues occur both in the original records and in the Hackathon data entry, and have to be handled at the cleaning stage.

Human Error: Mass data entry by individuals with limited domain familiarity is error prone when lacking knowledge, judgement and ability to appreciate differences. A common issue is the use of additional notations appended to data properties, such as changing 'Widow' to 'Widow.', where the extra dot requires additional cleaning. More complex are spelling and writing mistakes in people's name, as no knowledge-based tool can classify the correct name between two similar manual entries, like *Kent* and *Ken*. Due to these errors, each entry is manually verified by skilled DBDlrl project members, in a quality assurance process that ensures the highest possible standard for individual and family name entries. This precision could be critical in preserving family lineages and for any future planned ML operations. The Data Cleaning phase assists this task by highlighting probable errors.

Structured and Unstructured Data: While the direct transcription from the Tiff file records into Excel spreadsheets has allowed for some data fields to have well defined format

structures ('Group Registration ID', 'Tiff File Path'), other fields have lacked any singular formatting. Take the 'Duration of Illness' data field which can have various differing representations based on a registrar's preferring style of capture. An example to illustrate this could be the duration "1 year, 2 months" which can also be captured as "One year and two months", "1 year 2 months" or "1 year and two months". All these possibilities are seen in the data with other durations consisting of generalities such as "Some Months".

Excel also has an issue with date formatting as it assumes all dates of the format dd/mm/yy refers to the 21st century as opposed to the 20th. An original date of 23/12/01 should represent the year 1901 but is in fact processed as 2001 in Excel. Another issue is the Excel date system uses the floating point value 1.0 to represent January 1st 1900; therefore any value/date less than this is taken as a string, causing errors. It is well known that Excel is inadequate to deal with scientific data [29], and this is an additional reason to resort to more robust data storage technologies.

We dealt with these differences and anomalies in the Parsing phase, where the original 11 properties, containing complex data like a full address, move to a finer granular model that separates many complex entries in their constituting components: single columns are substituted by an equivalent set of columns, one for each component. This process restructures the data entries from the original 11 properties to a core set of 24 for each individual record (the format used in the Hackathon), further expanded to 63 columns during the parsing phase to achieve more granular detail. Table II shows the final 63 properties along with their description.

Missing Information: Many records are incomplete, due to missing entries by the registrar or wear and tear on the original document. Similarly, Hackathon participants skipped or missed some entry values. Essential missing properties were later highlighted during the cleaning phase for rectification.

Most of the GRO data still remains to be digitized. This work will happen with more procedural guidance and checks, to ensure a high quality input system and to reduce the need for any additional data cleaning operations. In the next digitization phase we will abandon the use of Excel. Instead, we will use the Historical Web Application that includes a proper database with a strong and immediate type and value checking. Users will input data in designated fields with predefined formats and values, and we developed classifiers that check at input time the plausibility and completeness of the values. This way the workload can be distributed to a further reaching community over the internet, where individuals interested in history can contribute to the effort. Parsing, cleaning, and the classifiers used in these phases for the final preparation of the currently digitized data are discussed in detail in Sections IV and VI.

IV. DATA PARSING AND CLEANING

A. Data Entry with Less Error

In the cleaning phase, data sets are first transformed to the new formatting standard, then the individual input fields are parsed to highlight potential input errors, that are detected on

the basis of constraints and correlation checks. An ad hoc parser developed in Java delivers both operations on the Excel spreadsheets containing the digitized data.

B. Data Parsing

The digitized data sets from the data ingestion and digitization phases contain 24 data fields, many still complex. Fields expressing time such as age and durations encapsulate and structure complex data with commas and brackets. For example, an age represented as *5 years, 2 months, 3 days* has 3 comma separated sub-fields. Due to the mentioned Excel format issues, ensuring that data values are within the boundaries of 1864 to 1920 uses constraints and programmed format transformations. Parsing the complex data types to extract their elements to dedicated columns results in the expansion to the 63 finer granular data fields described in Table II. This is needed to allow for ML and other analysis algorithms to operate on the finer granular data. The Historical Web Application to be used going forward indeed adopts this fine granular representation at its data entry standard.

