IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.15, No.2, April 2021, pp. 199~208
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: 10.22146/ijccs.64526

Author Obfuscation on Indonesian News Articles Using Genetic Algorithm

Rayhan Naufal Ramadhan*[1], Yunita Sari[2], Aina Musdholifah[3]
[1]Undergraduate Program of Computer Science; FMIPA UGM, Yogyakarta, Indonesia
[2,3]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]rayhannaufal2017@mail.ugm.ac.id**, [2]yunita.sari@ugm.ac.id, [3]aina_m@ugm.ac.id

Abstrak

Authorship attribution adalah metode untuk mengidentifikasi penulis suatu teks dari sekelompok penulis potensial yang dapat digunakan untuk memecahkan anonimitas penulis yang tidak diketahui. Hal tersebut mengancam kebebasan berpendapat dan privasi seseorang, terutama orang yang ingin menulis secara anonim. Untuk melawan ancaman tersebut, metode author obfuscation diusulkan untuk memodifikasi suatu teks supaya penulisnya sulit diidentifikasi tanpa mengaburkan topik utamanya.

Pada penelitian ini, dibuat model author obfuscation berbasis algoritma genetika untuk memodifikasi artikel berita berbahasa Indonesia supaya tidak teridentifikasi oleh model authorship attribution dengan tetap menjaga semantik artikel yang dimodifikasi sama dengan aslinya. Model tersebut secara iteratif mengubah beberapa kata dalam artikel menggunakan teknik crossover dan mutasi yang dipandu fungsi fitness yang melibatkan probabilitas identifikasi dan kemiripan dengan artikel asli.

Model tersebut dievaluasi berdasarkan parameter safety, soundness, dan sensibleness. Model tersebut memiliki safety yang baik karena dapat menurunkan akurasi model authorship attribution yang diberikan sebesar 0,3018, tetapi turun menjadi 0,1179 ketika diuji pada model yang berbeda dari yang dilibatkan pada fungsi fitness. Soundness model tersebut cukup baik karena kemiripan artikel yang dimodifikasi dengan aslinya mencapai 0,7817. Sensibleness dievaluasi secara manual dan diperoleh skor 2,571 dari skala 0 sampai 4 yang menunjukkan bahwa tata bahasa sebagian artikel dapat diterima, tetapi tak sedikit juga yang berantakan.

Kata kunci*— author obfuscation, authorship attribution, algoritma genetika*

Abstract

Authorship attribution is a method for identifying the author of a text from a group of potential authors and can solve the anonymity of unknown authors. Such method threatens anyone's privacy, especially those who wish to write anonymously. To address this issue, author obfuscation is proposed to modify a text to disguise its author.

In this research, a genetic algorithm-based author obfuscation model was created to modify Indonesian news articles to avoid identification from authorship attribution while keeping its semantics. The model iteratively changed some words in the article using crossover and mutation techniques guided by a fitness function which involve identification probability and similarity to the original article.

The model is evaluated based on safety, soundness, and sensibleness parameter. The model has good safety since it can reduce the given authorship attribution model's accuracy by 0.3018 but drops to 0.1179 when tested on different models. Its soundness is pretty good since the similarity of the modified to the original articles reaches 0.7817. The model obtained a score of 2.571 on a scale of 0 to 4 in terms of sensibleness which indicates that some articles are acceptable in terms of grammar, but not a few are messy.

Keywords*— author obfuscation, authorship attribution, genetic algorithm*

Received March 6th,2021; Revised April 29th, 2021; Accepted April 29th, 2021

1. INTRODUCTION

Researches related to automatic text categorization began to be of great interest since the end of the 20th century [1]. One of its branches is stylometric-based categorization for authorship attribution purposes. Authorship attribution is a method for identifying the author of a text from a group of potential authors, assuming each writer has a unique writing style [2]. Stylometry is an analysis of writing style features that can be measured statistically, such as sentence length, vocabulary diversity, word frequency, and so on [3].

