The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7505460

IMPLEMENTATION MODEL OF MACHINE LEARNING ON NEWS CLASSIFICATION INFORMATION SYSTEM

MUHAMAD NUR GUNAWAN[1], NURYASIN[2], ARIEF AKBAR HIDAYAT[3] and SYOPIANSYAH JAYA PUTRA[4]

[1]Lecture and Researcher, Information System Department, Syarif Hidayatullah State Islamic University Jakarta, Jakarta, Indonesia. Corresponding author's Email: nur.gunawan@uinjkt.ac.id
[2]Lecture and Researcher, Information System Department, Syarif Hidayatullah State Islamic University Jakarta, Jakarta, Indonesia.
[3]Scholar, Information System Department, Syarif Hidayatullah State Islamic University Jakarta, Jakarta, Indonesia.
[4]Associate Professor, Information System Department, Syarif Hidayatullah State Islamic University Jakarta, Jakarta, Indonesia.

Abstract

Text classification is a grouping of text data that has not been grouped into groups automatically. The classification of news texts is done by the editor by reading the entire text first, so it takes a long time. For that we need a way to classify news automatically that can cut down the process. This study aims to classify news texts automatically with a text mining approach. This study uses the K-Nearest Neighbor algorithm which has simplicity and efficiency in classifying various types of text. To simplify the research flow, the CRISP-DM (Cross-industry standard process for data mining) method is used, which is the standard in conducting data analysis in industrial applications. The results showed satisfactory results, namely precision, recall, F1-Score and accuracy reached 95% with a value of k = 11. After the text classification application was made and an experiment was carried out by entering several new news texts, only a few seconds the text could be classified correctly. This study shows that the K-Nearest Neighbor algorithm can be used for news text classification and text classification applications can help cut the classification process time.

Keywords: Text Mining, News Text Classification, K-Nearest Neighbor, CRISP-DM.

A. INTRODUCTION

Text mining is taking the essence of information from data in the form of text through statistical science (Korde, 2012). Text classification is to group data that has not been grouped into groups automatically (Chan et al., 2001). Many studies discuss text classification, and from these studies, there are three algorithms that are often used in text classification [3], namely: (1)Support Vector Machine (SVM) [4]–[8], (2)K-Nearest Neighbor (KNN) [5], [9]–[13], (3)Naïve Bayes [5]–[7], [11], [12], [14], "The Naïve Bayes algorithm is a simple and efficient method used in text classification. However, it's not very efficient because it doesn't model text well, nor does it provide a good selection of features." [4]. "This algorithm does not take into account the number of events, which is a potentially useful source of additional information." [11].

Naïve Bayes has superior accuracy results than other algorithms in research [6] and [14] which yield 45% and 86% accuracy, respectively, with the number of documents 2.6 million and 300. Poor results in research [6] are caused by unbalanced number of each data label.

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7505460

The Support Vector Machine has the advantage of being able to produce a good classification model with a small amount of data. The weakness of SVM is that it is difficult to apply for very large amounts of data and dimensions [15].

In research [4], very high accuracy (90%) was produced by SVM with a total of 5000 and 2000 documents.

The K-Nearest Neighbor algorithm was introduced as one of the most widely used text classification algorithms because of its simplicity and efficiency in classifying various types of text. However, it has a weakness in determining the effective K parameter [4].

K-Nearest Neighbor has superior results than Naïve Bayes in studies [11], [12], [9] and [13]. In research [11] KNN produced a very high accuracy (98%) with a total of 21,000 documents. But in research (Trstenjak et al., 2014) the accuracy of KNN on one of the labels is not good because the text preprocessing used is not optimal, the number of documents applied is 500.

In a Reuters Institute Digital News Report 2019 report by [16] stated that "Across all countries, most people agree that the news media are always up-to-date on what is happening (62%), but only half (51%) say that the news media help them understand the news."

Most of the news media began to build websites to present information and news online within the last ten years [4]. During this period, news categorization still uses human or manual labor. The data that comes in and must be categorized is not proportional to the time available so that editors will find it difficult to categorize it, especially articles that have almost the same object of discussion but different topics such as technology and science. It is the difference in categories in similar objects that requires an editor to know the contents of the article to be uploaded as a whole so that it is then put into the right category. Automatic news categorization with machine-learning method can be the solution [17].

