Feature Extraction Methods and Classification for Malware Incident News

Gugum Gumilar, Eka Budiarto, Maulahikmah Galinium and Charles Lim

October 23, 2023

Feature Extraction Methods and Classification for Malware Incident News

1st Gugum Gumilar
Information Technology Department
Swiss German University
Tangerang, Indonesia
gugum.gumilar@student.sgu.ac.id

2nd Eka Budiarto
Information Technology Department
Swiss German University
Tangerang, Indonesia
eka.budiarto@sgu.ac.id

3rd Maulahikmah Galinium
Information Technology Department
Swiss German University
Tangerang, Indonesia
maulahikmah.galinium@sgu.ac.id

4th Charles Lim
Information Technology Department
Swiss German University
Tangerang, Indonesia
charles.lim@sgu.ac.id

***Abstract*— Studies related to data mining are one of the topics that have received much interest recently, including for the form of unstructured data. One that is commonly discussed is the automatic classification process using machine learning methods. A large amount of data is the main obstacle in the manual classification process. However, there are still many people who have difficulty determining the right combination between feature extraction and classification methods, so with this, we provide suggestions for using a variety of ways that can produce better accuracy in text classification. This research compares several feature extraction methods, including Bag-of-Word (BoW), Term Frequency - Inverse Document Frequency (TF-IDF), and Word2Vec with focusing on the Skip-gram model. On the other hand, this research also uses several classification methods, which include Support Vector Machine (SVM), Decision Tree, Logistic Regression, Gaussian Naive Bayes, K-Nearest Neighbor, Neural Network, Random Forest, and Doc2Vec. This research used two hundred crawled articles from several web blogs that have been labeled manually and have been split into two classes, malware incident news, and non-malware incident news class. The dataset quality was also measured using an open-source Python library known as "Cleanlab".**

Keywords*— *text mining, web crawling, malware incident, document embedding, text classification

I. Introduction

The number of companies switching from conventional to online makes a lot of data traffic spread on the internet. That is certainly an exciting thing for every cybercrime because there is an opportunity to get confidential data. The number of hacking techniques that constantly evolving, it often makes cybersecurity actors difficult to deal with the crimes that attack them. Analysis of cybersecurity continues to be developed, and even sharing related to CTI (Cyber Threat Intelligence) is increasing, but in reality, there are still many methods that have succeeded in penetrating defense gaps.

Information exchange is now more accessible, including sharing news about malware incidents on a website. This way can support cybersecurity in learning new attack techniques, which makes it a reference for strengthening defenses. The way of presenting the information or explanations given by each writer will undoubtedly be different, with the lengthy writing sometimes making it very time-consuming to read. Apart from that, news related to malware does not always explain the incident, many of them explain the research.

The application of the classification method will undoubtedly be very beneficial in responding to data separation conditions. With a machine learning approach, of course, the process can be done easily. Machine learning for the classification process can be applied to distinguish data whether talking about malware incidents or not malware incidents in the news. With the help of the data mining process, news spread on the internet can certainly be extracted and managed more efficiently, especially with the use of web crawling methods. The web crawling process needs to be slightly customized to adjust the data retrieval process, and to produce datasets according to the need.

Training data is essential in the classification process, by reading the data obtained one by one and then labeling or classing it manually. The training data will become the reference in the machine learning process so that the machine can carry out a new data classification process to separate data as needed. The number of methods that are available to use in classification and feature extraction makes us confused about choosing the proper method to process the data we have. Thus, it is necessary to apply the accuracy comparison process for several methods, including classification and feature extraction from unstructured data.

The main objective of this research is to find the best combination between the feature extraction and classification method, by comparing the accuracy and processing time between the methods used. The specific contributions of this paper are as follows:

1. Custom data collection is using the web crawling method and manually labeling data to create the training data.
2. Method comparison not only uses the default settings but also changes the parameter values in it. For example, for the decision tree, use some custom tree depth to ensure that the tree doesn't cause overfitting.

The structure of this paper is designed as follows. Section II will review related research to show the differences in this paper with others. Section III will discuss the framework used by the author to achieve the goal. Section IV will discuss the experimental results. Section V will focus on discussing the conclusions from the results.

