International Information and Engineering Technology Association
Advancing the World of Information and Engineering

International Journal of Safety and Security Engineering
Vol. 13, No. 5, October, 2023, pp. 835-844
Journal homepage: http://iieta.org/journals/ijsse

Classifying Indonesian Cyber Crime Cases under ITE Law Using a Hybrid of Mutual Information and Support Vector Machine

Romi Fadillah Rahmat[1*], Aina Hubby Aziira[2], Sarah Purnamawati[1], Yunita Marito Pane[1], Sharfina Faza[3], Al-Khowarizmi[4], Farhad Nadi[5]

[1] Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan 20155, Indonesia
[2] Faculty of Information Technology, Universitas Andalas, Padang 25166, Indonesia
[3] Computer Engineering and Informatics Department, Politeknik Negeri Medan, Medan 20155, Indonesia
[4] Department of Information Technology, Universitas Muhammadiyah Sumatera Utara, Medan 20238, Indonesia
[5] School of Information Technology, UNITAR International University, Petaling Jaya 47301, Malaysia

Corresponding Author Email: romi.fadillah@usu.ac.id

https://doi.org/10.18280/ijsse.130507

Received: 15 May 2023
Revised: 29 September 2023
Accepted: 13 October 2023
Available online: 10 November 2023

Keywords:
mutual information, cyber law, text processing, TF-IDF

ABSTRACT

In Indonesia, the process of identifying and categorizing cyberlaw infringements traditionally involves manual procedures administered by experts, lawyers, or law enforcement personnel. This study introduces a method to enhance the analysis and processing of case chronological data through the application of text mining. Using the Support Vector Machine for classification, alongside feature extraction both with and without Mutual Information, the study aims to automate the classification of cybercrime cases. The preprocessing phase encompasses text cleaning, case folding, stop word removal, stemming, and tokenization and weighting with TF-IDF. The model achieved an accuracy rate of 95.45% during evaluation and 91.42% when tested on 35 data points with 1500 selected features. This performance surpasses the classification accuracy obtained in previous research.

1. INTRODUCTION

The advent of information technology has undeniably brought various advantages to society. Yet, this development is not without its drawbacks as it has inadvertently given rise to unethical behavior, known as cybercrime [1, 2]. Cybercrime, defined as the creation, distribution, theft, misuse, and destruction of data through computer software manipulation, has become a pervasive issue in the digital era [3-7].

In response to this growing threat, the Indonesian government instituted Law No. 11 of 2008 on Electronic Information and Transactions, further revised in Law No. 19 of 2016 pertaining to Information and Electronic Transactions [8-12] (hereinafter referred to as the ITE Law). At present, the identification of ITE Law violations is primarily a manual process, requiring the examination of event chronology and direct inquiry, often assisted by expert witnesses [13]. However, the reliance on expert witnesses for the detection of ITE Law violations in cybercrime cases presents significant challenges due to limited accessibility and high cost [11]. Consequently, there is a pressing need for a system that can facilitate the identification of ITE Law violations in cybercrime cases [14].

The ITE Law is designed with the intention of regulating information and electronic transactions, and mitigating the adverse effects of technological advancements, notably cybercrimes. These crimes often exploit technological progress in Information Technology and Electronics (ITE), threatening the legal interests of individuals, communities, and the state [15].

In the realm of academic research, prior studies have addressed the application of text mining to the ITE Law. For instance, a study conducted by Saputra et al. [16] employed the Latent Semantic Indexing (LSI) method to locate articles concerning the ITE Law in cybercrime cases. This approach yielded an accuracy result of 83.33% for recall, 50% for precision, and 62.5% for f-measure. Another research endeavor by Hakim et al. [17] explored the clustering of cybercrime cases under the ITE Law, utilizing the k-means algorithm to classify the cases into five distinct clusters.

Numerous studies have employed machine learning methods for classification tasks. For instance, a study by Yıldırım et al. [18] aimed at classifying breaking news on financial portals was conducted. Data for this research were collected from the Dow Jones news portal spanning from 2013 to 2017, comprising 10,000 headline news items and 10,000 non-headline news items. An accuracy of 95.5% was achieved in this study. Further research by Zheng et al. [19] involved the identification of email authors using Support Vector Machine (SVM) and the Analytic Hierarchy Process (AHP). The researchers applied TF-IDF for weighting and AHP to determine the weights of the elements, while SVM was used for classification. The study reported an accuracy of 95%. Meanwhile, Celine et al. focused on classifying Turkish language texts [20]. This research utilized doc2vec to create the model and TF-IDF for word weighting. The classification process involved several methods, including SVM, KNN, CC, and CFSVM. The combination of SVM and TF-IDF, with a split of 99% training data and 1% testing data, yielded the highest accuracy of 94.17%. In another study, Shill and Paul

[21] conducted sentiment analysis on product reviews for watches in 2016. The researchers used mutual information for feature selection and the multinomial naive Bayes algorithm for data classification, achieving an accuracy of 88.54%.