Based on the weaknesses and strengths of the algorithm, as well as the research that has been described previously, considering that the text has large dimensions, the algorithm that will be used in this study is K-Nearest Neighbor.

B. METHOD

A. Dataset

In the experiments carried out, the material needed is a dataset derived from the scraping of the bbc.com news portal website which was carried out from May to June 2020 with coverage of five categories on the website, namely: entertainment, health, business, technology and science.

B. Process

The text classification process is applied according to the CRISP-DM method which has several stages, namely, business understanding; data understanding which includes web scraping and data merging; data preparation which includes text preprocessing and features extraction; modeling which includes KNN algorithm and hyperparameter tuning; Evaluation

ISSN 1533-9211

DOI 10.5281/zenodo.7505460

which includes confusion matrix and classification report; Deployment which includes interface creation, backend and deploy to PaaS.

C. CRISP-DM method

In this study, the CRISP-DM method was used to conduct text classification experiments. The CRISP-DM method was chosen because it has been proven to perform data mining or text mining based on a survey conducted by KDNuggets [32] as well as research conducted by [33]. In CRISP-DM there are six stages that must be carried out, namely:

1) Business Understanding

The business understanding phase relates to the objectives to be carried out. Based on the facts obtained from previous studies, there is a gap that there are still many classifications of news texts that are done manually by editors and this is stated in the research objectives. Therefore, the business goal to be achieved is to classify news texts automatically.

2) Data Understanding

The first step of this phase is data collection. Data collection is done by means of web scraping on the bbc.com news portal with a limit only on the categories of science, health, technology, entertainment and business.

Data collection is done by means of web scraping on the bbc.com news portal with a limit only on the categories of science, health, technology, entertainment and business.

After the data is collected, 31 csv files are obtained in each folder (there are 5 folders with different categories). If you add up, there are a total of 155 csv files. To simplify the analysis, the next step is to combine all the files first into one csv file.

3) Data Preparation

The purpose of data preparation is to create quality data. In text mining, the data preparation carried out is text preprocessing and features extraction. The first step in text preprocessing is to check the quality of the text data. After being traced, it turned out that the amount of text data obtained had redundancy with an initial number of 3685 lines, then after filtering it was obtained 813 lines with the number of each label: business 290 lines, entertainment 180 lines, technology 151 lines, science 121 lines and health 71 lines.

From these details, there are discrepancies between labels. To balance the labels, 100 lines were sampled from each label (except for the health label). The next step is to check the presence or absence of noise in the text data. It turns out that there is noise in some texts such as website emails, journalists' social media and sentences requesting feedback that should not be in the news. The interesting thing about these noises is that they have a pattern, namely in the last 2-3 sentences. For this reason, some text preprocessing is carried out in the following flow

ISSN 1533-9211

DOI 10.5281/zenodo.7505460

D. Modeling

After the dataset has been cleaned and converted to vector/numeric form, the next step is modeling. Because the purpose of business understanding is to classify news texts automatically, the appropriate task for modeling in this research is classification. In the classification task, the KNN algorithm is used.

E. Evaluation

After the model has been created, then do an evaluation to find out whether the results are good or not. Because this task is a classification, the evaluation used is to calculate precision, recall, f1 score, and accuracy. Then it is also evaluated by the confusion matrix.

F. Deployment

The process carried out is to deploy the model that has been made into a website so that it can be used by end users. Interface creation and integration to the backend uses the flask framework which supports the python programming language and its own web server. To deploy to the general public, one of the Platform as a Service (PaaS) providers is used which is required to use git for deployment requirements.

In making interfaces, use case diagrams are needed to describe the interactions that can be carried out between actors and the system. To create a use case diagram, it is necessary to identify the actors and identify the use case first.

In making interfaces, use case diagrams are needed to describe the interactions that can be carried out between actors and the system. To create a use case diagram, it is necessary to identify the actors and identify the use case first.

