Parameter extraction from technical datasheet documentation using technique optimized through genetic algorithm

Angel Marinov
Electronics and microelectronics
Technical University of Varna
Varna, Bulgaria
a.marinov@tu-varna.bg

***Abstract*— The paper presents an approach for extracting specific parameters from technical datasheet documentation. The extraction is aimed at various electrical properties of different electronic components. The presented approach is part of an automated data gathering system that is used for the composition of a complex database of electronic components. Key variables of the proposed approach are optimized through the implementation a genetic algorithm. The paper presents the specifics and challenges related to the extraction, the steps taken to resolve them and optimize accuracy, as well as verification and assessment applied over 8000 documents.**

Keywords— Data collection, Datasheet, Genetic algorithm, Optimization, Text extraction,

I. INTRODUCTION

Data science and artificial intelligence (AI) are modern concepts that describe various applications and algorithms that can be used for solving different and complex problems. A common feature and specific of those techniques is the requirement for data that can be used for training, testing and analysis. Quantity and quality of the data are crucial for most applications. [1,2]

With the development of modern computer and information systems, data science and AI are more often used in engineering sciences for decision making, optimization, design automation and generation, etc. This is true for most engineering fields including electrical and electronic engineering [3,4]. A problem for the wider spread and effective use of that type of applications however is related to the availability of complete and detailed datasets with technical information for various components and devices. Although available, this type of information is usually suited more for human reading and processing, rather than a machine one. There are some solutions for similar text extraction relevant to technical datasheets of electronic components [5]. Those that were reviewed however, deal with information extraction from the document as an image.

In this regard the paper presents an automated algorithm for composition of datasets for the electrical parameters of electronic components. These datasets can be used for training and verifying various AI based algorithms for automated design and optimization of electronic topologies.

For most electronic components technical information is contained in datasheet files, usually in pdf format. Thus, the algorithm in its basis is a text extraction software, optimized to extract specific parameters that have high usage and application in design of electronic topologies and circuits.

In order to improve efficiency and accuracy for the given task, the suggested text extraction software is optimized trough the implementation of a dedicated genetic algorithm (GA) [6,7,8].

In this paper, section 2 summarizes the challenges that were observed, while sections 3 and 4 discuss the algorithm used to resolve them and how it can be optimized to provide better accuracy. Verification and accuracy assessment are presented in section 5, while conclusions and suggestions for future work are presented in section 6.

II. CHALLENGES TO PARAMETER EXTRACTION

In order to explain the motivation related to the suggested algorithm and its optimization, the various challenges that were faced during attempts to automate parameter extraction from technical datasheets, have to be discussed. The following discussions are on the basis that information from the datasheets is extracted as a single string per page**.** In addition, for simplification purposes presentation is made on the basis that datasheet of Metal Oxide Silicon Field Effect Transistors (MOSFETs) are used. All conclusions and suggestions, however, should be valid for most other datasheets of electronic semiconductor components.

A. Document presentation

In datasheets of electronic components, parameters of interest are usually presented as tables. An example for such a presentation is given in Table 1. As it can be seen parameters are described and have an appointed symbol. The table then contains the conditions at which measurements are taken, followed by the value that is to be extracted for the specific parameter and its unit. Parameters can be presented through their minimum, typical and maximum values, where availability of the aforementioned can vary, depending on the device and the specific manufacturer.

This type of presentation is usually consistent for different documents and manufactures, with minor variances that are described in the following sub-sections.

B. Tables and parameter positioning

A significant problem for the correct interpretation of the data and more specifically value extraction is that the values themselves are not positioned next to their corresponding text and symbol values. As it can be seen from Table I, values related to the conditions at which measurements were taken are positioned between the parametric and symbol fields and the value field. In addition, parameters may have three different values – minimum, typical and maximal. It should be

978-1-7281-9308-3/20/$31.00 ©2020 IEEE

TABLE I. EXAMPLE OF DATA TABLE IN AN OBSEREVED DATASHEET

Parameter	Symbol	Conditions	Values			Unit
			min.	typ.	max.	
Dynamic characteristics						
Turn-on delay time	$t_{d(off)}$	V_{DD}=300 V, V_{GS}=10V, I_D=0.01 A, $R_{G,ext}$=6Ω	-	***6.1***	***19***	***ns***
Rise time	t_r		-	9.7	14.5	
Turn-off delay time	$t_{d(off)}$		-	14	21	
Fall time	t_f		-	115	170	

clear when extracting which value is being assigned to the specific parameter of interest.

