International Journal on Advanced Science Engineering Information Technology

Vol.9 (2019) No. 4
ISSN: 2088-5334

Comparison between the Stemmer Porter Effect and Nazief-Adriani on the Performance of Winnowing Algorithms for Measuring Plagiarism

Alam Rahmatulloh[#1], Neng Ika Kurniati[#2], Irfan Darmawan[*], Adi Zaenal Asyikin[#3], Deden Witarsyah J[*]

[#] *Department of Informatics, Siliwangi University, Tasikmalaya, Indonesia*
E-mail: [1]alam@unsil.ac.id, [2]nengikakurniati@unsil.ac.id, [3]adi.zaenala@gmail.com

[*]*Department of Information System, Telkom University, Bandung, Indonesia*
E-mail: irfandarmawan@telkomuniversity.ac.id, dedenw@telkomuniversity.ac.id

***Abstract*—Current technological developments change physical paper patterns into digital, and this has a very high impact. Positive impact because paper waste is reduced, on the other hand, the rampant copying of digital data raises the amount of plagiarism that is increasing. At present, there are many efforts made by experts to overcome the problem of plagiarism, one of which is by utilizing the winnowing algorithm as a tool to detect plagiarism data. In its development, many optimizing winnowing algorithms used stemming techniques. The most widely used stemmer algorithms include stemmer porter and nazief-adriani. However, there has not been a discussion on the comparison of the effect of performance using stemmer on the winnowing algorithm in measuring the value of plagiarism. So it is necessary to research the effect of stemmer algorithms on winnowing algorithms so that the results of plagiarism detection are more optimal. The results of this study indicate that the effect of nazief-adriani stemmer on the winnowing algorithm is superior to the stemmer porter, only decreasing the detection performance of the 0.28% similarity value while the Porter stemmer is superior in increasing the processing time to 69% faster.**

***Keywords*—Nazief-Adriani; plagiarism; porter; stemmer; winnowing.**

I. INTRODUCTION

The development of information technology is now more advanced making documents that were previously physical have now been made in digital form so that digital copying can easily be carried out, which can lead to plagiarism. The vulnerability of plagiarism in digital documents encourages researchers to develop plagiarism checker software in detecting plagiarism, by measuring the level of similarity of the document with other documents using various techniques or algorithms [1]–[3]. In detecting digital document plagiarism, several methods can be used to measure the level of similarity of a document, namely the full-text comparison method, keyword similarity method, and fingerprinting method [4].

Fingerprinting is a method that traces characters one by one in a character sequence. The working principle of this fingerprinting method is to use the hashing technique. The advantage of the fingerprinting method is that the processing time is faster than the full-text comparison method and the keyword similarity method. Some algorithms included in the fingerprinting method are Rabin Karp Algorithm, Winnowing Algorithm, and Manber Algorithm. The winnowing algorithm is most widely used based on several studies, with a better, more efficient, and reliable level of accuracy for plagiarism detection [4]–[8]. In optimizing the detection of plagiarism, most of the text processing adds pre-process, one of which is called stemming by changing the word to root word [9], [10].

Previous research that applied stemming Porters to the Winnowing algorithm was carried out by [11], his research showed that the Porter stemmer algorithm helped speed up the winnowing algorithm in determining the value of fingerprints of a text. Furthermore, the effect of Nazief-adriani's stemming algorithm on the performance of the Winnowing algorithm to detect Indonesian language plagiarism by [12], his research resulted in a stemming process in the Winnowing algorithm which tends to reduce the similarity level but speeds up processing time by approximately 30%.

As for the comparison of the effect of porter and Nazief-Adriani stemmer on the performance of the Winnowing algorithm, it has never been done. In this study, a comparison of the effect of the Porter stemmer and Nazief-Adriani stemmer on the performance of the Winnowing algorithm in measuring plagiarism is based on similarity and speed of the process.

II. MATERIALS AND METHOD

A. Materials

1) Plagiarism: The word plagiarism comes from the Latin *plagiare* verb which means to kidnap. Ben Johnson used the term in 1601. Plagiarism is an act of misuse, theft or seizure, publishing, statement or declaring itself as a thought, idea, writing, or creation that belongs to someone else [12], [13].