Due to the specialized domain knowledge needed for almost every data element in the records, no standard tool was able to provide acceptable extraction quality. The ad hoc parser implemented in Java has two primary tasks: 1) process the unstructured or semi-structured Excel data to produce the fine granular data, and 2) identify invalid input fields, highlighting various types of errors for further revision by DBDlrl members. Together, the two processes produce high-quality data sets amenable to various kinds of analysis at any desired granularity.

C. Data Cleaning

The Data Cleaning phase began with consultations between the humanities scholars and the computer scientists to examine the quality of the original data and decide how to ensure that any future research could be carried out in good faith with assured veracity [30] of the data. Here, the meaning of data and its connection to a proper representation play a central role, determining specific qualities for each input field and required or forbidden correlations across fields. As a consequence, the ad hoc parser was extended with an error identification protocol that evaluates all the properties for each

Sample-Data	Group Registration ID	Date of Death	Sex	Civil Status
<u>Correct Entry</u>	1234567	01/02/1900	Female	Married
<u>Incorrect Entry</u>	ABC*123		Male	Spinster

User Input Error

Missing Entry Error

Correlation Error

Fig. 2. Example of data cleaning approach. (Column numbers reduced for clarity)

TABLE II
IDENTIFIED PROPERTIES FOR THE FINE GRANULAR DATA SET REPRESENTATION

No.	Property	Description
1	Group Registration ID	Registration ID of the deceased record, part of TIFF file batch
2	District	District of the deceased. Specific area of county, e.g Clontarf
3	Date of Death	Date of Death for the deceased (dd/mm/yyyy)
4-10	Place of Death - House Number, Street, District, County, City, Latitude, Longitude	Deceased Place of Death Address details to allow for potential detailed geographical research
11-17	Residence - House Number, Street, District, County, City, Latitude, Longitude	Deceased Residence Address details to allow for potential detailed geographical research
18	Sex (Male, Female, Unknown)	Sex of the deceased represented by an acronym in the data set, e.g Female - F
19	Deceased Forename	Forename of the deceased
20	Deceased Surname	Surname of the deceased
21-27	Deceased Age - Years, Months, Weeks, Days, Hours, Minutes, Other	Age of the of the deceased at time of death
28	Civil Status (Married, Spinster, Bachelor, Widow, Widower, Unknown)	Civil Status of deceased represented by an acronym in the data set, e.g Married - M
29	SR District/Reg Area	Registration District of deceased. More general than the earlier District, e.g Dublin North
30	Rank - Relation	Rank in terms of life status of the deceased, e.g Lady
31	Rank - Occupation	Occupation of the deceased
32	Cause of Death 1	Primary Cause of Death
33-39	Duration of Illness 1 - Years, Months, Weeks, Days, Hours, Minutes, Other	Duration of the illness of the deceased for their primary cause of death
40	Cause of Death 2	Secondary Cause of Death
41-47	Duration of Illness 2 - Years, Months, Weeks, Days, Hours, Minutes, Other	Duration of the illness of the deceased for their secondary cause of death
48	Certified (Certified, Uncertified)	Death registration certified for deceased, represented by an acronym in the data set, e.g Certified - C
49	Name of Informant - Forename	Forename of informant
50	Name of Informant - Surname	Surname of informant
51	Qualification of Informant	Qualification of informant, such as relation to the deceased
52	At Time of Death (Present, Not Present)	Informant present at the time of death of the deceased, represented by an acronym in the data set, e.g Present - P
53-59	Residence of Informant - House Number, Street, District, County, City, Latitude, Longitude	Informant Residence Address details to allow for potential detailed geographical research
60	Date of Registration	Date registration record was created
61	Name of Registrar	Name of registrar who created the record, identified through their signature
62	Tiff File Path	Location of original TIFF file along with batch number
63	Misc - Notes	Relevant notes for deceased/registration record that no input field can hold

record captured in Excel by DBDlrl members and in the Hackathons. This protocol highlights all the errors, marking them for further revision.

