

Optimization of Edit Distance Algorithm for Sanctions Screening Risk Score Assessment



Rogie Nino¹, Ariel Sison, Ruji Medina

Graduate Program of Technological Institute of the Philippines
Quezon City, Philippines

ABSTRACT

Evaluating Risk Score Assessment for sanctions screening is necessary to calculate and gauge the risk rate of data elements involved during screening. It includes string matching process in reviewing sanctions lists to check if any investor in a fund is involved in fraud by matching the investor information (as Stop Descriptor) with the Sanctions List which contains the names of individuals who are known to be involved in financial crime or terrorism. This paper will present the inherent capability of Edit Distance algorithm or the Damerau-Levenshtein (DL) Distance algorithm to address many common misspellings and typos in string matching through insertion, deletion, transposition and substitution which are considered as a significant component of fuzzy possible success rating used in Sanction Screening. The paper also aims to optimize the DL Distance Algorithm by applying the theories of phonetic algorithm which expected to provide big impact on speed performance problem of computing the edit distance of two longer strings.

Key words: Fuzzy Logic, Sanctions Screening, Stop Descriptor, Damerau- Levenshtein Distance

1. INTRODUCTION

Sanctions screening is the process of reviewing sanctions lists to check if any investor in a Fund is involved in financing crime or terrorism [1-3]. The insurance industry plays an equally important role in combating financial crime and promoting international security. Life insurance policies and annuities have been used in money laundering and the financing of terrorism [4]. Different algorithms are being used in Sanctions Screening and one of this is Fuzzy Logic. The Fuzzy Logic is based on the Edit Distance algorithm (or Damerau-Levenshtein Distance algorithm) in which using Stop Descriptors; the algorithm can address many common misspellings and typos such as insertion, deletion, substitution and transposition. The Damerau-Levenshtein distance algorithm is a popular method of fuzzy string matching. It is a string metric for measuring the difference between two sequences [5]. The likelihood of a match being either “true match” or “false positive” during sanctions \

screening process will depend on the Risk Score Assignment (RSA). It will be the basis for each match being generated during the screening process. It compares the data elements or a component involved with the match and based on those elements’ similarities/differences produces a highly-tunable score that represents the similarity of a match.

There were several studies conducted are related to Damerau-Levenshtein distance algorithm. Most of the previous research and modifications aid the actual DL distance algorithm to help improve the speed of string comparisons by adding tools such as computer memory, others develop a pre-evaluation method prior to edit distance metric processing to decrease the computation required of a string matching and integrate code theory which tasks to correct some of the code operation in computing edit distance. Applying the theories of phonetic algorithm in DL distance algorithm will provide big impact on speed performance problem of computing the edit distance of two longer strings to mitigate the known problem of Edit Distance algorithm.

This paper aims to enhance the Edit Distance (Damerau–Levenshtein) algorithm to improve its speed performance of measuring the edit distance between strings with longer values by adding additional operation to compare its phoneticity likelihood during string matching. Result will be compared to original Damerau-Levenshtein distance to validate the improvement in terms of speed during the string matching and calculating edit distance (using English Language).

The rest of this paper is organized as follows: Section 2 is discussion of some related works to the Damerau–Levenshtein Distance Algorithm enhancement; Section III presents and describes the proposed Optimized Edit Distance Algorithm and it’s results string comparison is provided; lastly Section IV concludes the paper and recommended study for future works.

2. REVIEW OF RELATED LITERATURE

This section presents some related works that are relevant to the Sanctions Screening, Damerau-Levenshtein Distance algorithm and Damerau-Levenshtein Distance algorithm modification and enhancements.

2.1 Sanctions Screening

Sanctions Screening (SS) is the process of reviewing sanctions lists. These sanctions lists are also referred as “watchlists”. A compilation of multiple regulatory and enhanced due diligence lists from all major sanctioning bodies around the world, including global lists such as OFAC, UN sanctions, EU sanctions, HM Treasury and PEP, and in-country lists [6]. The sanction lists may contain different entities to ensure that there is no a breach of Country Sanctions Programs by considering the following information during sanction screening validation: 1) Extent of policy coverage; 2) Country of residence for an individual 3) Name, address, and date of birth; 4) Nationality of an individual; 5) Country of registration for a company; 6) Domicile of a company; 7) Location of loss; 8) Destination of travel; 9) Shipments to and from and transits through sanctioned countries; and 10) Recipient of goods if cargo is to be delivered. Currently, some of the companies are limiting the number entities to be checked in sanctions screening particularly to those with high volume number of policies and claim transactions being screened and its risk score assignment may vary depends on the risk score rating result of the identified entities or components.