Authorship attribution plays an important role in many domains. One example is in the world of forensics, authorship attribution can determine the linguistic profile of a suspicious text writer [4]. On the other hand, [5] stated that authorship attribution poses a threat to freedom of speech and privacy, especially for activists who wish to publish their articles anonymously. To fight this threat, a method called author obfuscation is proposed to obfuscate text, namely modifying the text so that the author's identity is disguised and can't be identified correctly by the authorship attribution model [6].

According to [7], author obfuscation performance is measured based on three parameters, namely safety (the ability to disguise the original author from a given text), soundness (the level of similarity between the text modified from the obfuscation process with the original text), and sensibleness (the quality of grammar and legibility of the resulting text).

Author obfuscation can be done manually by the author of an article itself. Even so, the manual process can be a challenge for the writer in determining changes in their article, so it can also be done automatically. [8] obfuscated texts by translating them into another language then translating them back into their original language using a machine translation API such as Google Translate. Besides, rule-based automated author obfuscation such as changing synonyms [9], certain parts of speech such as nouns and verbs [6], and manipulating punctuation frequency [10] are already conducted.

In this research, author obfuscation is implemented using genetic algorithm on Indonesian news articles. Genetic algorithm can make changes to the articles using crossover and mutation techniques while being guided by a fitness function that involves attribution probability and semantics relevance. Attribution probability can be interpreted as the level of confidence of the authorship attribution model in identifying the author of the text. Using these techniques, the algorithm can find the right set of modifications to the article to successfully disguise the author of the text while maintaining its semantics.

2. METHODS

In this section, the proposed method is explained in detail. This includes the data used in this research, the authorship attribution, and the author obfuscation model.

2.1 Data Collection

The data used in this research is Indonesian news articles that mainly discuss the Indonesian government issues such as politics, economy, and handling of COVID-19, taken from Kompas.com and Kumparan.com. Three news writers from each site were selected and their 1500 latest articles were collected, meaning there are 9000 news articles in the data. The data contains the news articles as the feature and the authors as the label. The oldest article was published on September 23rd 2019, while the latest one was on October 26th 2020.

2.2 Authorship Attribution Model

Figure 1 shows the flowchart of the authorship attribution process. There are three activities that must be done in the process, i.e. preprocessing, feature extraction, and authorship attribution that includes the training and testing process of the model.

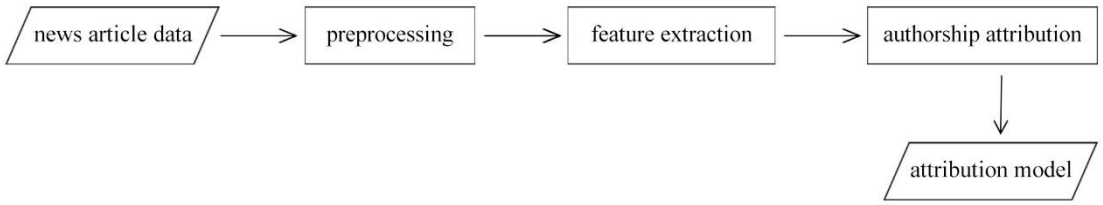

Figure 1 Flowchart of the authorship attribution process

Further explanation of the authorship attribution is as follows.

1. Preprocessing
   Preprocessing is done to clean the data from noises. It consists of lowercasing, removing non-alphabetic characters, and tokenization or splitting the article into sequence of tokens.
2. Feature extraction
   Two thousand tokens with the most frequent occurrence from the tokenization process are used as features. There are 6 types of features representing different authorship attribution models, which means there are 6 models used in this research. Those types are as follows:
   a. *Word unigram*. Each token consist of one word. For sentence "beliau dilarikan ke rumah sakit", the feature will be ["beliau", "dilarikan", "ke", "rumah", "sakit"].
   b. *Word bigram*. Example, ["beliau dilarikan", "dilarikan ke", "ke rumah", "rumah sakit"].
   c. *Character 4-gram*. Each article is splitted into tokens representing sequence of 4 characters; for example, ["beli", "elia", "liau", "iau ", …, " sak", "saki", "akit"].
   d. *Character 5-gram*. Example: ["belia", "eliau", "liau ", …, "h sak", " saki", "sakit"].
   e. *Character 6-gram*. Example: ["beliau", "eliau ", "liau d", …, "h saki", " sakit"].
   f. *Combination of best word and character n-gram*. After all authorship attribution models are evaluated, there will be another model using the combination of feature (a or b) and (c, d, or e) whose model has the best accuracy.