Examples of correct and incorrect user inputs can be seen in Fig.2. Data cleaning must preserve the correct meaning and values [31], therefore we highlight probable errors for future examination by DBDlrl members, instead of trying to fix them automatically.

TABLE III
NUMBER OF ERRORS IDENTIFIED PER ERROR CATEGORY

Data Sets	Data Set 0	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Data Set 6	Data Set 7	Data Set 8	Data Set 9	Data Set 10	Data Set 11	Average
Data Fields (j)	39123	39060	39060	39060	39060	39060	39060	39060	39060	39060	39060	38178	38991.75
<i>Errors Highlighted (%)</i>													
RED ($i=0$)	5.02	2.42	3.10	2.98	1.95	2.56	2.58	2.28	1.75	1.44	2.92	4.42	2.78
YELLOW ($i=1$)	17.86	33.71	41.86	40.14	41.01	36.72	44.34	35.62	52.38	40.03	33.93	39.63	38.10
BLUE ($i=2$)	0.04	0.01	0.00	0.01	0.01	0.01	0.005	0.007	0.00	0.01	0.01	0.002	0.01

V. CLEANING RESULTS: ERROR CATEGORIZATION

The parser worked across the 12 data sets created by students in the Hackathon, each containing approximately 618 entries and 39,000 individual data fields, stored in Excel spreadsheets. The parser analyzes each field, reformats it if needed, assigns it an error type, and in the case of an error highlights it in the corresponding color. The percentages of errors per data set are calculated using (1):

$$Error_{i,j} = \frac{ErrorsHighlighted(i)}{DataFields(j)} \times 100 \quad (1)$$

Here, $i = [0..2]$ represents the error type, $j = [0..11]$ represents the data set index. Table III reports the percentage of errors highlighted, per type (colour) for each data set as well as the average.

The majority of errors highlighted by the parser concern **missing data fields** (YELLOW) with an average error rate of 38.1% (minimum 17.86%, maximum 52.38%) across all required fields. This is due to either incomplete original records or users unable to transcribe the handwritten element with sufficient confidence, resulting in no data being input. Resolution of these missing data fields is dependent on their potential importance to future ML analysis. Set 8 is deemed anomalous as the group completed only 6 of 24 fields, thus many fields were empty and highlighted as erroneous.

User input errors (RED) occurred on average 2.78% per data set (min 1.75%, max 5.02%) and are deemed errors of high importance. Errors of this nature would typically be input errors by the user that would fall outside the constraints set by the historians for that specific field type, and as such would need immediate rectification by a DBDrl expert. While the percentage is low, it still accounts for approximately 1083 high importance errors per data set, requiring a significant amount of time and resources to resolve.

Correlation errors (BLUE) were rare, with an average of just 0.01% (min 0%, max 0.04%). These errors span across multiple fields where syntactically valid data was entered, but did correlate semantically, e.g. ‘Sex’ value ‘Male’ and ‘Civil Status’ value ‘Spinster’. Due to the nature and rarity of these errors, the rectification can be completed quickly.

VI. MODEL-DRIVEN APPROACH: WEB APPLICATION AND GENERAL FORM CLASSIFIER

The work and analysis reported so far concerned the treatment of data received through the first phase, where data

was manually input through Excel. We implement a more scalable and systematic approach for the future by developing a bespoke Historical Web Application using MDD processes. Through this web application, users will enter a record’s data directly into 63 specific fields, thus preventing many formatting mistakes due to the coarse granularity. The data is automatically stored in an underlying relational database, currently Postgres. Local data checks performed by the application relate to the properties of each field. Syntactic restrictions connected to each field’s data-type ensure type correctness (e.g numeric values for time, alphabetic values for names) and that certain input fields may not be empty. However, semantic errors can still occur and require immediate identification at data capture time to preempt the need for further cleaning.