2.2 Edit Distance and Damerau-Levenshtein Distance Algorithm

Edit distance is a measure of similarity between two strings evaluated based on the minimum number of operations required to transform one string into the other while Damerau–Levenshtein distance between two words is the minimum number of operations (consisting of insertions, deletions or substitutions of a single character, or transposition of two adjacent characters) required to change one word into the other. The **Damerau–Levenshtein** distance differs from the classical Levenshtein distance by including transpositions among its allowable operations in addition to the three-classical single-character edit operations (insertions, deletions and substitutions). The said four operations correspond to more than 80% of all human misspellings. Each spelling mistake is a wrong, missing, extra letter, or the wrong type of the order of two different consecutive letters, for example, “ab” typed as “ba” is considered as 1 mistake while it is 2 according to Levenshtein edit-distance [7]. Figure 1 Illustrates the Damerau–Levenshtein distance between two strings a and b a function $d_{a,b}(i,j)$ is defined, whose value is a distance between an i -symbol prefix (initial substring) of string a and

a j -symbol prefix of b. It is where $1_{(a_i \neq b_i)}$ is the indicator function equal to 0 when $a_i = b_i$ and equal to 1 otherwise.

$$d_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} d_{a,b}(i-1,j) + 1 \\ d_{a,b}(i,j-1) + 1 \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_i)} \\ d_{a,b}(i-2,j-2) + 1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i \text{ and } b_{j-1} \\ & \text{and } a_{i-1} \text{ and } b_j \\ \min \begin{cases} d_{a,b}(i-1,j) + 1 \\ d_{a,b}(i,j-1) + 1 \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_i)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 1: Recursive Function Representation of Damerau-Levenshtein Distance

Each recursive call matches one of the cases covered by the Damerau–Levenshtein distance above.

- $d_{a,b}(i-1,j) + 1$ corresponds to a deletion (from a to b).
- $d_{a,b}(i,j-1) + 1$ corresponds to an insertion (from a to b).
- $d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_i)}$ corresponds to a match or mismatch, depending on whether the respective symbols are the same.
- $d_{a,b}(i-2,j-2) + 1$ corresponds to a transposition between two successive symbols.

2.3 Modifications and Enhancements related to Damerau-Levenshtein Distance

The Levenshtein distance algorithm is a popular method of fuzzy string matching. It is a string metric for measuring the difference between two sequences. Other popular measures of edit distance, which are calculated using a different set of allowable edit operations are: 1) the Damerau–Levenshtein (DL) distance allows insertion, deletion, substitution, and the transposition of two adjacent characters [7]; 2) the Longest Common Subsequence (LCS) distance allows only insertion and deletion, not substitution [8]; 3) the Hamming Distance (HD) allows only substitution, hence, it only applies to strings of the same length [9]; and 4) the Jaro distance allows only transposition [10]. These edit distance algorithms can also be computed between two longer strings, but the cost to compute it, which is roughly proportional to the product of the two string lengths, makes this impractical.

With the enhancement implemented in Levenshtein distance algorithm by utilizing the four editing operations (insertion, deletion, substitution and transposition) in measuring the edit distance between string values, the original Levenshtein and Damerau–Levenshtein distance algorithms were used as baseline for further research studies. CUDA based Graphics Processing Unit (GPU) and the newly introduced Unified Memory (UM) was used to speed up the

most common algorithms (Levenshtein and Damerau) to compute the edit distance between two strings [9]. A study focused on the problem of constructing codes capable of correcting a single deletion and a number of adjacent transpositions for the Damerau distance. Researcher extended work includes constructions of codes capable of correcting block adjacent transpositions and deletions [5]. A study contribution was the development of an improved “filter and verify” method (Fast Bitwise Filter (FBF)), which substantially decreases the computation required to compare short alphabetic, numeric and alphanumeric strings for demographic data fields prior to evaluation with an edit distance metric in particular to big data in Cloud [12].