II. Related Work

Many experiments were carried out by other researchers to get the best results of accuracy in a classification process, even studies on unstructured data preprocessing. T. B. Shahi and A. K. Pant [1] classify news obtained from national news portals and perform accuracy comparisons covering several methods, including SVM, Naive Bayes, and Neural Networks. In their study, SVM became the method with the best accuracy. G. L. Yovellia Londo, et al. [2] perform a comparison of the accuracy of classification methods, which include SVM, Naive Bayes, and Decision Tree for Indonesian news articles, and in their research, SVM became the method with the best accuracy. H. S. Al-Ash, et al. [3] in their research conducted a classification to detect Indonesian fake news using several methods, which included Random Forest, Naive Bayes, and SVM, and in their research, Random Forest became the method with the highest F1 score. M. G. Hussain, et al. [4] classify news to detect Bangladesh fake news using the Naive Bayes and SVM methods, and the results obtained show that SVM can produce better accuracy. K. Shah, et al. [5] classify text obtained from BBC News in several categories, including business, entertainment, politics, sports, and technology, using several classification methods including Logistic Regression, Random Forest, and KNN, and the results obtained show that Logistic Regression can produce the best accuracy value. O. Mendsaikhan, et al. [6] classify text-based documents taken from online sources related to cybersecurity using the Doc2Vec model. This research will focus on comparing several classification methods, which include SVM, Decision Tree, Logistic Regression, Naive Bayes, KNN, Neural Network, Random Forest, and Doc2Vec to classify text containing malware incident news.

A. Al Hamoud, et al. [7] compared the classification accuracy based on the feature extraction results by comparing BoW, TF, and TF-IDF. S. Garg [8] performs a comparison of the accuracy of several classification methods based on the results of feature extraction, which includes comparisons between BoW, TF-IDF, Word2Vec, and manual features. G. M. Barrientos, et al. [9], in their research, also tried to compare the accuracy of several classification methods using the BoW, TF-IDF, and Word2Vec feature extraction methods. Hoyeon Park, et al. [10] tries to research the impact of word embedding methods on sentiment analysis using the BoW, TF-IDF, and Word2Vec method, which is then compared for their accuracy in several classification methods. N. Chayangkoon and A. Srivihok [11] try to combine BoW with Word2Vec (called BWF) and compare it with BoW and TF-IDF in performing feature extraction and then processing it with several classification methods. This research uses BoW, TF-IDF, and Word2Vec methods and compares the accuracy results obtained from their combination with the classification methods used.

For the evaluation method of accuracy, F. Tempola, R. Rosihan, and R. Adawiyah [12] carry out the validation process using the Holdout method, which works by dividing two sets of data into training and test data and then measuring its accuracy. A. Al Hamoud, et al. [7] measure the performance using K-Fold Cross Validation using k=10. N. Chayangkoon and A. Srivihok [11] also used the same method for measuring performance. G. M. Barrientos, et al. [9] also use K-fold cross-validation, but the difference is in the value of K, which is only 5. K. Pal and B. V. Patel [13] carried out the validation process using the K-Fold cross-validation method and the Holdout method. In their research, they said that to measure the accuracy of the classification method properly, it needs to use different data sets and an excellent method to use is K-fold Cross Validation because can split and train data and test data alternately. This research focuses on using K-Fold cross-validation to measure the accuracy of the classification method.

III. Proposed Method

In the process of classifying the news data obtained, several steps need to be done beforehand, such as manually labeling the training data, normalizing the text, and extracting it into a collection of words, which will then be weighted for each word. The steps taken to achieve this research objective include data collection, manual labeling, preprocessing that includes the text normalization and feature extraction process, classification, and evaluation, as can be seen in Fig. 1.

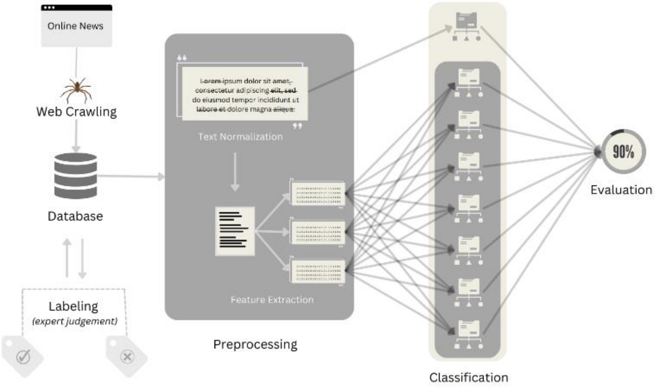
Online News
Web Crawling
Database
Labeling
(expert judgement)
~~Lorem~~ ipsum dolor sit ~~amet,~~ consectetur adipiscing ~~elit, sed~~ do eiusmod tempor incididunt ut ~~labore et~~ dolore magna ~~aliqua.~~
Text Normalization
Feature Extraction
Preprocessing
Classification
90%
Evaluation