We use a new *Classifier DSL* for error detection, embedded in the application’s back-end.

A. The Boolean Classifier as a DSL

Semantic checks are implemented as a DSL with ADD-Lib [7], an MDD tool for the definition of classifiers which produces code that is easy to embed in the Historian Web Application in a service-oriented fashion. In our case, the rules and correctness checks specified by the historians are input in ADD-Lib and compiled into a library of classifiers. These classifiers are then embedded in the Historian Web Application, where they work as an embedded verifier: They

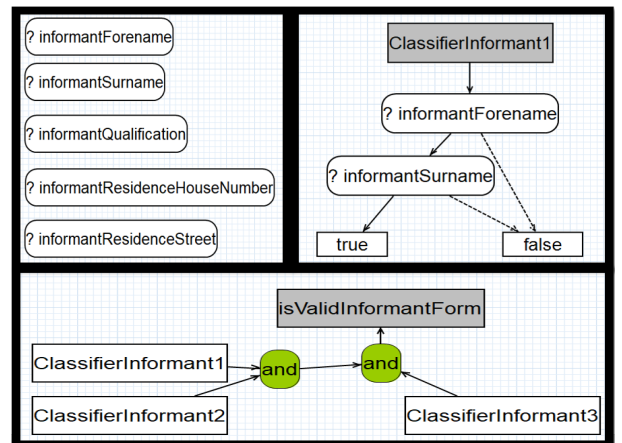


Fig. 3. Informant Classifier Model Example: Predicates (top-left), Binary Decision Diagram (top-right), Classifier Composition (bottom)

Fig. 4. Informant Form from the Historical Web Application

run a set of runtime checks whenever a user inputs a new record.

The ADD-Lib tool is itself a Cinco-product [32] that provides an open-source Java framework for Decision Diagrams (DDs) built upon the CUDD library [33], a standard library for the use and manipulation of Decision Diagrams. The important feature for us is that its user-friendly, graphical interface allows non-programmers to work proficiently with such data structures on the basis of very understandable basic predicates, like `informantSurname`, `informantResidenceStreet` (see Fig.3 top left). ADD-Lib provides an intuitive graphical composition of predicates into, firstly, Boolean decision structures (top right), and then full classifiers (bottom). A detailed description of this process, principles and the tool’s architecture is available in [7].

The user’s model of the classifier is automatically reduced to a canonic DD, this way producing a minimal directed acyclic graph (DAG) structure that expresses the classifier. Efficient Java code is automatically generated [34] based on the canonical structure of the user’s model, with this code embedded in the Historian Web Application, to provide an immediate check at runtime while the user (volunteer) inputs the data record by record. In this way errors can be spotted while the data is being entered, and the volunteers can recheck and correct many more errors than with the offline procedure used before. This new ability to perform runtime checks will greatly reduce the manual intervention by the DBDirl experts.

B. Implementing the Classifiers

To decide whether an input is correct or erroneous we use Boolean logic. For our Web Application it is convenient to define four classifiers, one per page needed to input the full data for an individual record: General Form, Registration Form, Informant Form, and Deceased Form (see Fig.4). This way each page can be checked independently and validated

or corrected immediately. The canonical reduction of these classifiers produces four separate Reduced Ordered Decision Diagrams (ROBDDs), and the ADDLib auto-generates the Java code for each classifier model.

The parser of the Data Cleaning phase already contained an implementation of the logic needed for each individual data property. That logic was continuously updated in collaboration with the domain experts along new insights. We extracted that logic and transformed it into a Java service that takes record entries which instead of checking for errors, states whether an entry is valid or not, returning a Boolean value for each property. This Boolean vector is then checked against the “golden vector”, produced by the ADD-Lib classifier, defining the complete characterization of a perfect valid data entry form, ready to be stored in the main database.