Several studies were also embedded the phonetics algorithm in edit distance. A phonetic algorithm is being used for indexing of words by their pronunciation and was developed for English language; consequently, applying the rules to words in other languages might not give a meaningful result. But, it did not stop researchers to use this algorithm for cross linguistic name matching such as English and Arabic. Standard string comparison measures perform poorly on this task due to varying transliteration conventions in both languages and the fact that Arabic script does not usually represent short vowels. Significant improvement was achieved by augmenting the classic Levenshtein edit-distance algorithm with character equivalency classes [13]. Increase the performance of Editex Algorithm by developing new weighting and distance calculation. The source of mismatching is grouped into phonetic and typographic errors. Characters are divided into groups of phoneticity and typography, which have their own weight. By using this letter grouping, researcher’s proposed method was proven suitable for implementation in homomorphic encrypted data [14]. Another study was developed an application of phonetic encoding for analyzing similarity of patient’s data (Bangladesh perspective) that resolves the underlying problem of misspelled names of patients in healthcare systems and proposed a modified version of NameSignificance algorithm [15].

3. PROPOSED OPTIMIZED EDIT DISTANCE ALGORITHM

Sanctions screening is a process used to screen real-time and or batch file transactions against Watch List to determine if economic and regulatory sanctions are to be applied against a person. The basis of matching depends on the Stop Descriptor (phrase that caused the match) and Match Parameters which includes the Matched Field, Match Type (Watch List to be compared), and Match Score (or Risk Score Assignment). To assess the appropriate RSA, this paper will use the Fuzzy Expert System using the Fuzzy Risk Score Assessment Framework shown in **Figure 2** [16].

Below framework will conform if the following process of developing a fuzzy expert system is applicable.

- a. Specify the problem and define linguistic variables – (Linguistic Assessment)
- b. Determine fuzzy sets – (Translation of Linguistic Assessment)
- c. Elicit and construct fuzzy rules - (Translation of Linguistic Assessment)
- d. Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system – (Fuzzy Number Aggregation and Inference)
- e. Evaluate and tune the system – (Evaluate and Tune the System).

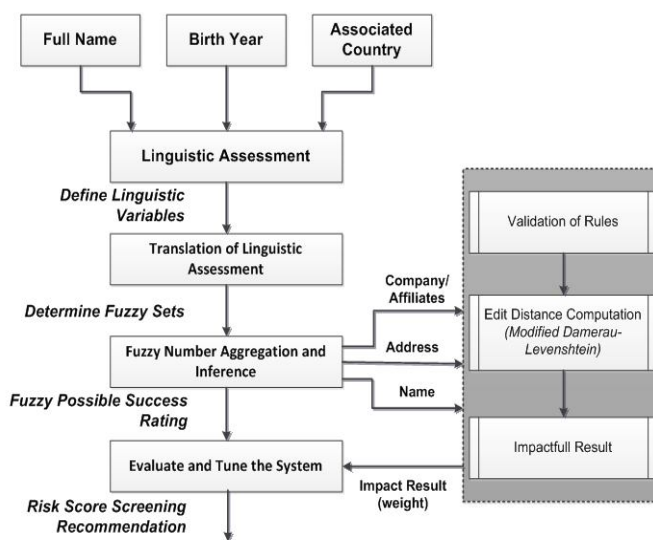


Figure 2: Fuzzy Risk Score Assessment Framework.

The Damerau- Levenshtein Distance will be used in the process of fuzzy inference involves all of the entities that are described in Membership Functions, Logical Operations, and If-Then Rules of Risk Score Assessment Fuzzy Logic in sanctions screening. The development and testing of the application using the Damerau- Levenshtein Distance algorithm will undergo assessment process utilizing metrics or the string *match* (Full Name and Associated Country) weight result.

The Damerau-Levenstein Distance algorithm was implemented in assessing the likelihood of Full Name and Associated Country entities. The principle of modified edit distance will be implemented in matching process to address many common misspellings and typos such as insertion, deletion, transposition and substitution. This is a measure of the similarity between two strings, which we will refer to as the source string (name and associated country) and the target string (Sanctions List or Watch List).

DL Distance Weight Process

Step 1: Check for empty string
 $m = |s|$

```
n = |t|
if m = 0 return n end-if
if n = 0 return n end-if
```

Step 2: Create distance matrix *d*
 for *i* = 0 to *m*: *d*_{*i*,0} = *i* end-for
 for *j* = 0 to *n*: *d*_{0,*j*} = *j* end-for