2) Preprocessing: Text that will be carried out by the text mining process generally has several characteristics including having high dimensions, there is noise in the data, and there is a text structure that is not good. The method used in studying text data is first to determine the features that represent each word for each feature in the document. Before determining the features that represent, the preprocessing stage is needed which is generally done in text mining on documents, namely folding cases, tokenizing, filtering, stemming. Preprocessing stages can be seen in Fig. 1 [9].

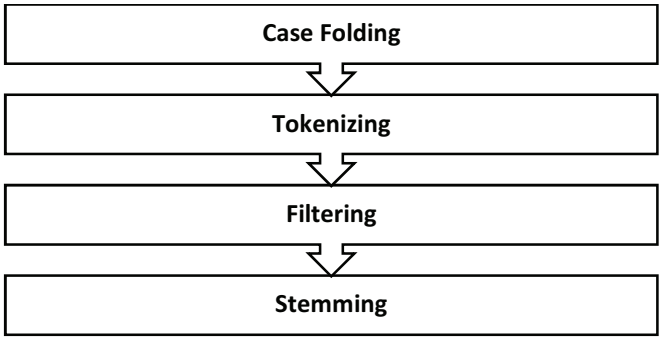

Fig. 1 Preprocessing stages

3) Stemmer: Stemmer is a necessary word search process by cutting affixes (prefixes, suffixes, inserts, combinations) that are run with specific algorithms [14]. The stemmer algorithm that was first developed was by Martin Porter, who worked on cutting deductions in English.

4) Porter Stemmer Algorithm: Porter Stemmer for Indonesian based on English Porter Stemmer (S_P) developed by W.B. Frakes in 1992 [15]. Because English comes from different classes, several modifications have been made to make the Porter Algorithm usable by Indonesian [16]–[18]. The Porter Stemmer algorithm for Indonesian has the following steps:

- Step 1. Deleting particles like: -kah, -lah, -tah

TABLE I
EXAMPLE OF STEP 1

Suffix	Replacement	Additional Condition	Example
-kah	Null	Null	Siapakah
-lah	Null	Null	Hadapilah
-pun	Null	Null	Adapun

- Step 2. Removing the pronouns (Possessive Pronoun), like -ku, -mu, -nya

TABLE II
EXAMPLE OF STEP 2

Suffix	Replacement	Additional Condition	Example
-ku	Null	Null	Rumahku
-mu	Null	Null	Suamimu
-nya	Null	Null	Istrinya

- Step 3. Erasing the first prefix. If not found, then go to step 4a, and if there is, go to step 4b.

TABLE III
EXAMPLE OF STEP 3

Prefix	Replacement	Additional Condition	Example
ber-	Null	Null	Bertelur→telur
bel-	Null	Ajar	Belajar→ajar
Pel-	Null	Ajar	Pelajar→ajar

- Step 4:

a. Delete the second prefix, and continue in step 5a
b. Deleting suffix, if it is not found, the word is assumed to be a root word. If found, then go to step 5b.

TABLE IV
EXAMPLE OF STEP 4

Prefix	Replacement	Additional Condition	Example
meny-	S	V … *	Menyapu→sapu
mem-	P	V …	Memaksa→paksa
peny-	S	V …	Penyapu→sapu

* This notation means that the stem starts with a vowel.

- Step 5:

a. Deleting endings and end words are assumed as root words.
b. Removing the second prefix and the final word are assumed to be root words.

TABLE V
EXAMPLE OF STEP 5

Prefix	Replacment	Additional Condition	Example
-kan	Null	prefix ∉ {ke, peng}	*tarikkan → tarik (meng)ambilkan → ambil*
-an	Null	prefix ∉ {di, meng, ter}	*makanan → makan (per)janjian → janji*
-i	null	V|K...c1c1, c1 ≠ s,c2 ≠ i and prefix ∉ {ber, ke, peng}	*tandai → tanda (men)dapati → dapat*

5) Nazief-Adriani Stemming Algorithm: The Nazief-Adriani (S_NA) stemming algorithm (1996) was developed based on Indonesian morphological rules which classify affixes into prefixes (prefixes), inserts (suffixes), suffixes (suffixes) and combined prefixes (confixes) [17], [19].