VII. CLASSIFICATION RESULTS

To analyse the quality of the ADD-Lib classifiers, we tested them against the 12 data sets obtained from the student Hackathon, after the Cleaning Phase. Data fields were processed to replicate each Web Application form, and then sent to that form’s corresponding classifier.

Table IV shows the results of each of the four classifiers. There,

- **Number of Entries** is the number of rows available per data set.
- **Classifier Positive Results** is number of positive entries per respective classifier:

$$Error_{i,j} = \frac{PositiveClassifications(i)}{NumberOfEntries(j)} \times 100 \quad (2)$$

Here, $i = [0..3]$ indicates the applied classifier, and $j = [0..11]$ indicates the data set.

- **Time to Classify** reports the run time (in ms) to classify all entries per data set.

We processed the same 12 data sets, and report the results in Table IV. The highly incomplete data set 8 contains only 11 of the 63 properties after parsing and is clearly anomalous. Thus we discuss the results as averaged across the remaining 11 data sets, which are representative of the original data, with in average 618.81 data entries, each with 63 fine granular properties.

Table IV shows that the *General Form classifier*, a single BDD checking Group Registration IDs and their correlation to the TIFF file path batch values, records a high level of acceptance with an average of 99.01% positives. This is to be expected, because it concerns the most uniform and standardized set of fields.

The next highest rate of positive results comes from the *Registration Form classifier*, averaging 36.81%. This is a dramatic drop in acceptance, despite the BDD checking only two complex fields: *Date of Registration* and *Name of Registrar*.

The rate of positive results falls further to 20.02% for the *Informant Form classifier* (9 fields checked across 3 BDDs)

TABLE IV
CLASSIFICATION OF EXISTING DATA SETS USING BOOLEAN CLASSIFIERS

<i>Data Sets</i>	Data Set 0	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Data Set 6	Data Set 7	Data Set 8	Data Set 9	Data Set 10	Data Set 11	*Averages
Number of Entries	621	620	620	620	620	620	620	620	620	620	620	606	618.81
<i>Classifier Pass Results (%)</i>													
General Form ($i=0$)	99.83	99.67	99.83	100.0	100.0	100.0	99.83	100.0	100.0	99.83	100.0	90.26	99.01
Deceased Form ($i=1$)	9.17	6.77	3.54	7.90	2.74	6.29	0.00	12.25	0.00	1.77	11.29	2.97	5.89
Informant Form ($i=2$)	40.41	20.12	22.25	17.09	4.03	20.48	14.51	26.93	0.00	20.96	25.64	9.73	20.02
Registration Form ($i=3$)	50.24	52.25	32.25	33.38	32.25	15.80	24.03	45.32	0.00	31.77	53.87	33.66	36.81
<i>Time to Classify (ms)</i>													
All Entries	6761	6739	6781	6827	6790	6774	6765	6777	6748	6815	6760	6579	6760.72

and to a low of 5.89% for the *Deceased Form classifier* which comprises all the remaining fields across 14 BDDs.

As we see, decrease in positive results is related with the number of fields (thus more BDDs in the relative classifiers), as well as greater complexity of the input fields themselves. Not surprisingly, the most incomplete/non-conformant data concern the deceased: with the highest number of data points, the likelihood that the original record is missing some data, of illegible or damaged data, or true input errors is higher than for other entities like the registrar, that has only two highly regular fields. The same registrar may be in employment for many years, appearing multiple times across various records, thus that data has a high regularity and low variance.

If we were to use a full compliance classifier, applying an atomic nature to our data quality assurance measures to accept only positive results across all four classifiers, we would have an upper bound of only 5.89% perfectly compliant entries, set by the Deceased Form classifier. This percentage would likely fall further when checked by the remaining three classifiers. This indicates that the constraints within our classifiers may be too restrictive at present and perhaps need weakening. In practice, manual transcription of handwritten records can be difficult, and we must be able to adapt to such situations, not immediately ruling out valid data entries due to seemingly minor data entry errors or missing values relating to fields that may not be of high importance. The balance between gathering high quality data and the ease of use from a user-experience perspective will have to be examined.