   These features are then represented in the form of TF-IDF (Term Frequency – Inverse Document Frequency). It is used for term weighting. Term frequncy is the number of occurrences of a certain term or token in a document or article, while document frequency is the number of documents containing that term [11]. Equation (1) is utilized to calculate the inverse document frequency (IDF). In the equation, idf_i is the inverse document of term i, D is the total of articles, and df_i is the document frequency of term i.

$$idf_i = \log \frac{D}{df_i} \tag{1}$$

   TF-IDF of a term or token in a single article or document is obtained using Equation (2) by multiplying the term frequency in that article and its inverse document frequency. In the equation, $tfidf_{i,j}$ is the weight of term i in article j, $tf_{i,j}$ is the number of occurences of term i in article j, and idf_i is the same as in Equation (1).

$$tfidf_{i,j} = tf_{i,j} \times (idf_i + 1) \tag{2}$$

3. Authorship attribution
   After the features are extracted, the data is divided into training and test data. The features and the labels of all articles in the training data are utilized to train the authorship attribution model to identify the author of each article. After being trained, the model is tested by asking it to identify the author of the articles in the test data. Testing is done to calculate the accuracy and F1-score of the model. The training and

testing process uses LinearSVC class from *scikit-learn* library that implements Support Vector Machine (SVM) with a linear kernel.

2.3 Author Obfuscation Model

Author obfuscation is the most important part of this research. The author obfuscation model tries to fool the authorship attribution models mentioned above by obfuscating the articles using a genetic algorithm approach until the attribution models misidentify the authors. The articles involved in the obfuscation process are only from the test data. Since there are six attribution models, the obfuscation process is done against each one of the models. The concept of the process is simple; if an article is already misclassified by the attribution model, it will be skipped. Otherwise, the obfuscation model will try to obfuscate it until a stopping criteria that will be explained later is met. Figure 2 shows the obfuscation flow on a single article.

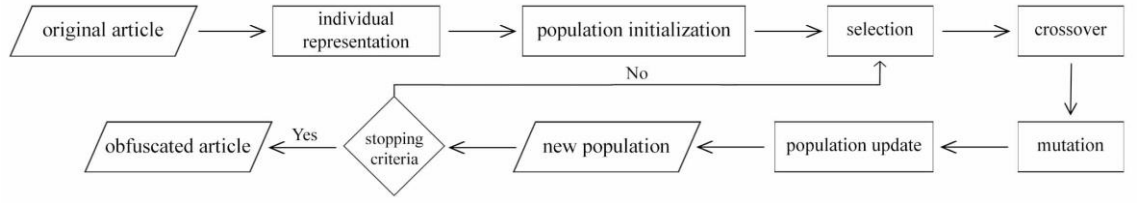

Figure 2 Flowchart of the author obfuscation model

The detailed process of the author obfuscation model is as follows.