Step 3: Calculate distance matrix *d*
 for *i* = 1 to *m*
 for *j* = 1 to *n*
 if *s*_{*i*-1} = *t*_{*j*-1}
 *d*_{*i*,*j*} = *d*_{*i*-1,*j*-1}
 else
 *d*_{*i*,*j*} = min (*d*_{*i*-1,*j*}, *d*_{*i*,*j*-1}, *d*_{*i*-1,*j*-1}) + 1
 if *i* > 1 and *j* > 1
 if *s*_{*i*-1} = *t*_{*j*-2} and *s*_{*i*-2} = *t*_{*j*-1}
 *d*_{*i*,*j*} = min (*d*_{*i*,*j*}, *d*_{*i*-2,*j*-2}, +1)
 end-if
 end-if
 end-if
 end-for
 end-for
 return *d*_{*m*,*n*}

The Phonetics Algorithms theories will be embedded in the above algorithm as the additional operation of edit distance computation to mitigate the issue of high cost of computing the two longer strings. A phonetic operation indexes names by sound and matches each character or combination of characters against phonetic encoded group. The phonetic group is expected to handle the phonetic error. Thus, the following considerations will be included in the proposed additional operation. Handling of Phonetic transcription (also known as phonetic script or phonetic notation) will utilize phonetic alphabet or the International Phonetic Alphabet (IPA) for English language.

- a. IPA is designed to represent only those qualities of speech that are part of oral language such as phonemes.
- b. In Table 1, the proposed enhancement may include phonemes with 21 unique sounds. There are sounds help distinguish one word or meaning from another. Various letters and letter combinations known as graphemes are used to represent the sounds.

Table 1: 21 UNIQUE SOUNDS PHONEMES

Sound #	Sound	Graphemes
1	/b/	b, bb
2	/d/	d, dd, ed
3	/f/	f, ff, ph, gh, lf, ft
4	/h/	h, wh
5	/j/	j, ge, g, dge, di, gg
6	/k/	k, c, ch, cc, lk, qu ,q(u), ck, x
7	/l/	l, ll
8	/m/	m, mm, mb, mn, lm

9	/n/	n, nn, kn, gn, pn
10	/p/	p, pp
11	/r/	r, rr, wr, rh
12	/s/	s, ss, c, sc, ps, st, ce, se
13	/t/	t, tt, th, ed
14	/v/	v, f, ph, ve
15	/w/	w, wh, u, o
16	/y/	y, i, j
17	/z/	z, zz, s, ss, x, ze, se
18	/a/	a, ai, au
19	/e/	e, ea, u, ie, ai, a, eo, ei, ae, ay
20	/i/	i, e, o, u, ui, y, ie
21	/u/	u, o, oo, ou

Optimized DL Distance Weight Process

Check for Unique Sounds Phonemes

OrigSearchString = Stop Descriptor

ComparisonString {*OrigSearchString* String with the Sanctions Lists}

SoundexSearchString {Replace all phonetics characters on *OrigSearchString* with the single character using the Unique Sounds Phonemes in Table 1.}

If whole *OrigSearchString* value is within the *ComparisonString* then

 Use the *OrigSearchString* to DL Distance Weight Process

Else

 Use the *SoundexSearchString* to DL Distance Weight Process

End If

Experimental Simulation Result

Using one of the longest Name and longest Country Name (*found in internet*) as a source for string matching with the identified target string below, Table 2 and Table 3 shows significant improvement in computing the DL Distance Weight of DL Distance algorithm with more than calculated weight value (name =109, country=45) as well as its processing time (time=623.05 ms) over the Optimized DL Distance algorithm (name=103, country=41, time=565.92 ms).

String Name (Source):

Pablo Diego Jose Francisco de Paula Juan Nepomuceno María de los Remedios Cipriano de la Santísima Trinidad Ruiz y Picasso

String Name (Target):

Pablo Picasso

String Country Name (Source):

United Kingdom of Great Britain and Northern Ireland

String Country Name (Target):

Ireland

Implementing the above experimental simulation, the paper conducted further simulation with random sampling of minimum 10 names and country names (*assumed stop descriptors*) and compare it to at least 500 names (*assumed sanctions lists*) with the same country data. This database can be freely downloaded in the internet (*i.e.* <https://www.generatedata.com>) as sample data in this paper. The Table 4 and Table 5 shows significant improvement in

computing the DL Distance Weight of DL Distance algorithm with more than total calculated weight value ((total weight name=103,368.00, total weight country=7,500) as well as its total processing time (time=115,806.78 ms) over the Optimized DL Distance algorithm (total weight name=100,0783, total weight country=7,500, time=102,704.44 ms).