VIII. CONCLUSIONS AND FUTURE WORK

From a point of view of computational archival science, we see that working with digital records can be an arduous task even if a first partial digitization has taken place. In the case of the GRO data, an excel spreadsheet with a pointer to the TIFF files of the original register page could not be deemed a meaningful resource from which researchers could aggregate desired knowledge, with quantifiable data still in the TIFF image thus not accessible in textual form.

The work on fixing the issues in the data sets collected in Excel so far is a time consuming and arduous process. Once it was assessed that the highly automated Transkribus approach did not work to an acceptable level, with the Excel based approach also proving to be error prone, we decided to elicit

as much specialized knowledge as possible from the initial project phases to form a precise definition and refinement of the data specification. This knowledge and specification then led to the efficient, automated and scalable classifiers used in the new Historian Web Application, that is now going to be used for further data collection. From a historian’s point of view, many of these data elements are present also in other historical data collections, such as the census data of 1901 and 1911, and we plan to reuse large parts of the classifiers when later dealing with those collections.

In the absence of a born-digital archive, if we want to support ease of access allied with new ways for both researchers and public to engage with archival material, we need to provide a high quality, verified and complete data set. Which computational methods are best adopted or adapted for an affordable and reliable large-scale digitization of analogue archives is still largely a matter of design and debate. Historical registers, with fragmentary alphanumeric records populated with various handwritings of registrars or informants, are certainly different from ancient manuscripts copied by scribes in terms of suitability for today’s ML approaches. In our case, the transdisciplinary collaboration in the project has seen historical and archival thinking inform and lead the computational thinking and IT competence: first towards the IT-rectification of domain specific knowledge in terms of the formulation and implementation of the first classifiers, then in a second phase towards addressing efficiency and scalability. The existence and the results of the “ideal” ADD-Lib classifier on existing data show the need for real-time IT-supported guidance in the data capture, as happens now in the Historian Web Application, and the benefit of immediate verification as the data is being entered in a manual transcription process.

A full integration of this service into the Historian Web Application will ensure higher accuracy rates from data Hackathons, improve the usefulness and the success rate of this crowd-based contribution to a long lasting corpus of high quality data. Furthermore it involves the creation of a reusable “Classifier DSL” for these data types that would have never been built without this project, thus benefiting a number of data collections from those years, and incorporating a growing library of classifiers and error detectors as services in the Web Application design framework, for further extension and reuse. Part digitalization and limited access beleaguers the

archival space, and there are several projects that would benefit from the tools, methods and outputs of this project. The DSL comprises so far the Java service and the auto-generated code implementing the four classifiers and their constituent BDDs, with work continuing on the identification and development of suitable extensions.

Many of the errors detected by the ADD-Lib classifiers are due to missing fields in the original records: this shows how historical data incompleteness can dramatically reduce the subset of entries available for research on specific issues. The, sometimes much smaller, eligible ‘valid data set’ may possibly fail to represent the full reality. We also likely need a different provision to distinguish in the classifiers originally missing data from data capture issues.

The challenge, and medium and long term goal of the IT component of this project, is indeed to build over time a Web based Digital Humanities platform for researchers and the public, for easy consultation and research on these historical and, once digitized, high quality digital data archives.

ACKNOWLEDGMENT

We are grateful for the full cooperation of the Registrar General of Ireland for permission to use these data for research purposes. This research is funded by the Irish Research Council Laureate Award 2017/32 and by Science Foundation Ireland through the grants 13/RC/2094 to Lero - the Irish Software Research Centre (www.lero.ie) and 18/CRT/6223 to the Centre for Research Training in Artificial Intelligence.