C. *Variaty of parameter definitions, parameters as part of words*

For some of the observed datasheets it was noted that:

- Names in the parameter fields can vary, with additional words and different capital and lower letters.
- The places of the symbol and the parameter fields can be reversed.
- As the symbols are usually 2 to 4 letters long they can be part of other words or parameters, in the given example this is especially true for the parameter t_r

Those irregularities often pose a problem and in conventional text extraction algorithms increase the recognition error.

D. *Irregular text extaction*

For some of the observed datasheet, based on the formatting and the encoding of the files, it was noted that in the extracted string values appear before or in between parameters and symbols, rather after them.

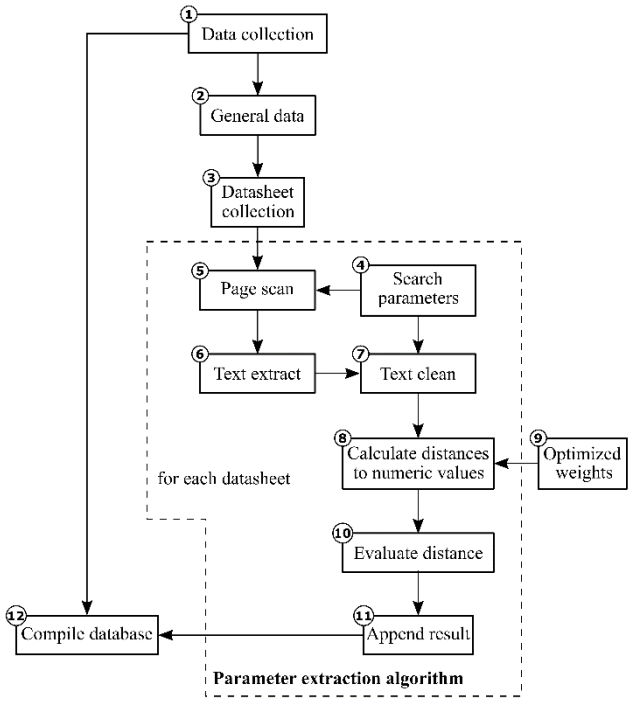

Fig. 1. Flowchart of the suggested algorithm

III. SUGGESTED APROACH

In order to answer to the challenges described in the previous section and allow for automated extraction without the need of specialized human addressed text cleaning or algorithm adjustment, a dedicated approach was developed. The flow chart of the suggested approach is presented at Figure 1. The approach is split into 12 distinct steps, which can be described as follows:

- **Step 1**, **Step 2** and **Step 3** – those steps are outside the core of the suggested text extraction algorithm and involve initial data acquisition form web sources – usually online vendor. This acquisition involves gathering some general available parameters for the devices and downloading the datasheets themselves.
- **Step 4** - Definition of the parameters that are object of interest for extraction. The definitions required for the approach to work include: a) the parameter name, b) the parameter symbol and b) the parameter units – if more dimensions are expected they have to be included, for example – ns, µs, ms, etc.
- **Step 5.** Pages of the datasheet are scanned for the defined parameters.
- **Step 6.** If a parameter is detected the text from the page is extracted as a string.
- **Step 7.** Text clean is applied removing unnecessary data. The text clean leaves only numeric values and the defined parameters, symbols and units. In addition, numeric values that can be classified as data related to the measurement conditions are removed. This is done under several assumptions, mainly removing numeric objects placed after the '=' sign (see Table I.). A sample cleaned text is presented at figure 2.
- **Step 8**, **Step 9** and **Step 10**. The distance between each numeric value and respectively the parameter field, the symbol field and unit field are calculated. In the calculation, weighted Euclidian distance was used – the distance is calculated based on:

$$dist = \sqrt{W_p.(d_p - d_n)^2 + W_s.(d_s - d_n)^2 + W_u.(d_u - d_n)^2} \quad (1)$$