1. Individual representation
   In the author obfuscation model, the input article is called as an individual. An individual is represented as a sequence of word unigram as mentioned in the authorship attribution model, but without lowercasing and non-alphabetic characters removal to keep its writing format when it is translated back into its article form.
2. Population initialization
   The population size is set to 16 and initially only contains a single individual representing the original article. To fill it up, that individual is mutated 15 times to create new individuals. The mutation process will be explained later. Then, the individuals in the popualtion are descendingly sorted based on their fitness.
3. Selection
   In this phase, individuals in the population are selected to enter a mating pool with the size of 12 using a selection method called tournament selection. In tournament selection, two individuals are randomly picked and the fittest one is selected to enter the mating pool [12]. This process is repeated until the mating pool is full.
4. Fitness function
   Fitness of an individual is calculated using fitness function. For an individual x representing an article written by author a, the components of the fitness function $F(x)$ are as follows:
   a. $P(x, a)$. Attribution probability or the probability the given attribution model can identify a as the author of x.
   b. $S(x)$. The similarity between the article represented by x and the original article. It is measured by collecting all modified words in x, calculating their similarity to their respective original words in the original article, and finally calculating the average. The similarity between the modified and the original is calculated using a word embedding library called FastText [13]. It can transform words into vectors and measure the similarity between two words by calculating the cosine similarity of their vector representations.
   c. α. A parameter whose value set to 0.7 that denotes that attribution probability is more significant than the similarity to calculate the fitness.

$$F(x) = \frac{(1-\alpha)S(x)}{\alpha P(x,a)} \quad (3)$$

Equation (3) defines the fitness function. Based on the equation, an individual is fit if it has a low attribution probability and high similarity to the original article. The fitness function helps the obfuscation model to preserve individuals with these criteria.

5. Crossover
   The goal of crossover is to produce a new individual by combining portions of two parent [6]. This new individual is called offspring. Eight pairs of individuals in the mating pool are randomly selected as parent individuals for crossover using single-point mechanism. Specifically, the parent individuals are divided into three parts and a random point is selected from the middle part to divide both of them in two halves. Each pair form two offspring by combining (1) the first half of the first parent and the second half of second parent and (2) the first half of second parent and the second half of the first parent. Therefore, there are 16 offsprings produced from this process.
6. Mutation
   The goal of mutation is to alter individuals by making word replacements [6]. Two offsprings are randomly selected for this. For each selected offspring, the number of word replacements is the integer of the offspring's length divided by 15. Words are targeted for replacement only if their part of speech (POS) is either adjective, common noun, verb, or pronoun. Thus, for example, proper nouns such as name of persons or places are not changed. Indonesian POS tagger designed by [14] is used to tag words based on their part of speech.
   A targeted word is replaced by another word with a high similarity. Hence, FastText is utilized again. It is trained to find a vector representation of any Indonesian word using Indonesian Wikipedia corpus that was last updated on October 1st 2020. To replace a word, let's say *x*, FastText looks for 15 most similar words to *x*. Afterwards, those words are tagged by the POS tagger. If there is one or more words whose POS tag is the same as *x*, one of them will be randomly selected to replace *x*. Otherwise, the most similar word, i.e. the word with highest similarity to *x*, is chosen to be the replacement.
7. Population update
   After offsprings are produced from crossover and mutation, they are descendingly sorted based on their fitness. The population is later updated by combining 12 fittest offsprings and 4 fittest old individuals from the mating pool. Some old individuals are not excluded because crossover and mutation may not result in better offsprings.
8. Stopping criteria
   The author obfuscation model iteratively runs the selection to population update process until one of the stopping criteria is met. First, if there is already one or more obfuscated articles or individuals in the population, i.e. the one that the authorship attribution model misidentifies its author. The model will immediately stop without continuing to the selection process if this criteria is already met right after population initialization. Second, if the maximum number of iterations which is set to 40 is reached.
9. Output
   The output of the obfuscation process on an article is the fittest individual in the population regardless of whether the model can obfuscate the article or not. If the model cannot obfuscate the article on the first run, it will run again until the article is obfuscated or up to 3 runs. The best result of all runs will be chosen as the final output.

3. RESULTS AND DISCUSSION

3.1 Evaluation of Authorship Attribution Models

The author obfuscation model requires black-box knowledge of the target authorship attribution model to test its performance. Consequently, six authorship attribution models are created using the Support Vector Machine (SVM) algorithm with a linear kernel. Each model uses different feature representations. Table 1 shows the accuracy and F1-score of each authorship attribution model. Word unigram performs better than word bigram, and character 5-gram performs better than other character n-gram representations. A new model with both features combined works better than the others. The average accuracy of all models is 0.6222 means that the models are not good enough in identifying the author of the articles, but they are not that bad either.