REFERENCES

- [1] C. Breathnach, N. M. Ibrahim, S. Clancy, and T. Margaria, “Towards model checking product lines in the digital humanities: An application to historical data,” in *From Software Engineering to Formal Methods and Tools, and Back*, pp. 338–364, Springer, 2019.
- [2] M. Mernik, J. Heering, and A. M. Sloane, “When and how to develop domain-specific languages,” *ACM computing surveys (CSUR)*, vol. 37, no. 4, pp. 316–344, 2005.
- [3] T. Margaria, B. Steffen, and M. Reitenspieß, “Service-Oriented Design: The Roots,” in *Proc. of the 3rd Int. Conf. on Service-Oriented Computing (ICSOC 2005), Amsterdam, The Netherlands*, vol. 3826 of *LNCS*, pp. 450–464, Springer, 2005.
- [4] F. Gossen, T. Margaria, A. Murtovi, S. Naujokat, and B. Steffen, “Dsls for decision services: A tutorial introduction to language-driven engineering,” in *ISoLA 2018, Limassol, Cyprus, Proceedings, Part I*, pp. 546–564, 2018.
- [5] S. Jörges, C. Kubczak, F. Pageau, and T. Margaria, “Model Driven Design of Reliable Robot Control Programs Using the jABC,” in *Proceedings of 4th IEEE Int. Worksh. on Engineering of Autonomic and Autonomous Systems (EASE 2007)*, pp. 137–148, 2007.
- [6] S. Boßelmann, M. Frohme, D. Kopetzki, M. Lybecait, S. Naujokat, J. Neubauer, D. Wirkner, P. Z Weihoff, and B. Steffen, “DIME: A Programming-Less Modeling Environment for Web Applications,” in *Proc. ISoLA, Part II*, vol. 9953 of *LNCS*, pp. 809–832, Springer, 2016.
- [7] F. Gossen, A. Murtovi, P. Z Weihoff, and B. Steffen, “Add-lib: Decision diagrams in practice,” *arXiv preprint arXiv:1912.11308*, 2019.
- [8] J. Marti-Henneberg, X. Franch-Auladell, and J. Solanas-Jiménez, “The use of digital tools for spatial analysis in population geography,” *Frontiers in Digital Humanities*, vol. 3, p. 9, 2016.
- [9] T. Fortibuoni, S. Libralato, E. Arneri, O. Giovanardi, C. Solidoro, and S. Raicevich, “Fish and fishery historical data since the 19th century in the adriatic sea, mediterranean,” *Scientific data*, vol. 4, p. 170104, 2017.
- [10] WHO, “Global health and ageing - world health organization,” *Ageing and life-course*, vol. 32, October 2011.
- [11] G. Manogaran, C. Thota, D. Lopez, and R. Sundarasekar, “Big data security intelligence for healthcare industry 4.0,” in *Cybersecurity for Industry 4.0*, pp. 103–126, Springer, 2017.
- [12] G. H. Elder, M. K. Johnson, and R. Crosnoe, “The emergence and development of life course theory,” in *Handbook of the life course*, pp. 3–19, Springer, 2003.
- [13] M. Golfarelli and S. Rizzi, “A model-driven approach to automate data visualization in big data analytics,” *Information Visualization*, vol. 19, no. 1, pp. 24–47, 2020.
- [14] WHO, “Who, the global health observatory. explore a world of health data.” <https://www.who.int/data/gho>, 06 August. 2020.
- [15] K. S. Khan, D. Wojdyla, L. Say, A. M. Gülmezoglu, and P. F. Van Look, “Who analysis of causes of maternal death: a systematic review,” *The lancet*, vol. 367, no. 9516, pp. 1066–1074, 2006.
- [16] C. AbouZahr, D. De Savigny, L. Mikkelsen, P. W. Setel, R. Lozano, E. Nichols, F. Notzon, and A. D. Lopez, “Civil registration and vital statistics: progress in the data revolution for counting and accountability,” *The Lancet*, vol. 386, no. 10001, pp. 1373–1385, 2015.