Table 2: String Match Result (String Name) using DL Distance Weight Process

Name	Weight of Comparison (Name)	Associated Country	Weight of Comparison (Country)	Processing Time (milliseconds)
Pablo Diego Jose Francisco de Paula Juan Nepomuceno María de los Remedios Cipriano de la Santísima Trinidad Ruiz y Picasso	109	United Kingdom of Great Britain and Northern Ireland	45	623.05

Table 3: String Match Result (String Country Name) using Optimized DL Distance Weight Process

Original Name	Name with Unique Sounds Phonemes	Weight of Comparison (Name)	Associated Country	Associated Country with Unique Sounds Phonemes	Weight of Comparison (Country)	Processing Time (milliseconds)
Pablo Diego Jose Francisco de Paula Juan Nepomuceno María de los Remedios Cipriano de la Santísima Trinidad Ruiz y Picasso	Pablo Digo Jos Francisco de Pala Juan Nepomusno María de los Remejos Cipriano de la Santísima Trinidad Riz y Picaso	103	United Kingdom of Great Britain and Northern Ireland	Unitd Kingdom of Gret Britain and Nortern Ireland	41	565.92

Table 4: String Match Result (Name compared to Sanctions Lists) using DL Distance Weight Process

Name	Total Weight of Comparison (Name)	Associated Country	Total Weight of Comparison (Country)	Total Processing Time (milliseconds)
Vida Humphrey	9,933.00	Philipnes	1,000.00	9,658.02
Nomrmanh P. Ratipp	12,031.00	Philippines	500.00	15,195.76
Brena Fiutxpatrik	11,045.00	Philippines	0.00	13,852.18
Tory Bec	8,835.00	Philiffines	2,000.00	7,939.65
Mary E. Shafer	10,049.00	Philippnes	500.00	10,868.59
Angiella Cetta	9,893.00	Philippines	500.00	11,917.92
Honorto M Gusman	11,237.00	Philippines	500.00	12,586.93
Lucie Trieston	9,745.00	Pilippines	500.00	10,876.42
Fatima U Brridge	10,937.00	Philippines	500.00	13,511.03
Jak A. Frank	9,663.00	Phillipine	1,500.00	9,400.28
	103,368.00		7,500.00	115,806.78

Table 5:String Match Result (Name compared to Sanctions Lists) using Optimized DL Distance Weight Process

Original Name	Name with Unique Sounds Phonemes	Total Weight of Comparison (Name)	Associated Country	Associated Country with Unique Sounds Phonemes	Total Weight of Comparison (Country)	Total Processing Time (milliseconds)
Vida Humphrey	Vida Humfrey	9,547.00	Philippines	Philippines	1,000.00	8,861.36
Nomrmanh P. Ratipp	Nomrmanh P. Ratip	11,543.00	Philippines	Philippines	500.00	13,399.23
Brena Fiutxpatrik	Brena Fiutxpatrik	11,045.00	Philippines	Philippines	0.00	13,945.94
Tory Bekc	Tory Bekc	8,835.00	Philippines	Philippines	1,500.00	7,478.74
Mary E. Shafer	Mary E. Shafer	10,049.00	Philippines	Philippines	1,000.00	10,439.04
Original Name	Name with Unique Sounds Phonemes	Total Weight of Comparison (Name)	Associated Country	Associated Country with Unique Sounds Phonemes	Total Weight of Comparison (Country)	Total Processing Time (milliseconds)
Angiella Cetta	Angila Ceta	8,875.00	Philippines	Philippines	500.00	8,812.17
Honorto M Gusman	Honorto M Gusman	11,237.00	Philippines	Philippines	500.00	12,485.33
Lucie Trieston	Luci Trison	9,217.00	Philippines	Philippines	1,000.00	8,252.11
Fatima U Brridge	Fatima U Brij	10,067.00	Philippines	Philippines	500.00	10,109.49
Jak A. Frank	Jak A. Frank	9,663.00	Philippines	Philippines	1,000.00	8,921.03
		100,078.00			7,500.00	102,704.44

4. CONCLUSION

Including the Phonetic algorithm to optimize the Damerau-Levenshtein Distance shows important improvements in weight calculation and string matching processing. This will provide significant contribution in conducting a suitable amount of evaluation to determine whether the likelihood of match is a “true match” or a “false positive” during sanction screening. With the said sanctions screening improvement, it will eliminate any obstacle for batch and real-time sanctions interdiction and will increase due diligence in screening. For future research, other associated phonetic algorithms such as Metaphone and Double Metaphone may assist to further enhance the efficiency of the sanctions screening process by encoding input names with varying lengths by their English pronunciation.