Table 1 Testing result of authorship attribution models using linear-SVM

Feature representation	Accuracy	F1-score
word unigram	0.6194	0.6153
word bigram	0.6005	0.6021
character 4-gram	0.595	0.5979
character 5-gram	0.6316	0.6333
character 6-gram	0.6011	0.6018
word unigram + character 5-gram	0.6855	0.6834
Average	0.6222	0.6223

3.2 Evaluation of Author Obfuscation Model

After the accuracy and F1-score of all authorship attribution models are recorded, author obfuscation is evaluated on each of these models by including its attribution probability in the author obfuscation model's fitness function. Author obfuscation is evaluated based on three parameters, *safety*, *soundness*, and *sensibleness*.

3.2.1 Safety evaluation

The author obfuscation model is safe if the authorship attribution model is unable to correctly identify the authors of the articles in the obfuscated test data, so its performance drops from the previous one. Safety is measured by doing author obfuscation that involves the attribution probability of a specific authorship attribution model. This authorship attribution is reevaluated using the obfuscated test data. From the six authorship attribution models, the average decrease in accuracy and F1-score is 0.3018 and 0.3147, respectively. Therefore, the obfuscation model is said to be safe because it can decrease the accuracy and the F1-score of the given authorship attribution models. Table 2 shows the results of safety testing in more detail.

Table 2 Safety evaluation result

Feature representation	Pre-obf. accuracy	Pre-obf. F1-*score*	Post-obf. accuracy	Post-obf. F1-*score*	Accuracy drop	F1-*score* drop
word unigram	0.6194	0.6153	0.3711	0.3358	0.2483	0.2795
word bigram	0.6005	0.6021	0.3066	0.3116	0.2939	0.2905
character 4-gram	0.595	0.5979	0.2961	0.2903	0.2989	0.3018
character 5-gram	0.6316	0.6333	0.2944	0.283	0.3372	0.3503
character 6-gram	0.6011	0.6018	0.28	0.2609	0.3211	0.3409
word unigram + character 5-gram	0.6855	0.6834	0.3738	0.3581	0.3117	0.3253
				Average	0.3018	0.3147

The target authorship attribution model might be different from the one used in this research. For example, the author obfuscation model uses attribution probability of word unigram model. It means that the author obfuscation model only tries to fool any target model using word unigram representation. However, the target may use an authorship attribution

model with different feature representation, feature set, or even classification algorithm. Hence, safety evaluation is carried out again by taking the obfuscated articles from the obfuscation process against the character 5-gram model to be tested on the target model with the following scenario:

1. The feature unit is the same, but the value of n is different (character 6-gram, code: A)
2. Different unit and n value (word unigram, code: B)
3. Unit and n value are the same but combined with another feature representation (word unigram + character 5-gram, code: C)

Also, there is an additional scenario where the obfuscated articles from the obfuscation process against the word bigram model is evaluated on an target model with word unigram + character 5-gram representation (code: D). The evaluation result with this scenario is shown in Table 3. The accuracy and F1-score shown in the table refer to the accuracy and F1-score of the target model. The average safety of the author obfuscation model decreased significantly from 0.3018 to 0.1179. This result indicates that the effectiveness of the author obfuscation model in maintaining safety depends on the target model. The closer the target model to the one used in the obfuscation model's fitness function, the greater the safety, and vice versa.