Fig. 1. Proposed Framework

A. *Data Collection*

In the data collection process, a text mining method is selected to automate the crawling process from several blog articles [14]. Because each web has its own style and attribute naming, custom web crawling for extracting data needs to be used so that the data could be processed into new information based on the requirements [15]. The news criteria used to conduct data collection focused on, it is the news that only discussed topics that are related to malware, because the target of this research is to classify between news about malware incidents and not malware incidents, and the type of news selected was only English news.

B. *Labeling*

In this step, manual labeling is needed to create a machine-learning pattern in which data that already has a class or label will then be divided into two sets, namely training data and test data. The label itself is between 1 and 0, with the meaning of 1 being malware incident news, and 0 being non-malware incident news. The reference for the labeling process is based on whether the news contains an attack victim.

The training data will be used as a learning reference and then used to predict test data, as mentioned by [16]. After the data has been labeled manually, the data quality measurement would be carried out using an open source library on Python Programming, called “cleanlab” which is proposed by C. G. Northcutt, et al. [17] trying to detect mislabeled data automatically by identifying errors in dataset label based on the estimation of join distribution between unknown label with the noisy label.

C. *Text Normalization*

Based on the word comparison process, it tends to be sensitive. Changing each letter to lowercase is applied in this process. Apart from that, removing special characters is used to avoid changes in the meaning of each word, similar needs to remove white space for too many empty tokens, and also removing words that have no meaning is used to avoid noise in the pattern of the machine learning process. After the text cleaning process is complete, the next step is breaking the text into word sets, which is known as the tokenization process. Another thing that is no less important in this stage is lemmatization, which is the process of finding the root of a word to handle duplicate words with the same meaning so that words containing word affixes can be standardized to become the main word or root word [18]. This research focused on lemmatization based on verbs.

D. *Feature Extraction*

To complete the preprocessing needs before performing the text classification process, the program should make all the text the same length. For handling that case, feature extraction can help convert text to be vector. This research uses BoW, TF-IDF [7], and Word2Vec to convert a document to a set of vectors to handle the different lengths of each document. Word2Vec has 2 models, including Continuous Bag-of-Words (CBOW) and Skip-gram. This study only focuses on using Skip-gram, which can predict similar words based on a given center word [19].

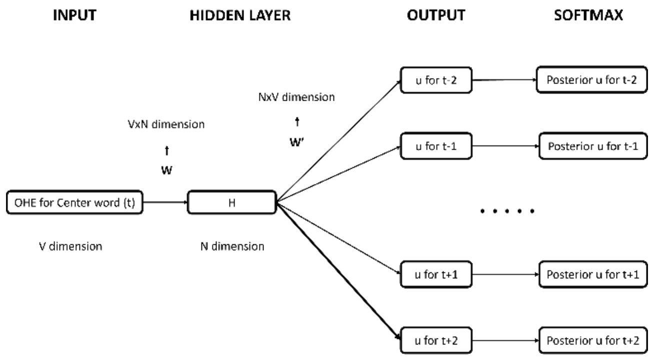

Fig. 2. Word2Vec Skip-gram Model

Based on how Skip-gram works [20] as shown in Fig. 2, all the words in a document will be converted into a dictionary first. Each word in the dictionary will be converted into a One-Hot Encoded (OHE) vector as an input value, and the output obtained is a vector matrix that represents the similarity between the words to another word in the dictionary.

E. *Classification*

This research uses some classification models, including Support Vector Machine (SVM), Decision Tree, Random Forest, Logistic Regression, Naïve Bayes, K-Nearest Neighbor (KNN), Neural Network Classifier, and Doc2Vec which combines each classification model with our three feature extractions method, except Doc2Vec, and compare them to find the best feature extraction method for each classification model. Doc2Vec has its vector generation method, which can convert text to a document vector, so the model does not need to convert all the words in the sentence to word vectors.