Table 1: Identification of actors

No	Actor	Description
1.	Editor	Input text to be classified automatically by the system

Table 1 identifies actors or users who will use the system. In this study are news editors. News editors can input news text without having to read it, because it will be classified automatically.

Table 2: Identification of Use Cases

No	Use Case Name	Description	Actor
1.	Text classification	Put ready-made text or input text directly to be classified automatically by the system	Editor

Table 2 describes the use cases created. In this research, the purpose of making the system is text classification. This system can classify text news automatically according to the input entered. As shown in Figure 1.

ISSN 1533-9211

DOI 10.5281/zenodo.7505460

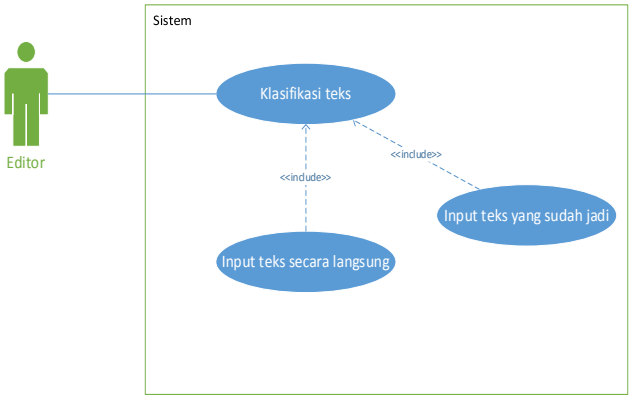

Figure 1: Use case diagrams

Figure 1 describes the use case of the system, where the editor can classify news text automatically by inputting text directly or inputting ready-made text. The system made is web-based, which has an interface as shown in Figure 2.

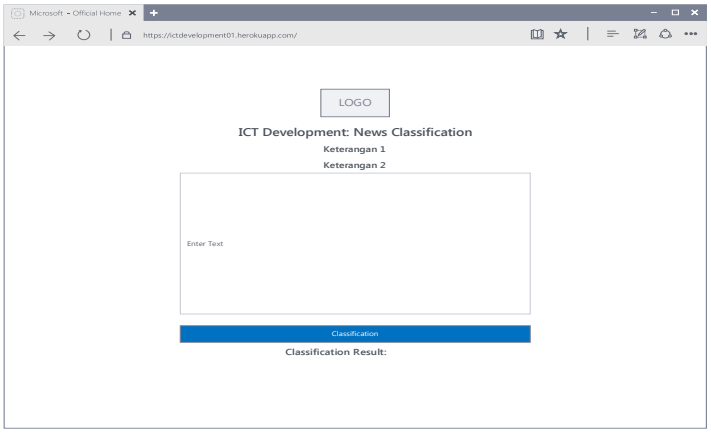

Figure 2: System Wireframes

Figure 2 describes the wireframe system for news classification. There is a white box which is a place to enter text. The blue button serves to classify the text that has been inputted.

C. RESULT AND DISCUSSION

The experiment was carried out through the Google Collaboratory cloud with detailed hardware specifications of 13 GB RAM, 33 GB HDD, 2 vCPU 2-core Xeon 2.2 GHz and Linux operating system.

The dataset used has the following information:

ISSN 1533-9211

DOI 10.5281/zenodo.7505460

```
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   judul     813 non-null    object
 1   isi       813 non-null    object
 2   tanggal   813 non-null    object
 3   kategori  813 non-null    object
```

(a)

```
business          290
entertainment     180
technology        151
science           121
health             71
```

(b)

Figure 3: (a) dataset column information (b) number of labels for each category

The features selection is carried out by taking only two columns that will be used for modeling, namely: the content column and the category. After that, text preprocessing is done to make the dataset more qualified. After completion, features extraction and modeling are done. Initially, the number of data for each label was used, then two experiments were carried out. In the first experiment, all data were used for modeling. The second experiment, to overcome data inequality, under sampling was carried out, namely random sampling of 100 lines per category/label (except health).