Table 3 Advanced safety evaluation result

Scenario code	Pre-obf. accuracy	Pre-obf. F1-*score*	Post-obf. accuracy	Post-obf. F1-*score*	Accuracy drop	F1-*score* drop
A	0.6011	0.6018	0.4078	0.3962	0.2049	0.2056
B	0.6194	0.6153	0.4833	0.4629	0.1361	0.1524
C	0.6855	0.6834	0.6011	0.5914	0.0844	0.092
D	0.6855	0.6834	0.6394	0.6357	0.0461	0.0477
				Average	0.1179	0.1244

3.2.2 Soundness evaluation

The author obfuscation model is sound if the obfuscated articles are related to their originals. Soundness evaluation is carried out to ensure that the news content does not change much by calculating the average level of similarity of all articles in the obfuscated test data with the original. Table 4 shows the soundness of the obfuscation model against the aformentioned authorship attribution models. The obfuscation model gets the soundness score of 0.7817 on average. The numbers shown in the table do not include the unchanged articles which have similarity value of 1.

Table 4 Soundness evaluation result

Target model	Soundness
word unigram	0.7916
word bigram	0.7851
character 4-gram	0.7818
character 5-gram	0.776
character 6-gram	0.7764
word unigram + character 5-gram	0.7792
Average	0.7817

Qualitative result will give better insight on how the obfuscation preserves the article content.

Table 5 shows how a sentence from some articles changes after obfuscation process. We can see that modifications in some sentences do not change their content like in the example number 1,2, and "dimakamkan" → "dikebumikan" in 6. The model is able to modify the words into their appropriate synonym. However, the model can also introduces modifications that do not make any sense and change the sentence meaning into something else like in "bertindak" →

"mengangkang", "tegas" → "sikap", and "sanksi" → "pemberlakuan". Sometimes the modification introduces typo like in sentence 4 where "masyarakat" is replaced with "masyarakaat". There are couple of instances where a word's lemma remains the same but its form changes. This kind of modification looks fine in "mengeluarkan" → "keluarkan" in sentence 5, but not in "tewas" → "menewaskan" in sentence 3. Some words might be actually modified into their synonym, but the change of the word form makes it looks bad. For example, the word "kawasan" is replaced by "berwilayah" instead of "wilayah". It goes the same way for the word "dikeroyok" is modified into "menghajar" instead of "dihajar". These inappropriate modifications might happen due to the lack of accuracy of the POS tagger in tagging words; for instance, a noun is replaced with a verb because the POS tagger tags it as a verb. The lack of strict rules in the modification method might also causes this issue; for example, the modification method is not restricted to change active verbs into passive verbs.

Table 5 Example of word modifications made by the author obfuscation model

No.	Original sentence	Obfuscated sentence
1	Kalau 10-15 persen dari **jumlah** penduduk Indonesia sudah enterpreneur …	Kalau 10-15 persen dari **total** penduduk Indonesia sudah enterpreneur …
2	… Berli Hamdani **mengakui** kenaikan kasus terjadi …	… Berli Hamdani **menegaskan** kenaikan kasus terjadi …
3	Andik **tewas dikeroyok** di salah satu …	Andik **menewaskan menghajar** di salah satu …
4	… **masyarakat** diharapkan dapat segera beraktivitas normal …	… **masyarakaat** diharapkan dapat segera beraktivitas normal …
5	Sehingga aparat tidak perlu **bertindak tegas** sampai **mengeluarkan sanksi** pidana.	Sehingga aparat tidak perlu **mengangkang sikap** sampai **keluarkan pemberlakuan** pidana.
6	Yudi mengatakan Teddy akan **dimakamkan** di **kawasan** Jalan Maribaya	Yudi mengatakan Teddy akan **dikebumikan** di **berwilayah** Jalan Maribaya

3.2.3 Sensibleness evaluation

The author obfuscation model is sensible if people generally can understand the obfuscated articles and they do not know that there have been changes by machines. The sensibleness evaluation is done manually by asking 6 evaluators from Universitas Gadjah Mada students to provide scores from 0 to 4 on the obfuscated articles based on their content, grammar, and typo. Each person has 25 obfuscated articles that differ from one person to another. The corresponding original articles are provided as a reference in making an evaluation. Table 6 shows the average sensibleness score given by each evaluators. The first three evaluators are given the obfuscated articles with highest similarity (more than 0.82) to the original, whereas the other three are given the least similar (less than 0.75) obfuscated articles. It is done to determine the relationship between soundness and sensibleness. It also shows that the sensibleness of the obfuscated articles can be high or low. We can see that sensibleness is proportional to the soundness. If the soundness is high, so is the sensibleness, and vice versa.