This research will use the SVM model to classify our malware incident and non-malware incident news data by separating classes using hyperplane (H). H1 and H2 as the dividing boundaries between classes, also known as the class interval [21].

As the decision tree classification process [22], this research used some supported criteria for the decision tree, including gini, entropy, and log loss, to measure the split quality. To minimize the overfitting, some depth limits also tried to make sure that with the pruning method, the accuracy of this method can be improved [23]. The Random Forest model contains a combination of several tree classifiers [24]. Each tree classifier will make class predictions for the data, and then all the prediction results will be used to determine the final class.

Logistic Regression is one of the classification methods used in this research to predict the class of data based on computing the probability [5]. This method is suitable for classification because it can identify outlier data.

Naive Bayes classifier works based on a probabilistic model [25]. This research uses the model to classify documents based on comparing probability values, whether all the features contained in related documents have a more excellent probability value to state that the document talks about malware incident news.

K-Nearest Neighbor (KNN) [26] determines the class in the document whether it includes malware incident news based on other data classes that are considered close to it, calculated based on the closeness of the value of each feature in it, the nearest data used as a reference for class determination is the value of K as the determinant of the amount of data. To find a better distance of computation, this research compares two types of metrics, including Minkowski and cosine.

As described by [27], the problems in categorizing unstructured data can be solved using the backpropagation method. This research also tries to compare activation functions for the hidden layer, including Identity, Logistics, Tanh, and Relu. For the weight optimization method, this research also compares some solvers, including LBFGS which is a family of quasi-newton methods, stochastic gradient descent, and adam, which refers to the stochastic gradient-based optimizer.

Based on the explanation of Doc2Vec [28], it has two models, called DBOW (Distributed Bag-of-Words) and DMPV (Distributed Memory Paragraph Vectors), but DBOW can train faster and give better results. This research uses the DBOW model in implementing Doc2Vec. This model can classify documents by comparing similarities to each other. The test data is compared with the train data, and then the train data class values that are considered similar to the test data will be used as the predicted value for the document.

F. *Evaluation*

To measure the accuracy of each method combination (feature extraction and classification), this research uses a 10-fold cross-validation method that divides the data into ten parts, nine parts of it used as training data and another part as test data. This process will take ten iterations, as mentioned by [3]. This research evaluates the result based on the accuracy and computation time.

IV. EXPERIMENT

A. *Data Collection*

In collecting data, this research uses the text mining method, specifically web crawling, to get the text of some online news and obtain 39,709 data. The web crawling process uses Python with the Request, Beautifulsoup, and Selenium Web Driver libraries. For the result detail, several collected text documents are as follows:

TABLE I. CRAWLED DATA

Source	Total News Crawled
www.infosecurity-magazine.com/	23, 578
https://thehackernews.com/search/label/Malware	1,833
https://securityaffairs.co/	3,712
https://portswigger.net/daily-swig/malware	368
https://www.trendmicro.com/en_us/research.html?category=trend-micro-research:threats/malware	100
https://www.darkreading.com/vulnerabilities-threats	510
https://www.malwarebytes.com/blog/category/news	3,070
https://threatpost.com/category/malware-2/	4,955
www.bleepingcomputer.com/	1,075
www.scmagazine.com/topic/malware	508

B. *Labeling*

The labeling process carried out in this research produces 200 training data with a 50% division for data labeled 1 (true), which means the data is malware incident news, and 50% for data labeled 0 (false), which means the data is not malware incident news. To ensure that the data labels are good enough, this research measures the dataset's quality using the "cleanlab" library from Python.

C. *Text Normalization*

Below is an example of the text normalization process carried out in this research:

"Gugum is a Master of Information Technology student studying Data Science Cyber Security. Linked in address: https://www.linkedin.com/in/gugum-gumilar"

After the text is normalized, the text will be:

['gugum', 'master', 'information', 'technology', 'student', 'study', 'data', 'science', 'cyber', 'security', 'link', 'address', 'https', 'www.linkedin.com', 'gugum-gumilar'].

On that text, all letters changed to lowercase, all the space was removed, and special characters were also deleted except the dot (.) on the link and dash (-) character, and the text will separate into a collection of tokens. It is required to keep the dot symbol because the file name needs to use it, for example, trojan.exe. Dash symbol is needed because some naming usually uses it, including for attack names, for example, Denial-of-Service (DoS) attack.