The models produced by the two experiments were not much different. To improve the results of the model, hyper parameter tuning was carried out in experiment one and experiment two. Hyper parameter tuning that is done is to find the optimal k value from 1-50 and cross validation is carried out 5 times. The result is that in the first experiment the best value of k=29 was obtained and in the second experiment the best value of k=11 was obtained. Each value of k is entered into the parameters of the KNN algorithm. The resulting model in each experiment is increasing. Here are the results in each category.

Business

	Predicted: Not B	Predicted: B
Actual: Not B	TN=95	FP=2
Actual: B	FN=3	TP=18

Entertainment

	Predicted: Not E	Predicted: E
Actual: Not E	TN=93	FP=1
Actual: E	FN=0	TP=24

Health

	Predicted: Not H	Predicted: H
Actual: Not H	TN=97	FP=0
Actual: H	FN=0	TP=21

Science

	Predicted: Not S	Predicted: S
Actual: Not S	TN=85	FP=2
Actual: S	FN=0	TP=31

Technology

	Predicted: Not T	Predicted: T
Actual: Not T	TN=96	FP=1
Actual: T	FN=3	TP=18

Figure 4: Confusion Matrix of Each Category

ISSN 1533-9211

DOI 10.5281/zenodo.7505460

In the confusion matrix is presented with the help of the Scikit Learn library. In the confusion matrix with the scikit learn library there is support which is the number of true response samples representing each class. The resulting support in this experiment is 118.

The second experimental model (dataset with samples) had the best results, namely accuracy, precision, recall and the f1 score reached 95%. The model is then deployed on the website. The following is a display of the text classification application.

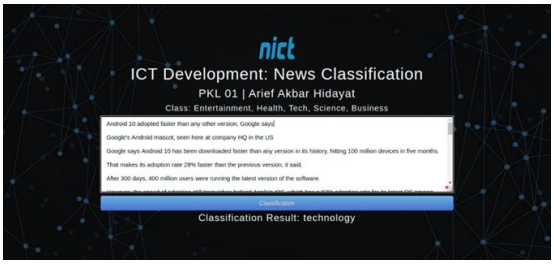

Figure 5: Text classification application display image

Web-based application for text classification has the benefit of saving time for text classification tasks. In just a few seconds, the inputted text is classified automatically without having to read the entire contents first. The classification results are quite satisfactory. When tested by the new testing data, it can produce the appropriate class. Even so, it still has several weaknesses, one of which is the error in categorizing news that is included in category A but contains many words that describe other categories.

In the experiments carried out, the interesting thing that happened was that although there were discrepancies in the health label with other labels, the classification results were still good compared to research [6] which the results were not good using the Naïve Bayes algorithm. One of the reasons why the experimental results are good is because the text preprocessing that has been done is good, since the data is in English, the stop words data provided are complete and stored in the natural language toolkit (nltk) library. In addition, the dataset used is news text whose language is more formal and structured, so that the results of each label remain consistent compared to research [10] which had poor results on one label.

Another reason for the good results is that hyper parameter tuning is done which can improve the results of the model, because of all the studies that have been reviewed, none of them have used hyper parameter tuning. The way hyper parameter tuning works is by trying every possibility of a predefined parameter (k number variable declaration 1-50). In performing hyper parameter tuning, it takes quite a long time to process it (depending on how many parameters will be determined), this experiment itself takes about three hours to get optimal parameter recommendations. From the selection of appropriate parameters, the evaluation results of the model also increase.

Another reason the results are still good despite the data discrepancy is because of the uniqueness of the words in the health label compared to other labels. For example, in the health

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7505460

label there is the word covid patient, which means it cannot be interpreted ambiguously in other categories, but only for health. In contrast to other labels such as technology, business, entertainment and science, which may have multiple meanings, such as the word google, which has a different meaning for each label. In the technology and business labels, Google refers to technology companies. On the entertainment label, Google can refer to the word Google Play Music.

In this study, there is a limitation that is only using the KNN algorithm. Then the dataset provided is still small because the dataset was taken by web scraping in a span of only one month. The classified topics are only limited to five topics and the language used is still English. In its implementation, it is only deployed through the website and is limited to text input forms. Therefore, for further research, it is hoped that other algorithms such as SVM, Decision Tree and others can be used in the hope of getting better results. Then the dataset used to be more reproduced and the topics are more diverse. Then the model is deployed on a website that can classify documents through file uploads.