Algorithms made by Bobby Nazief and Mirna Adriani have the following stages:

1. Look for words that will be stemming in the dictionary. If it is found, it is assumed that the word is the root word. Then the algorithm stops.
2. Inflection Suffixes ("-lah", "-kah", "-ku", "-mu", or "-nya") are discarded. If it is in the form of particles ("-lah", "-kah", "-tah" or "-pun") then this step is repeated again to delete obsessive pronouns ("-ku", "-mu", or "-nya"), If there is.
3. Delete Derivation Suffixes ("-i", "-an" or "-kan"). If the word is found in the dictionary, the algorithm stops. If not then go to step 3a
   a. If "-an" has been deleted and the last letter of the word is "-k", then "-k" is also deleted. If the word is found in the dictionary, the algorithm stops. If not found then do step 3b.
   b. Deleted suffixes ("-i", "-an" or "-kan") are returned, go to step 4.
4. Remove Derivation Prefix (be-, di, me-, pe-, se-, te-). If in step 3 there is a deleted suffix then go to step 4a, if not go to step 4b.
   a. Check the combination table prefix suffix that is not permitted. If it is found, the algorithm stops, if it does not go to step 4b.
   b. For i = 1 to 3, specify the type of prefix then delete the prefix. If the root word has not been found, do step 5, if the algorithm has stopped. Note: if the second prefix equals the first prefix of the stop algorithm.
5. Recoding.
6. If all steps have been completed but are not successful, then the first word is assumed to be the root word. Process complete.

6) Winnowing Algorithm: The Winnowing algorithm [6] is an algorithm to produce a unique number (fingerprint) series that represents a document. With the fingerprint, we can know the level of similarity between one document and another document [7].

The Winnowing algorithm works as follows:

1. Removal of irrelevant characters (whitespace insensitivity).
2. Formation of gram series with size k.
3. Calculation of hash values (1).

$$\boldsymbol{c1} * \boldsymbol{b}^{(k-1)} + \boldsymbol{c2} * \boldsymbol{b}^{(k-2)} + \cdots + \boldsymbol{c}(\boldsymbol{k}-\boldsymbol{1}) * \boldsymbol{b} + \boldsymbol{c_k} \quad (1)$$

   Information:
   c: character ASCII value
   b: base (prime number)
   k: lots of characters
4. Divide into certain windows.
5. Selecting multiple hash values into fingerprint documents
6. Similarity calculation using The Jaccard Similarity Coefficient

7) Jaccard Similarity Coefficient: The Jaccard's Similarity Coefficient (Jaccard 1912) is a standard index for binary variables. This is defined as the quotient between intersections and variable unions compared to pairs between two objects. To calculate the similarity of two documents, a Jaccard's Similarity Coefficient is required, with equation (2).

$$D(A,B) = \frac{|A \cap B|}{|A \cup B| - |A \cap|} x\ 100\% \quad (2)$$

Information:

$D(A,B)$ is a similarity value,
$|A \cap B|$ the number of documents one and two of the same fingerprints,
$|A \cup B|$ Number of document one and two fingerprints.

B. Related Work

The test results in the study [20] showed that the performance of the winnowing algorithm (91.8%) was not as good as the fingerprint algorithm (92.8%), but the winnowing algorithm had a better level of topic relevance. Also, the most widely used winnowing algorithm is based on several studies, with better, more efficient and reliable accuracy for plagiarism detection [4], [6]–[8].

The previous research [21], [11] which carried out the application of the stemmer algorithm S_P and S_NA on the winnowing algorithm resulted in the performance of the winnowing algorithm faster in plagiarism detection, but it is not yet known that the stemmer algorithm has a better influence on the winnowing algorithm in measuring the similarity value and the speed of the process.

The focus of this study compares the effect of the S_P and S_NA stemmer algorithm on the performance of the winnowing algorithm seen from the parameters of the similarity results and the processing time.

C. Research Method

The method used is a similar study methodology, where each algorithm is tested in terms of algorithm performance. Then the results of the tests are compared based on the similarity and processing time. The steps used in Fig. 2.