D. *Feature Extraction*

Below is an example of how the feature extraction works for two presented texts:

T1: Gugum is a student

T2: Mr. Maula, Mr. Eka, and Mr. Charles are lecturers

From those two texts, the program should make a dictionary of words first. In this case, the dictionary will contain the words including "gugum, student, mr., maula, eka, Charles, lecturer". The following table is an example of a word vector that represents the value of each word in those two sentences:

TABLE II. WORD VECTOR EXAMPLE

gugum	student	mr.	maula	eka	charles	lecturer
1	1	0	0	0	0	0
0	0	3	1	1	1	1

E. *Classification*

The classification methods used in this research are to predict labels in new data that have extracted features in the previous stage, and the prediction process refers to previously prepared learning data. The prediction results are in the form of values 1 (true) and 0 (false). Thus, the classification process for malware incident news runs automatically.

F. *Evaluation*

To produce the best result of accuracy, this research does some experiment, but not all result is written here, including the way to choose the best epoch for Word2Vec. Another thing that is not presented here, is the way to choose the number of trees for a random forest classifier. Some value was chosen, including 5,10,50,100, and 200, but 100 trees can produce the highest classification result so that value was picked for this experiment. For Doc2Vec, this research was also trying to use only the top 1, 5, and 10 similar documents, but finally, this research only used the top 10 as the best result.

TABLE III. EVALUATION OF BAG-OF-WORD VECTORIZER

Model	Accuracy	Process Time (s)
SVM	0.59	2.38389
Decision Tree	0.92	1.11884
Logistic Regression	0.91	1.93613
Naïve Bayes	0.77	0.9887
K-NN	0.89	1.1117
Neural Network	0.91	4.23132
Random Forest	0.88	1.40823

Table III shows that the better classification model to be combined with the Bag-of-Word vectorizer is the Decision Tree with Gini and the depth of tree value is 50. The second one is Logistic Regression, which produces similar accuracy, but the computation time is longer than Decision Tree. Neural Network classifier with Relu as an activation function for hidden layer and LBFGS weight optimizer also produces good enough accuracy, although this method needs more computation time. K-NN with cosine produces better accuracy than Minkowski, but the accuracy still only meets 89%.

TABLE IV. EVALUATION OF TF-IDF VECTORIZER

Model	Accuracy	Process Time (s)
SVM	0.9	1.66237
Decision Tree	0.9	4.93279
Logistic Regression	0.9	1.08156
Naïve Bayes	0.77	0.95867
K-NN	0.82	1.11012
Neural Network	0.93	3.80524
Random Forest	0.87	1.4414

Table IV shows that the Neural Network classifier with the identity as the activation function for the hidden layer and LBFGS for weight optimizer can produce the best accuracy when combined with TF-IDF. A decision tree using entropy (to measure the quality of a split), and also using a depth limit value of 100 can produce good quality, but still under a neural network. K-NN using Minkowski and cosine can have the same accuracy. SVM and Logistic Regression have good quality when combined with TF-IDF.

TABLE V. EVALUATION OF WORD2VEC SKIP-GRAM VECTORIZER WITH 100 EPOCHS

Model	Accuracy	Process Time (s)
SVM	0.55	9.94241
Decision Tree	0.92	24.7563
Logistic Regression	0.9	9.35198
Naïve Bayes	0.83	8.67811
K-NN	0.8	25.61673
Neural Network	0.93	30.09731
Random Forest	0.88	9.10074

Table V shows the feature extraction model using Word2Vec. The Neural Network classifier can produce the best accuracy, which meets 93% using Tanh as an activation function and LBFGS for weight optimizer. The second one is the Decision Tree classifier with Gini to measure split quality, and using unlimited depth of tree can produce accuracy which meets 92%.

TABLE VI. DOC2VEC

Epochs	Accuracy	Process Time (s)
10	0.87	7.43199
50	0.84	37.42535
100	0.79	66.87108
500	0.82	287.2701
1000	0.78	577.98967

As described in Chapter III, Doc2Vec has a default method to vectorize documents, so the classification process does not need to vectorize all words on each document first. Based on Table V, this research has tried some experiments with this model, especially using some epochs to find the best accuracy. In this research, Doc2Vec produces the best value when using 10 epochs, but the highest accuracy is only 87%, and the processing time is also higher than the other methods.