D. CONCLUSION

In this study, it was proven that the KNN algorithm succeeded in automatically classifying news texts with good results, namely 95% for precision, recall, f1 score and overall accuracy. The resulting model can classify the five classes well based on the content of the news text. When testing with the new news text, the model can produce a good class even though some of the tests are wrong. The reason for making the model with good results is due to good text preprocessing, the dataset has a structured, formal sentence structure and is supported by the availability of complete stop words and hyper parameter tuning that improves model performance.

The model created and applied to a web application can classify news text in a fast time so that it can be used to streamline news text classification tasks. Users (editors) can put the text of the news or write from scratch without the need to re-read the contents of the news, because it will be automated by the model that has been created.

E. ACKNOWLEDGEMENTS

Research on Neural Network or Machine Learning supported by Syarif Hidayatullah State Islamic University Jakarta, Indonesia.

References

1. V. Korde, “Text Classification and Classifiers: A Survey,” Int. J. Artif. Intell. Appl., vol. 3, no. 2, pp. 85–99, 2012, doi:10.5121/ijaia.2012.3208.
2. C.-H. Chan, A. Sun, and E.-P. Lim, “Automated Online News Classification with Personalization,” 4th Int. Asian Digit Conference. Libra., pp. 1–10, 2001, [Online]. Available: http://ncsi-net.ncsi.iisc.ernet.in/gsdl/collect/icco/index/assoc/HASH01de.dir/doc.pdf.
3. R. Jindal, R. Malhotra, and A. Jain, “Techniques for text classification: Literature review and current trends,” Webology, vol. 12, no. 2, pp. 1–28, 2015.

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7505460

4. SMH Dadgar, MS Araghi, and MM Farahani, “A novel text mining approach based on TF-IDF and support vector machine for news classification,” Proc. 2nd IEEE Int. conf. eng. Technol. ICETECH 2016, no. March, pp. 112–116, 2016, doi:10.109/ICETECH.2016.7569223.
5. BY Pratama and R. Sarno, “Personality classification based on Twitter text using Naive Bayes, KNN and SVM,” Proc. 2015 Int. conf. Software Data. eng. ICODSE 2015, pp. 170–174, 2016, doi:10.109/ICODSE.2015.7436992.
6. T. Pranckevičius and V. Marcinkevičius, “Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification,” Balt. J. Mod. Comput., vol. 5, no. 2, pp. 221–232, 2017, doi:10.22364/bjmc.2017.5.2.05.
7. Z. Jianqiang and G. Xiaolin, “Comparison research on text pre-processing methods on twitter sentiment analysis,” IEEE Access, vol. 5, pp. 2870–2879, 2017, doi:10.109/ACCESS.2017.2672677.
8. X. Fang and J. Zhan, “Sentiment analysis using product review data,” J. Big Data, vol. 2, no. 1, 2015, doi:10.1186/s40537-015-0015-2.
9. Arifin, “Classification of Emotions in Indonesian Texts Using K-NN Method,” Int. J. Inf. electrons. Eng., vol. 2, no. 6, 2012, doi:10.7763/ijiee.2012.v2.237.
10. B. Trstenjak, S. Mikac, and D. Donko, “KNN with TF-IDF based framework for text categorization,” Procedia Eng., vol. 69, pp. 1356–1364, 2014, doi:10.1016/j.proeng.2014.03.129.
11. V. Bijalwan, V. Kumar, P. Kumari, and J. Pascual, “KNN based machine learning approach for text and document mining,” Int. J. Database Theory Appl., vol. 7, no. 1, pp. 61–70, 2014, doi:10.14257/ijdta.2014.7.1.06.
12. ZE Rasjid and R. Setiawan, “Performance Comparison and Optimization of Text Document Classification using k-NN and Naïve Bayes Classification Techniques,” Procedia Comput. Sci., vol. 116, pp. 107–112, 2017, doi:10.1016/j.procs.2017.10.017.
13. J. Rajshree, SB Gaur, CKR, and M. Amit, “Text Classification using KNN with different Features Selection Methods Abstra,” Int. J. Res. publ. Vol. 8 – Issues. 1, July 2018 Text, no. August, 2018.
14. MA Fauzi, AZ Arifin, SC Gosaria, and IS Prabowo, “Indonesian News Classification Using Naïve Bayes and Two-Phase Feature Selection Model,” Indonesia. J. Electr. eng. Comput. Sci., vol. 2, no. 3, pp. 401–408, 2016, doi:10.11591/ijeecs.v2.i2.pages.
15. Suyanto, Data Mining for Classification and Clustering of Data. Informatics Bandung, 2017.
16. N. Newman, R. Fletcher, A. Kalogeropoulos, and RK Nielsen, “Digital News Report 2019,” pp. 70–72, 2019, doi:10.2139/ssrn.2619576.
17. D. Ariadi and K. Fithriasari, "Classification of Indonesian News Using the Naive Bayesian Classification Method and Support Vector Machine with Confix Stripping Stemmer," J. SCIENCE AND ART ITS Vol. 4, No.2, vol. 4, no. 2, pp. 248–253, 2015.
18. KLSumathy and M. Chidambaram, “Text Mining: Concepts, Applications, Tools and Issues An Overview,” Int. J. Comput. Appl., vol. 80, no. 4, pp. 29–32, 2013, doi:10.5120/13851-1685.
19. K. Kowsari, KJ Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text classification algorithms: A survey,” Inf., vol. 10, no. 4, pp. 1–68, 2019, doi:10.3390/info10040150.
20. K. Ganesan, “All you need to know about text preprocessing for NLP and Machine Learning,” 2019. https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html (accessed Jun. 11 , 2020).
21. B. Santosa and A. Umam, Data Mining and Big Data Analytics. Pustaka Media Publisher, 2018.