This study departs from previous research in several key respects. For example, the research conducted by Saputra et al. [16] employed the Latent Semantic Indexing (LSI) method to determine articles related to the ITE Law. In contrast, the current investigation uses mutual information, TF-IDF, and the SVM algorithm to classify cybercrime cases, presenting a different approach. Another study by Hakim et al. [17] used text mining to cluster cases linked to the ITE Law, with data sourced from Twitter posts. They applied TF-IDF for this purpose. However, the present study diverges by focusing on the classification of ITE Law types using SVM and mutual information, even though both studies make use of TF-DF. The research by Zheng et al. [19] implemented the Support Vector Machine (SVM) and Analytic Hierarchy Process (AHP) to identify spam emails, assessing representative features such as word frequency, syntax structure, word length, word format, and punctuation. In comparison, this study employs TF-IDF for weighting and AHP for weighing elements. The novelty of this study lies in its feature selection process, which reduces words when weighted, representing a departure from the aforementioned research. Further, the research conducted by Çelenli et al. [20] utilized doc2vec to create a model for classifying Turkish documents, with TF-IDF being used for word weighting. However, the present study distinguishes itself by applying mutual information for feature selection. Finally, the study by Shill and Paul [21] made use of mutual information for feature selection and the Multinomial Naive Bayes algorithm for data classification. Conversely, the current study employs TF-IDF for feature extraction and the SVM algorithm for classification, with both studies using mutual information for feature selection.

Further examination of previous studies reveals additional areas of distinction. A study on Indonesian News Classification using Latent Dirichlet Allocation yielded an average accuracy of 70% across all classes [22]. In another study, Romsaiyud et al. [23] automated the detection of cyberbullying through a pattern of clustering appearance, while Alami employed Text Mining to detect and predict criminal activity in microblog postings using Latent Dirichlet Allocation [24]. Yet another application of text processing using SVM is demonstrated by Suliani et al. [25]. These findings suggest that SVM can be applied to a variety of problems [26], indicating a substantial research gap in the field of automatic news classification. This study seeks to address two primary research questions: (1) the identification of a more accurate methodology for classifying text documents based on the semantic words within their sentences, and (2) the creation of a system that can provide the public with an accessible recommendation system and enhance their interpretation of the ITE Laws in Indonesia.

The current study contributes to the field by implementing an automatic system capable of classifying documents, texts, or reports regarding ITE Law violations with high accuracy. The proposed method, which could potentially be used on public websites, may significantly impact the Indonesian population's understanding of specific actions and their consequences from the perspective of ITE Laws. This research advances our previous study [13] by enhancing the SVM method with Mutual Information hybridization to achieve more accurate results. Notably, no similar application currently exists in Indonesia, either as a prototype or a system, with the same purpose as this study.

This research proposes a multi-step method for classifying types of ITE laws in cybercrime cases. The following processes will be undertaken: Initially, chronological documents of cybercrime cases will be inputted. Subsequent to this data entry, a text pre-processing procedure will be conducted. This pre-processing encompasses cleaning, case folding, stop word removal, stemming, and tokenization. Following this pre-processing stage, feature selection will be performed using mutual information. Subsequently, the TF-IDF method will be applied for the feature extraction process. The ensuing step entails the classification of cybercrime case chronology documents, which will be executed using the multi-class Support Vector Machine with mutual information algorithm. The aim is to classify news events into five categories based on the topics inherent in each document.

2. RESEARCH METHOD

In order to classify the law violation of cybercrime cases, the authors proposed a study consisting of several processes. These processes are as follows: data acquisition which was divided into two categories of training data and test data. After data acquisition, it would go through pre-processing. The process consists of data cleaning, case folding, stop word removal, stemming, and tokenization. After the pre-processing process, feature selection was carried out using mutual information while the feature extraction process was conducted using the TF-IDF method. The final step would be the classification of cybercrime cases. The classification would be performed using the Support Vector Machine algorithm. The general architecture of this study can be seen in Figure 1.