REFERENCES

[1] "IPES," 2017. [Online]. Available: https://www.ipes.com/sites/default/.../Sanctions%20Screening%20Leaflet_March%2015.pdf

[2] Apoorva Deshpande, and Ramnaresh Sharma, “Multilevel Ensemble Classifier using Normalized Feature based Intrusion Detection System”, International Journal of Advanced Trends in Computer Science and Engineering, Volume 5, Number 5, pp. 72-76, 2018. <https://doi.org/10.30534/ijatcse/2018/02752018>

[3] D. Hema Latha, and P. Premchand, Estimating Software Reliability using Ant Colony Optimization Technique with Salesman Problem for Software Process, International Journal of Advanced Trends in Computer Science and Engineering, Volume 7, Number 5, pp. 20-29, 2018. <https://doi.org/10.30534/ijatcse/2018/04722018>

[4] "PWC," 2017. [Online]. Available: <https://www.pwc.com/us/en/financial-services/publications/assets/global-sanctions-practice-insurance.pdf>

[5] Ryan Gabrys, Eitan Yaakobi and Olgica Milenkovic, “Codes in the Damerau Distance for DNA Storage”, 2016 IEEE International Symposium on Information Theory <https://doi.org/10.1109/ISIT.2016.7541778>

[6] "Trulioo," 30 August 2016. [Online]. Available: <https://www.trulioo.com/blog/aml-watchlist-screening>.

[7] Quyet Thang Dang, Trung Huy Phan, “Determining Restricted Damerau-Levenshtein Edit-Distance of Two Languages by Extended Automata”, 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)”, 2010 <https://doi.org/10.1109/RIVF.2010.5632914>

[8] Jiamei Liu, Suping Wu, “Research on longest common subsequence fast algorithm”, 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), 2011

[9] Haifeng Hu, Liang Zhang; Jianshen Wu, “Hamming distance based approximate similarity text search algorithm”, 2015 Seventh International Conference on

Advanced Computational Intelligence (ICACI), 2015 <https://doi.org/10.1109/ICACI.2015.7184772>

[10] H. A. Maarif, R. Akmeliawati, Z. Z. Htike, Teddy S. Gunawan, “Complexity Algorithm Analysis for Edit Distance”, 2014 International Conference on Computer and Communication Engineering, 2014 <https://doi.org/10.1109/ICCCE.2014.48>

[11] Khaled Balhaf; Mohammad A. Alsmirat; Mahmoud Al-Ayyoub; Yaser Jararweh; Mohammed A. Shehab, “Accelerating Levenshtein and Damerau edit distance algorithms using GPU with unified memory”, 2017 8th International Conference on Information and Communication Systems (ICICS), 2017 <https://doi.org/10.1109/IACS.2017.7921937>

[12] Joseph Jupin; Justin Y. Shi; Zoran Obradovic, “Understanding Cloud Data Using Approximate String Matching and Edit Distance”, 2012 SC Companion: High Performance Computing, Networking Storage and Analysis 2012 <https://doi.org/10.1109/SC.Companion.2012.149>

[13] Dr. Andrew T. Freeman, Dr. Sherri L. Condon and Christopher M. Ackerman, “Cross Linguistic Name Matching in English and Arabic: A “One to Many Mapping” Extension of the Levenshtein Edit Distance Algorithm”, The Mitre Corporation 7525 Colshire Dr. McLean, Va 22102-7505, Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 471–478, Association for Computational Linguistics, New York, June 2006. <https://doi.org/10.3115/1220835.1220895>

[14] Tohari Ahmad; Kukuh Indrayana; Waskitho Wibisono; Royyana M. Ijtihadie, “Edit distance weighting modification using phonetic and typographic letter grouping over homomorphic encrypted data”, 2017 3rd International Conference on Science in Information Technology (ICSITech), 2017 <https://doi.org/10.1109/ICSITech.2017.8257147>

[15] Abir Bin Ayub Khan; Mohammad Sheikh Ghazanfar; Shahidul Islam Khan, “Application of phonetic encoding for analyzing similarity of patient's data: Bangladesh perspective”, 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 2017

[16] Rogie Nino, Ariel Sison, Ruji Medina, “Development of Fuzzy Risk Score Assessment Framework for Sanctions Screening”, International Journal of Engineering Research and Technology, Volume 12, Number 4, pp. 554-562, 2019.