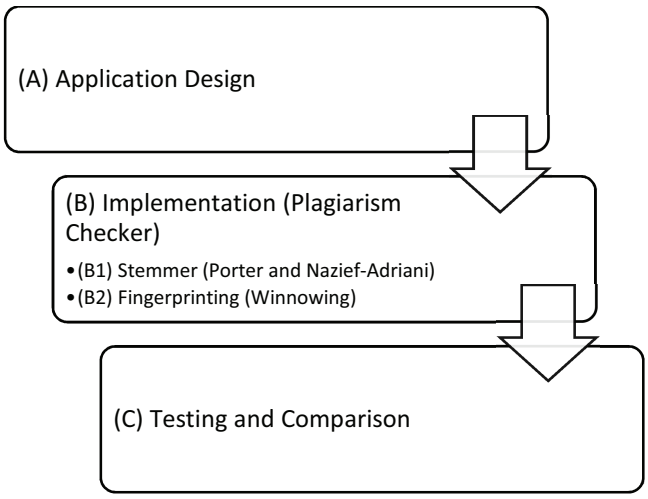

Fig. 2 Proposed Method

1) Application Design: Design a simple application to test the effect of the S_P and S_NA stemmer algorithms. The application is made web-based using PHP with the Apache web server.

2) Implementation: At this stage is the implementation stage, which consists of the implementation stage of the system interface, implementation of the process and implementation of the algorithm. The calculation stages in the application are adjusted to the stages in the porter, and nazief-adriani stemmer algorithm after the stemmer process is carried out then proceed with the winnowing algorithm according to the stages.

3) Testing and Comparison: At this stage, the test is based on similarity parameters and process speed. The results of the tests are then compiled and taken the average value, and then the data is compared between the algorithms used.

III. Result and Discussion

A. Testing Documents

Table 6 is the document information that will be used for testing. Documents tested consist of 8 documents with test data that can be obtained from http://bit.ly/Win-SP-SNA.

TABLE VI
Test Document Information

Document Name	Number of words	Description
Original Document	4720 word	Report chapter 1-3 Final Project Adi Zaenal Asyikin
Document 1	4720 word	Documents whose contents are the same as Original Documents
Document 2	3776 word	Documents that contain the text are randomly cut 20% of the words to produce 80% of the same words as the original documents.
Document 3	2832 word	Documents that contain the text are cut by 40% of the words randomly, resulting in 60% of the same words as the original documents.
Document 4	1888 word	Documents that contain the text are 60% randomly deducted so that they produce 40% of the same words as the original documents.
Document 5	944 word	Documents that contain the text are cut 80% of the words randomly to produce 20% of the same words as the original documents.
Document 6	4720 word	Documents whose contents are 100% the same as the original documents but some sentences are exchanged.
Document 7	4720 word	Documents that contain 2% of the word are spinning (such as replacing the word: software into piranti lunak) so that 98% is said to be the same as the original document.

B. Testing Scheme

Similarity tests are performed using the optimal k-gram 6 and w-gram 4 values based on the results of previous tests, with a testing scheme such as Table 7.

TABLE VII
Testing Scheme

Test	Practice Document	Test Document
1	Original Document	Document 1
2	Original Document	Document 2
3	Original Document	Document 3
4	Original Document	Document 4
5	Original Document	Document 5
6	Original Document	Document 6
7	Original Document	Document 7

C. Test Results

1) Similarity Test Results: The results of the similarity test for pure Winnowing algorithms (without stemming), the Winnowing-Stemmer Porter algorithm, and the Winnowing-Stemmer Nazief-Adriani algorithm. With the results of the pure Winnowing algorithm having an average similarity of 70.7%, the Winnowing algorithm - Porter Stemmer has an average similarity of 65.7%, and the Winnowing algorithm - Stemmer Nazief-Adriani has an average similarity of 70.5%. Data from the results of similarity tests are presented in Table 8, and the graph can be seen in Fig. 3.

TABLE VIII
Similarity Test Results

Test	Winnowing	Winnowing - Porter	Winnowing - Nazief
1	100.0%	100.0%	100.0%
2	77.8%	63.4%	77.7%
3	56.5%	42.5%	56.4%
4	43.2%	47.1%	42.8%
5	21.6%	21.9%	21.0%
6	98.0%	92.9%	98.1%
7	97.6%	92.2%	97.8%
Avg	**70.7%**	**65.7%**	**70.5%**