- [17] P. Mahapatra, K. Shibuya, A. D. Lopez, F. Coullare, F. C. Notzon, C. Rao, S. Szreter, *et al.*, “Civil registration systems and vital statistics: successes and missed opportunities,” *The Lancet*, vol. 370, no. 9599, pp. 1653–1663, 2007.
- [18] M. Brambilla, J. Cabot, and M. Wimmer, “Model-driven software engineering in practice,” *Synthesis lectures on software engineering*, vol. 3, no. 1, pp. 1–207, 2017.
- [19] A. R. Da Silva, “Model-driven engineering: A survey supported by the unified conceptual model,” *Computer Languages, Systems & Structures*, vol. 43, pp. 139–155, 2015.
- [20] T. Margaria and B. Steffen, “extreme model-driven development (xmdd) technologies as a hands-on approach to software development without coding,” *Encyclopedia of Education and Information Technologies*, pp. 732–750, 2020.
- [21] D. Withers, E. Kavas, L. McCarthy, B. Vandervalk, and M. Wilkinson, “Semantically-guided workflow construction in Taverna: the SADI and BioMoby plug-ins,” in *ISoLA 2010 - Part I*, vol. 6416 of *LNCS*, pp. 301–312, Springer Berlin / Heidelberg, Oct. 2010.
- [22] T. Margaria and A. Schieweck, “The digital thread in industry 4.0,” in *International Conference on Integrated Formal Methods*, pp. 3–24, Springer, 2019.
- [23] D. Breuker, “Towards model-driven engineering for big data analytics—an exploratory analysis of domain-specific languages for machine learning,” in *2014 47th Hawaii International Conference on System Sciences*, pp. 758–767, IEEE, 2014.
- [24] C. Ginzburg, J. Tedeschi, and A. C. Tedeschi, “Microhistory: Two or three things that i know about it,” *Critical Inquiry*, vol. 20, no. 1, pp. 10–35, 1993.
- [25] B. Steffen, F. Gossen, S. Naujokat, and T. Margaria, “Language-Driven Engineering: From General-Purpose to Purpose-Specific Languages,” in *Computing and Software Science: State of the Art and Perspectives*, vol. 10000 of *LNCS*, Springer, 2018.
- [26] D. Firmani, M. Maiorino, P. Merialdo, and E. Nieddu, “Towards knowledge discovery from the vatican secret archives. in codice ratio-episode 1: Machine transcription of the manuscripts,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 263–272, 2018.
- [27] T. T. at University of Innsbruck, “Transkribus. collaborate. share...” <https://readcoop.eu/transkribus/>, info@readcoop.eu.
- [28] P. by Alexander Thom for H.M.S.O., “A statistical nosology — ireland.” https://wellcomelibrary.org/item/b30566095_b30566095.
- [29] M. Ziemann, Y. Eren, and A. El-Osta, “Gene name errors are widespread in the scientific literature,” *Genome biology*, vol. 17, no. 1, pp. 1–3, 2016.
- [30] P. Kamakshi, “Importance of big data in healthcare system-a survey,” *International Journal of Applied Engineering Research*, vol. 13, no. 15, pp. 12184–12187, 2018.
- [31] F. Gossen and B. Steffen, “Large random forests: Optimisation for rapid evaluation,” *arXiv preprint arXiv:1912.10934*, 2019.
- [32] CINCO, “Scce meta tooling framework..” <https://cinco.scce.info>, Last Accessed 2020-08-22.
- [33] F. Somenzi, “Cudd: Cu decision diagram package,” <http://vlsi.colorado.edu/~fabio/CUDD/>, 1997.
- [34] F. Gossen, T. Margaria, and B. Steffen, “Towards explainability in machine learning: The formal methods way,” *IT Professional*, vol. 22, no. 4, pp. 8–12, 2020.