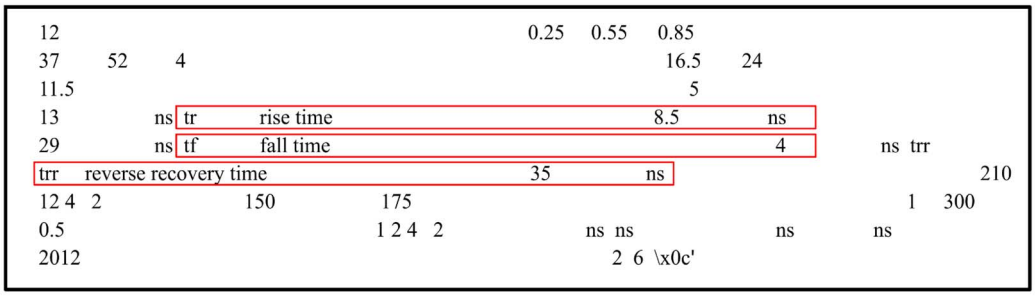

Fig. 2. Sample text after cleaning

Where: W_p, W_s and W_u are weights that respectively place importance on the parameter field, the symbol field or the unit field; d_p, d_s, d_u and d_n are the locations in the string of respectively the parameter field, the symbol field and the specific numeric value for which the distance is calculated. Then the distances relevant to each numeric value are sorted. The value with the lowest distance should be the value of interest of the given parameter. It can be seen from Figure 2 that the value of interest is at specific distances from each of its descriptors. By placing importance trough, the weighs, it is possible for the algorithm to correctly extract despite the challenges described in the previous section. The weights can be optimized to disregard remnants from the cleaned condition parameters and to specifically target minimum, typical or maximum values. The way the weights are optimized is discussed in the next section.

- **Step 11** and **Step 12**. The extracted result is appended to the database, together with any useful information obtained during the execution of step 1.

IV. Optimization through Genetic Algorithm

In order to optimize the weighs described in the previous section (step 9, figure 1) a dedicated GA is used. The flow chart of the applied GA is presented at Figure 3.

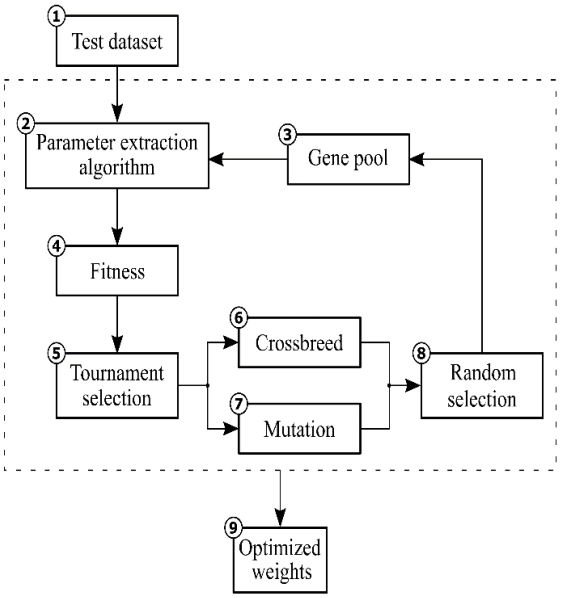

Fig. 3. Flowchart of the GA

The GA algorithm has 9 distinct steps. They can be described as follows:

- **Step 1.** The total number of documents to be observed is 8000. From this data set a training set was generated by randomly selecting 200 members for which several of the parameters to be observed were extracted by a human operator. For the optimization three parameters were observed.
- **Step 2** and **Step 3**. A chromosome pool is generated. Each chromosome contains three genes that correspond to each of the three weights. Parameters are extracted from each document for each chromosome.
- **Step 4.** The fitness of each extraction is evaluated. The fitness function estimates the extraction accuracy over all of the observed data.
- **Step 5, Step 6, Step 7 and Step 8.** Tournament selection, single point crossbreed, mutation and random selection is applied. The chromosome pool is refreshed, and all the steps are repeated until convergence or set number of epochs is reached.
- **Step 9.** Optimized weights are obtained and used in the main algorithm (Figure 1)

V. Verification and Accuracy Estimation

The presented approach is developed into a software script in the Python programing language. Text cleaning is conducted using Regular Expression operations [9]. The genetic algorithm optimization is performed using in-house developed functions. When evaluating the accuracy, errors in each extracted parameter are accounted for. During the optimization and accuracy testing, three parameters per document are being extracted.