V. CONCLUSION

The classification model that is suited to use a Bag-of-Word vectorizer is a Decision Tree using Gini, and the depth of the tree is only 50, it can produce an accuracy of 92%. The second one is Logistic Regression which can produce accuracy until 91%, and the last is Neural Network classifier, which can produce similar accuracy to Logistic Regression but needs longer computation time. This research also found that the Neural Network classifier is good enough to be combined with TF-IDF because it can produce the best accuracy, which can meet 93%. For classification model that is suited to use Word2Vec is Neural Network, which can have an accuracy of 93%, and Decision Tree can produce an accuracy of 92%. For Doc2Vec with 10 epochs, it can produce the highest accuracy than using other values of epochs but only meets 87% of accuracy.

VI. DISCUSSION AND FUTURE WORK

From the results, it shown that there are methods that tend to be faster processing time, but the accuracy is not better than methods that take longer processing time. Therefore, we need to adjust again to our needs, whether accuracy is the highest priority or data processing speed is considered more important. Further work includes text summarization, to generate conclusions from all documents that contain the same topic.

ACKNOWLEDGMENT

We are very grateful to Mr. Muhammad Ghifary, who guided the author to achieve the best results. We also acknowledge the support received from PT Xynexis International in the process of publishing this research.

REFERENCES

[1] T. B. Shahi and A. K. Pant, "Nepali news classification using Naïve Bayes, Support Vector Machines and Neural Networks," *Proc. - 2018 Int. Conf. Commun. Inf. Comput. Technol. ICCICT 2018*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/ICCICT.2018.8325883.

[2] G. L. Yovellia Londo, D. H. Kartawijaya, H. T. Ivariyani, P. W. P. Yohanes Sigit, A. P. Muhammad Rafi, and D. Ariyandi, "A Study of Text Classification for Indonesian News Article," *Proceeding - 2019 Int. Conf. Artif. Intell. Inf. Technol. ICAIIT 2019*, pp. 205–208, 2019, doi: 10.1109/ICAIIT.2019.8834611.

[3] H. S. Al-Ash, M. F. Putri, P. Mursanto, and A. Bustamam, "Ensemble Learning Approach on Indonesian Fake News Classification," *ICICOS 2019 - 3rd Int. Conf. Informatics Comput. Sci. Accel. Informatics Comput. Res. Smarter Soc. Era Ind. 4.0, Proc.*, pp. 2–7, 2019, doi: 10.1109/ICICoS48119.2019.8982409.

[4] M. G. Hussain, M. Rashidul Hasan, M. Rahman, J. Protim, and S. Al Hasan, "Detection of Bangla Fake News using MNB and SVM Classifier," *Proc. - 2020 Int. Conf. Comput. Electron. Commun. Eng. iCCECE 2020*, pp. 81–85, 2020, doi:

10.1109/iCCECE49321.2020.9231167.

[5] K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification,” *Augment. Hum. Res.*, vol. 5, no. 1, 2020, doi: 10.1007/s41133-020-00032-0.

[6] O. Mendsaikhan, H. Hasegawa, Y. Yamaguchi, and H. Shimada, “Identification of cybersecurity specific content using the Doc2Vec language model,” *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 1, pp. 396–401, 2019, doi: 10.1109/COMPSAC.2019.00064.

[7] A. Al Hamoud, A. Alwehaibi, K. Roy, and M. Bikdash, *Classifying political tweets using naïve bayes and support vector machines*, vol. 10868 LNAI. Springer International Publishing, 2018. doi: 10.1007/978-3-319-92058-0_71.

[8] S. Garg, “Drug recommendation system based on sentiment analysis of drug reviews using machine learning,” *Proc. Conflu. 2021 11th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 175–181, 2021, doi: 10.1109/Confluence51648.2021.9377188.

[9] G. M. Barrientos, R. Alaiz-Rodríguez, V. González-Castro, and A. C. Parnell, “Machine learning techniques for the detection of inappropriate erotic content in text,” *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 591–603, 2020, doi: 10.2991/ijcis.d.200519.003.

[10]) H Oyeon Park, K.-J. Kim, P. D. Candidate, • First, and H. Park, “Impact of Word Embedding Methods on Performance of Sentiment Analysis with Machine Learning Techniques,” *J. Korea Soc. Comput. Inf.*, vol. 25, no. 8, pp. 181–188, 2020.