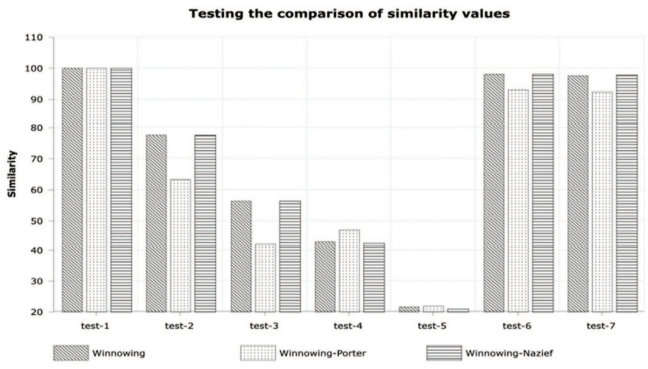

Fig. 3 Testing the comparison of similarity values

2) Process Speed Test Results: The results of the processing speed testing of the pure Winnowing algorithm (without stemming), the Winnowing-Stemmer Porter algorithm, and the Winnowing-Stemmer Nazief-Adriani algorithm. With the results of the Winnowing algorithm purely produce an average processing time of 0.711 seconds, the Winnowing algorithm - the Stemmer Porter produces an average processing time of 0.221 seconds, and the Winnowing algorithm-Stemmer Nazief-Adriani has an average processing time of 0.476 seconds. Data from the Process Speed test results are presented in Table 9, and the graph can be seen in Fig. 4.

TABLE IX
PROCESS SPEED TEST RESULTS

Test	Winnowing (second)	Winnowing – Porter (second)	Winnowing – Nazief (second)
1	0.825	0.289	0.566
2	0.779	0.222	0.504
3	0.625	0.203	0.429
4	0.663	0.181	0.402
5	0.473	0.147	0.334
6	0.817	0.253	0.552
7	0.794	0.252	0.547
Avg	0.711	0.221	0.476

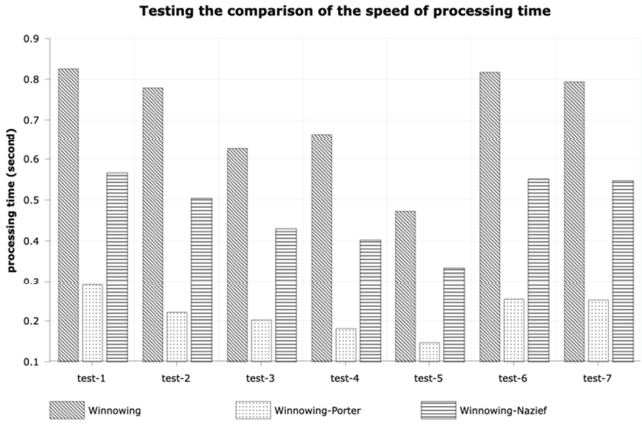

Fig. 4 Testing the comparison of the speed of processing time.

TABLE X
COMPARISON PERFORMANCE OF STEMMER ON WINNOWING ALGORITHM

Algorithm / Parameter	Winnowing	Winnowing-Porter	Nazief-Winnowing
Similarity	70.7%	65.7%	70.5%
Process Speed	0.711 s	0.221 s	0.476 s

Based on table 10, it is known that the use of Stemmer can affect the similarity and processing time of the Winnowing algorithm. The use of the Stemmer Porter Algorithm (S_P) in the Winnowing Algorithm is superior at the time of the process speed but decreases the performance of the detection rate of plagiarism. While the use of the Stemmer Nazief-Adriani (S_NA) Algorithm in the Winnowing Algorithm results in less significant plagiarism detection performance, and faster process performance than the stemless Winnowing Algorithm.

IV. CONCLUSION

Based on the results of testing and discussion it can be concluded several things as follows: The Stemmer process can affect the similarity and speed of the process of the Winnowing Algorithm. The similarity value decreased by 7% in the use of the Stemmer Porter algorithm in the Winnowing algorithm, while the processing speed increased by 69% compared to the Winnowing Algorithm without Stemmer (pure). While the use of the Nazief-Adriani Stemmer Algorithm produces a similarity value decreasing by only 0.28%, but the process speed increases by 33%.

The use of stemmer, on the one hand, is useful to speed up the process of the winnowing algorithm, but it influences the performance of the similarity value to detect plagiarism. There needs to be more in-depth discussion and experimentation with other algorithms or techniques so that the performance of the plagiarism detection engine becomes more optimal.