A. Accuracy before and after optimization.

As described in order to optimize the algorithm 200 documents are taken as a training set. For those documents parameters are extracted and recorded by a human operator. The GA was set to target typical values of the parameters of interest. When testing by using this approach the base accuracy of the algorithm – when all weights are set to 1 - is approximately **0.83**. Accuracy after running the genetic algorithm over 10 epochs is presented in Table II. Results are floored to the 2 number after the floating point.

TABLE II. RESULTS FROM THE GA OPTIMIZATION

Epoch	Accuracy	Epoch	Accuracy
1	0.83	6	0.9
2	0.85	7	0.91
3	0.86	8	0.95
4	0.86	9	0.99
5	0.87	10	0.99

B. Final accuracy

The algorithm with the optimized parameters was applied to all of the members of the database, where the parameters selected for extraction were different for then the ones used for optimization. At random 50 members were picked and crossed checked. Accuracy for those 50 members was evaluated at approximately **0.99**. Confirming the results obtained during the GA optimization.

VI. CONCLUSIONS

The paper presents a specialized algorithm for text extraction from technical documentation and more specifically, datasheets of electronic components.

Due to the formatting and data presentation specifics, accurate extraction can be a challenge. Thus the presented algorithm uses dedicated GA based optimization.

Optimization, testing and verification was done using over 8000 different files, with over 250 members picked at random to check accuracy. The top accuracy of the algorithm after optimization was estimated at about 0.99. This can be considered sufficient and can be used for database compilation.

The algorithm was successfully implemented for the compilation of a MOSFET database. This database was used in several AI based algorithms for optimization and generation of topologies of power electronic converters.

Future work on the algorithm may include the use of several set of weights and distance calculators in order to extract all types of values – minimum, typical and maximal as well as the conditions under which they were recorded.

ACKNOWLEDGMENT

The current study is funded by budgetary subsidies of the Technical University of Varna, allocated for research and development activities related to project “НП4/2020“.

REFERENCES

[1] M. Kantarcioglu and F. Shaon, "Securing Big Data in the Age of AI," 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Los Angeles, CA, USA, 2019, pp. 218-220, doi: 10.1109/TPS-ISA48467.2019.00035.

[2] AFan, W. 2015, "Data quality: From theory to practice", SIGMOD Record, vol. 44, no. 3, pp. 7-18.

[3] D. R. Williams, M. P. Foster, C. M. Bingham and D. A. Stone, A genetic algorithm for designing LCLC resonant converters, 2008 4th IET Conference on Power Electronics, Machines and Drives, York, 2008, pp. 732-736.

[4] H. Helali et al., Power converter's optimisation and design. Discrete cost function with genetic based algorithms, 2005 European Conference on Power Electronics and Applications, Dresden, 2005, pp. 7 pp.-P.7.

[5] M. Traquair, E. Kara, B. Kantarci and S. Khan, "Deep Learning for the Detection of Tabular Information from Electronic Component Datasheets," 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1-6, doi: 10.1109/ISCC47284.2019.8969682.

[6] Goldberg, D.E. & Holland, J.H. 1988, "Genetic Algorithms and Machine Learning", Machine Learning, vol. 3, no. 2, pp. 95-99.

[7] Sastry, K., Goldberg, D.E. & Kendall, G. 2014, "Genetic algorithms" in Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Second Edition, pp. 93-118.

[8] Harik, G.R., Lobo, F.G. & Goldberg, D.E. 1999, "The compact genetic algorithm", IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 287-297.

[9] R. Chowdhury, M. R. Babu, V. Mishra and H. Jain, "Regular expressions in big data analytics," 2017 International Conference on Intelligent Computing and Control (I2C2), Coimbatore, 2017, pp. 1-10, doi: 10.1109/I2C2.2017.8321906.