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7505460

22. Informatics, "K-Nearest Neighbor (K-NN) Algorithm | INFORMATIKALOGI," 2017. https://informatikalogi.com/algoritma-k-nn-k-nearest-neighbor/#1 (accessed Jun. 14, 2020).
23. Informatics, "Vector Space Model (VSM) and Distance Measurement in Information Retrieval (IR) | INFORMATIKALOGI," 2016. https://informatikalogi.com/vector-space-model-pengukur-jarak/#1 (accessed Jul. 14, 2020).
24. S. Narkhede, "Understanding Confusion Matrix," 2018. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.
25. M. DEI, "Hyperparameter Tuning Explained — Tuning Phases, Tuning Methods, Bayesian Optimization, and Sample Code!," 2019. https://towardsdatascience.com/hyperparameter-tuning-explained-d0ebb2ba1d35 (accessed Jul. 02, 2020).
26. Sanjay M, "Why and how to Cross Validate a Model," 2018. https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f (accessed Jul. 17, 2020).
27. Python (programming language) - Wikipedia, "Python (programming language) - Wikipedia," 2020. https://en.wikipedia.org/wiki/Python_(programming_language)#Indentation (accessed Jun. 14, 2020).
28. scikit-learn - Wikipedia, "scikit-learn - Wikipedia," 2020. https://en.wikipedia.org/wiki/Scikit-learn (accessed Jul. 02, 2020).
29. Flask (web framework) - Wikipedia, "Flask (web framework) - Wikipedia," 2020. https://en.wikipedia.org/wiki/Flask_(web_framework) (accessed Jul. 02, 2020).
30. Google, "Colaboratory – Google," 2020. https://research.google.com/colaboratory/faq.html (accessed Jun. 14, 2020).
31. Platform as a service - Wikipedia, "Platform as a service - Wikipedia," 2020. https://en.wikipedia.org/wiki/Platform_as_a_service#Public,_private_and_hybrid (accessed Jul. 17, 2020).
32. Piatetsky, "CRISP-DM, still the top methodology for analytics, data mining, or data science projects," 2014. https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data -mining-data-science-projects.html.
33. Martinez-Plumed et al., "CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories," IEEE Trans. knowl. Data Eng., vol. 4347, no. c, pp. 1–1, 2019.