Table 6 Sensibleness evaluation result

Evaluator	Average score
A	3.896
B	3.296
C	3.001
D	2.08
E	1.76
F	1.391
Average	2.571

Table 7 and Table 8 show two examples of obfuscated articles with high and low sensibleness, respectively. The piece of the obfuscated article shown in Table 7 still makes sense and the modifications happen on it do not break its semantic a lot. On the other hand, the modifications happen on the article shown in Table 8 are gramatically messy and the article itself does not make sense after it gets obfuscated.

Table 7 Example of piece of an obfuscated article with high sensibleness

Original article	Obfuscated article
"Beliau dilarikan ke rumah sakit, **itu** cepat waktunya, saya **menerima** kabar **sakitnya** kan kemarin hari Selasa (4/8)," **kata** dia. Teddy **diketahui** sempat muncul ke publik pada sepekan lalu, tepatnya Rabu (29/7) di Gedung Mohamad Toha, Kecamatan Soreang, Kabupaten Bandung. Pada **saat itu**, Teddy mendampingi Bupati Bandung, Dadang Naser dalam agenda Rapat Koordinasi (Rakor) Implementasi SIPD di Lingkungan Pemerintah Kabupaten Bandung **Tahun** 2020. Sementara itu, Yudi mengatakan Teddy akan **dimakamkan** di **kawasan** Jalan Maribaya, Desa Kayuambon, Kecamatan Lembang, Kabupaten Bandung Barat.	"Beliau dilarikan ke rumah sakit, **tersebut** cepat waktunya, saya **mendapat** kabar **sakitan** kan kemarin hari Selasa (4/8)," **diartikan** dia. Teddy **diketahu** sempat muncul ke publik pada sepekan lalu, tepatnya Rabu (29/7) di Gedung Mohamad Toha, Kecamatan Soreang, Kabupaten Bandung. Pada **ketika tersebut**, Teddy mendampingi Bupati Bandung, Dadang Naser dalam agenda Rapat Koordinasi (Rakor) Implementasi SIPD di Lingkungan Pemerintah Kabupaten Bandung **Awal** 2020. Sementara tersebut, Yudi mengatakan Teddy akan **dikebumikan** di **berwilayah** Jalan Maribaya, Desa Kayuambon, Kecamatan Lembang, Kabupaten Bandung Barat.

Table 8 Example of piece of an obfuscated article with low sensibleness

Original article	Obfuscated article
Sebab, **jumlah** kasus **positif** di tiga negara **itu** di atas 3,5 juta jiwa. Berikut sejumlah kabar **corona dunia**: Filipina Laporkan Penambahan Kasus Baru Corona Terendah Filipina mencatatkan **penambahan** 2.218 **kasus** baru virus corona pada Rabu (2/9). **Jumlah tersebut merupakan** yang **terendah** dalam lima **minggu** terakhir. Filipina **merupakan negara jumlah** kasus **virus corona tertinggi** di Asia Tenggara.	Sebab, **persentase** kasus **taknegatif** di tiga negara **ini** di atas 3,5 juta jiwa. Berikut sejumlah kabar **coronado saat**: Filipina Laporkan Penambahan Kasus Baru Corona Terendah Filipina mencatatkan **pengganti** 2.218 **diinvestigasi** baru virus corona pada Rabu (2/9). **Persentase itu menjadikan** yang **tertinggi** dalam lima **pagi** terakhir. Filipina **merupakam tersentralisasi kalisusu** kasus **filovirus coronado bertinggi** di Asia Tenggara.