REFERENCES

[1] H. Lamba and S. Govilkar, "A Survey on Plagiarism Detection Techniques for Indian Regional Languages," Int. J. Comput. Appl., 2017.

[2] A. M. El Tahir Ali, H. M. D. Abdulla, and V. Snasel, "Survey of plagiarism detection methods," in Proceedings - AMS 2011: Asia Modelling Symposium 2011 - 5th Asia International Conference on Mathematical Modelling and Computer Simulation, 2011.

[3] D. Namdev, "A Survey Paper on Plagiarism Detection Techniques," Int. Conf. ICT Healthc., pp. 30–34, 2015.

[4] L. Lulu, B. Belkhouche, and S. Harous, "Overview of fingerprinting methods for local text reuse detection," in Proceedings of the 2016 12th International Conference on Innovations in Information Technology, IIT 2016, 2017.

[5] E. G. Hasan, A. Wicaksana, and S. Hansun, "The Implementation of Winnowing Algorithm for Plagiarism Detection in Moodle-based E-learning," Proc. - 17th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2018, pp. 321–325, 2018.

[6] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local Algorithms for Document Fingerprinting," in ACM International Conference on Management of Data (SIGMOD), 2003.

[7] N. Elbegbayan, "Winnowing, a Document Fingerprinting Algorithm," Science (80-.). 2005.

[8] N. Alamsyah, "Perbandingan Algoritma Winnowing Dengan Algoritma Rabin Karp Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi," Technologia, vol. 8, no. 3, pp. 124–134, 2017.

[9] T. Mardiana, T. Bharata Adji, and I. Hidayah, "Stemming Influence on Similarity Detection of Abstract Written in Indonesia," Telkomnika (Telecommunication Comput. Electron. Control. 2016.

[10] Z. Ceska and C. Fox, "The Influence of Text Pre-processing on Plagiarism Detection," Int. Conf. RANLP 2009, pp. 55–59, 2009.

[11] H. T. Nugroho, "Pengaruh Algoritma Stemming Nazief-Adriani Terhadap Kinerja Algoritma Winnowing Untuk Mendeteksi Plagiarisme Bahasa Indonesia," J. Ultim. Comput. vol. 9, no. 1, pp. 36–40, 2017.

[12] J. Vassallo, "WASP (Write a Scientific Paper): Plagiarism and the ethics of dealing with colleagues," Early Hum. Dev., vol. 124, pp. 65–67, 2018.

[13] Kock and Davison, "Dealing with Plagiarism in the Information Systems Research Community: A Look at Factors That Drive Plagiarism and Ways to Address Them," MIS Q., 2017.

[14] D. Sharma, "Stemming Algorithms: A Comparative Study and their Analysis," Int. J. Appl. Inf. Syst., 2013.

[15] P. Willett, "The Porter stemming algorithm: Then and now," Program, 2006.

[16] R. Sugumar and M. R. Priya, "Improved Performance of Stemming Using Enhanced Porter," Int. J. Eng. Sci. Res. Technol., vol. 7, no. 4, pp. 681–686, 2018.

[17] J. Asian, H. E. Williams, and S. M. M. Tahaghoghi, "Stemming Indonesian: a confix-stripping approach," Conf. Res. Pract. Inf. Technol. Ser., vol. 38, pp. 307–314, 2005.

[18] V. Gurusamy and S. K. K. Nandhini, "Performance Analysis : Stemming Algorithm for the English Language," IJSRD - Int. J. Sci. Res. Dev., vol. 5, no. 05, pp. 1933–1938, 2017.

[19] J. Asian, "Effective Techniques for Indonesian Text Retrieval," 2007.

[20] A. T. Wibowo, K. W. Sudarmadi, and A. M. Barmawi, "Comparison between fingerprint and winnowing algorithm to detect plagiarism fraud on Bahasa Indonesia documents," in 2013 International Conference of Information and Communication Technology, ICoICT 2013, 2013.

[21] R. Sutoyo, I. Ramadhani, and A. D. Ardiatma, "Detecting Documents Plagiarism using Winnowing Algorithm and K-Gram Method," *Cybern. Comput. Intell. (CyberneticsCom), 2017 IEEE Int. Conf.*, pp. 67–72, 2017.