[11] N. Chayangkoon and A. Srivihok, “Text classification model for methamphetamine-related tweets in Southeast Asia using dual data preprocessing techniques,” *Int. J. Electr. Comput. Eng.*, vol. 11, no. 4, pp. 3617–3628, 2021, doi: 10.11591/ijece.v11i4.pp3617-3628.

[12] F. Tempola, R. Rosihan, and R. Adawiyah, “Holdout Validation for Comparison Classification Naïve Bayes and KNN of Recipient Kartu Indonesia Pintar,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1125, no. 1, p. 012041, 2021, doi: 10.1088/1757-899x/1125/1/012041.

[13] K. Pal and B. V. Patel, “Data Classification with K-fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques,” *Proc. 4th Int. Conf. Comput. Methodol. Commun. ICCMC 2020*, no. Iccmc, pp. 83–87, 2020, doi: 10.1109/ICCMC48092.2020.ICCMC-00016.

[14] A. Sharma, I. Singh, and V. Rai, “Fake News Detection on Social Media using K-Nearest Neighbor Classifier,” *2022 2nd Int. Conf. Adv. Comput. Innov. Technol. Eng. ICACITE 2022*, pp. 803–807, 2022, doi: 10.1109/ICACITE53722.2022.9823660.

[15] D. M. Thomas and S. Mathur, “Data Analysis by Web Scraping using Python,” *Proc. 3rd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2019*, pp. 450–454, 2019, doi: 10.1109/ICECA.2019.8822022.

[16] H. Jo, J. Kim, P. Porras, V. Yegneswaran, and S. Shin, “GapFinder: Finding Inconsistency of Security Information from Unstructured Text,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, no. c, pp. 86–99, 2021, doi: 10.1109/TIFS.2020.3003570.

[17] C. G. Northcutt, L. Jiang, and I. L. Chuang, “Confident learning: Estimating uncertainty in dataset labels,” *J. Artif. Intell. Res.*, vol. 70, pp. 1373–1411, 2021, doi: 10.1613/JAIR.1.12125.

[18] L. Ignaczak, G. Goldschmidt, C. A. Da Costa, and R. D. R. Righi, “Text Mining in Cybersecurity: A Systematic Literature Review,” *ACM Comput. Surv.*, vol. 54, no. 7, 2022, doi: 10.1145/3462477.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[20] X. Rong, “word2vec Parameter Learning Explained,” pp. 1–21, 2014, [Online]. Available: http://arxiv.org/abs/1411.2738

[21] Y. Zhang, “Support vector machine classification algorithm and its application,” *Commun. Comput. Inf. Sci.*, vol. 308 CCIS, no. PART 2, pp. 179–186, 2012, doi: 10.1007/978-3-642-34041-3_27.

[22] B. Charbuty and A. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning,” *J. Appl. Sci. Technol. Trends*, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[23] C. Schaffer, “Overfitting Avoidance as Bias,” *Mach. Learn.*, vol. 10, no. 2, pp. 153–178, 1993, doi: 10.1023/A:1022653209073.

[24] Y. Liu, Y. Wang, and J. Zhang, “New machine learning algorithm: Random forest,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7473 LNCS, pp. 246–252, 2012, doi: 10.1007/978-3-642-34062-8_32.

[25] V. Narayanan, I. Arora, and A. Bhatia, “Fast and accurate sentiment classification using an enhanced Naive Bayes model,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8206 LNCS, pp. 194–201, 2013, doi: 10.1007/978-3-642-41278-3_24.

[26] O. Kramer, “Dimensionality Reduction with Unsupervised Nearest Neighbors,” *Intell. Syst. Ref. Libr.*, vol. 51, pp. 13–23, 2013, doi: 10.1007/978-3-642-38652-7.

[27] F. Harrag and E. El-Qawasmah, “Neural network for Arabic text classification,” *2nd Int. Conf. Appl. Digit. Inf. Web Technol. ICADIWT 2009*, pp. 778–783, 2009, doi: 10.1109/ICADIWT.2009.5273841.

[28] J. H. Lau and T. Baldwin, “An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation,” pp. 78–86, 2016, doi: 10.18653/v1/w16-1609.