4. CONCLUSIONS

An author obfuscation model on Indonesian news article is proposed in this paper by implementing genetic algorithm-based approach. The evaluation result shows that this model has a good safety after successfully dropped the attribution accuracy by 0.3018 on average, given the black-box knowledge of the target authorship attribution model. However, it only dropped the attribution accuracy by 0.1179 on average when the given black-box knowledge of the authorship attribution model is different from the actual target. In terms of soundness, the result is pretty good since the model can maintain the similarity of the obfuscated articles to the original by 0.7817. After manually evaluated by human evaluators, the model gains a sensibleness score of 2.571 on a scale of 0 to 4 which indicates that some articles are acceptable in terms of grammar, but not a few are messy.

This study in author obfuscation has many rooms for improvements in the future, one of them is to use a more accurate POS tagger to minimize the modification where a word is replaced by another with different POS tag. Furthermore, applying additional rules in the modification method is also an interesting area for future work.

REFERENCES

[1] M. Koppel, S. Argamon, and A. R. Shimoni, "Automatically Categorizing Written Texts by Author Gender," *Lit. Linguist. Comput.*, vol. 17, no. 4, pp. 401–412, 2002.

[2] M. Sage, P. Cruciata, R. Abdo, J. C. K. Cheung, and Y. F. Zhao, "Investigating the influence of selected linguistic features on authorship attribution using German news articles," in *CEUR Workshop Proceedings*, 2020, vol. 2624.

[3] H. Gomez-Adorno, J.-P. Posadas-Duran, G. Rios-Toledo, G. Sidorov, and G. Sierra, "Stylometry-based Approach for Detecting Writing Style Changes in Literary Texts," *Comput. y Sist.*, vol. 22, no. 1, pp. 47–53, 2018, doi: 10.13053/CyS-22-1-2882.

[4] E. Lundeqvist and M. Svensson, "Author profiling: A machine learning approach towards detecting gender, age, and native language of users in social media," Uppsala Universitet, 2017.

[5] T. Gröndahl and N. Asokan, "Effective writing style transfer via combinatorial paraphrasing," in *Proceedings on Privacy Enhancing Technologies*, 2020, vol. 2020, no. 4, pp. 175–195, doi: 10.2478/popets-2020-0068.

[6] A. Mahmood, F. Ahmad, Z. Shafiq, P. Srinivasan, and F. Zaffar, "A Girl Has No Name: Automated Authorship Obfuscation using Mutant-X," in *Proceedings on Privacy Enhancing Technologies*, 2019, vol. 2019, no. 4, pp. 54–71, doi: 10.2478/popets-2019-0058.

[7] M. Potthast, M. Hagen, and B. Stein, "Author Obfuscation : Attacking the State of the Art in Authorship Verification," 2016, [Online]. Available: https://pan.webis.de/downloads/publications/papers/potthast_2016a.pdf.

[8] Y. Keswani, H. Trivedi, P. Mehta, and P. Majumder, "Author masking through translation," in *CEUR Workshop Proceedings*, 2016, vol. 1609, pp. 890–894.

[9] M. Mansoorizadeh, T. Rahgooy, M. Aminiyan, and M. Eskandari, "Author obfuscation using WordNet and language models," in *CEUR Workshop Proceedings*, 2016, pp. 1–8.

[10] G. Karadzhov, T. Mihaylova, Y. Kiprov, G. Georgiev, I. Koychev, and P. Nakov, "The Case for Being Average: A Mediocrity Approach to Style Masking and Author Obfuscation," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, 2017, pp. 173–185.

[11] Y. Yunitasari, A. Musdholifah, and A. K. Sari, "Sarcasm Detection For Sentiment Analysis in Indonesian Tweets," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 13, no. 1, p. 53, 2019, doi: 10.22146/ijccs.41136.

[12] C. F. Lima, F. G. Lobo, and M. Pelikan, "From Mating Pool Distributions to Model Overfitting," in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, 2008, pp. 431–438, doi: 10.1145/1389095.1389174.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[14] A. Dinakaramani, F. Rashel, A. Luthfi, and R. Manurung, "Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus," in *Proceedings of the International Conference on Asian Language Processing 2014, IALP 2014*, Oct. 2014, pp. 66–69, doi: 10.1109/IALP.2014.6973519.