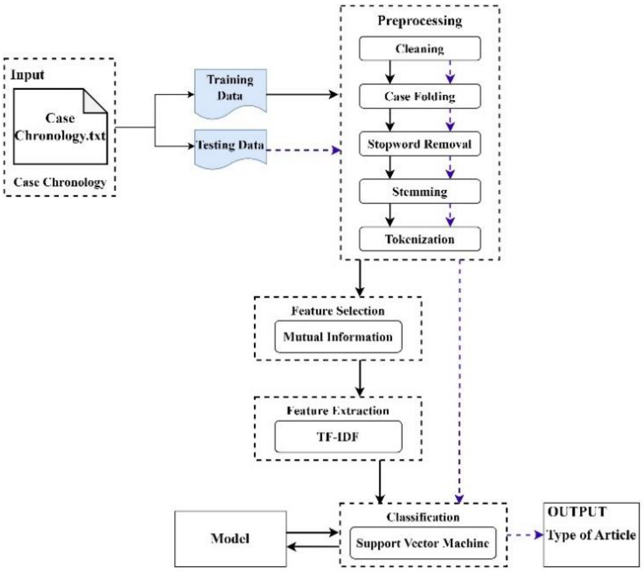


Figure 1. General architecture

2.1 ITE Law Article 27-30

The ITE Law contains nine articles related to cybercrimes: from article 27 to article 35, with 20 forms or types of ITE crimes. Furthermore, criminal threats from ITE cases are regulated in articles 45 to 52. The contents of the articles on ITE crimes are as follows:

Article 27:

(1) Any person intentionally and without rights distributes and/or transmits and/or causes an electronic information and/or electronic document with contents against decency to be accessible.

(2) Any person intentionally and without rights distribute and/or transmits and/or causes an electronic information and/or electronic document with gambling contents to be accessible.

(3) Any person intentionally and without right distributes and/or transmits and/or causes an electronic information and/or electronic document with defamation and/or slander contents to be accessible.

(4) Any person who intentionally and without rights distributes and/or transmits and/or causes an electronic information and/or electronic document with extortion and/or threat contents to be accessible.

Article 28:

(1) Any person intentionally and without rights spreads false and misleading news that results in consumer losses in Electronic Transactions.

(2) Any person intentionally and without rights disseminates information aimed at causing feelings of hatred or hostility to certain individuals and/or groups of people based on ethnicity, religion, race, and inter-group (SARA).

Article 29:

Any person intentionally and without rights sends electronic information and/or electronic documents that contain threats of violence or intimidation aimed at personally.

Article 30:

(1) Any Person intentionally and without rights or against the law accesses computers and/or electronic systems belonging to other persons in any way.

(2) Any Person intentionally and without rights or against the law accesses a computer and/or electronic system in any way with the purpose of obtaining electronic information and/or electronic documents.

(3) Any person intentionally and without rights or against the law accesses a computer and/or electronic system in any way by violating or breaking through the security system.

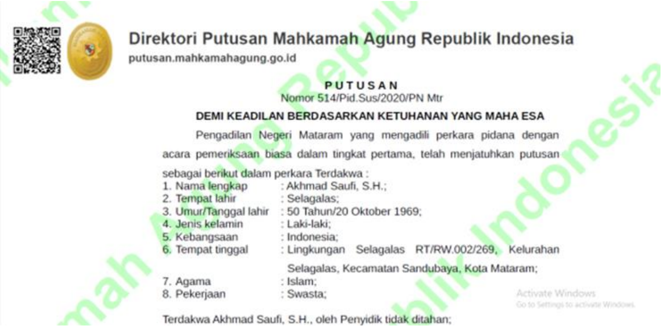
2.2 Dataset

The dataset used in this study is a chronology of cybercrime cases obtained from cybercrime case decision documents of case verdict documents downloaded manually from https://ujungan3.mahkamahagung.go.id/. We gather up around 500 documents then we check every document and classify it to several law violation categories such as article 27 paragraph 1, article 27 paragraph 3, article 27 paragraph 4, article 28 paragraph 1, article 28 paragraph 2, and other documents. One think to be considered in this process, we need to exactly find the real decision of this case, weather the verdict goes to guilty or not guilty. If it is not guilty then we are not considered it as violation and we are not input it in our dataset. For the sample of the raw data that has been collected can be seen in Figure 2.

From the case verdict document, we took the chronology of the case in the document that will be used as input in this study and it is manually labeled to our system shown in Figure 3. In this study, the input consists of two parts, i.e., training data and testing data. Training data is the data that has been defined based on the article on ITE Law. Testing data is the data that will be tested to be classified based on the related article. This sorting process then resulted in 255 file which categorized as a good data to be processed. It is shown in Table 1, which consist of 220 data to be used in training process and 35 data to be used in testing process.

Table 1. Training and testing data

No	Category	Training Data	Testing Data
1	Article 27 paragraph 1	30	5
2	Article 27 paragraph 3	70	10
3	Article 27 paragraph 4	30	5
4	Article 28 paragraph 1	30	5
5	Article 28 paragraph 2	30	5
6	Other documents	30	5
	Total	220	35



Direktori Putusan Mahkamah Agung Republik Indonesia
putusan.mahkamahagung.go.id

PUTUSAN
Nomor 514/Pid.Sus/2020/PN Mtr

DEMI KEADILAN BERDASARKAN KETUHANAN YANG MAHA ESA

Pengadilan Negeri Mataram yang mengadili perkara pidana dengan acara pemeriksaan biasa dalam tingkat pertama, telah menjatuhkan putusan sebagai berikut dalam perkara Terdakwa :

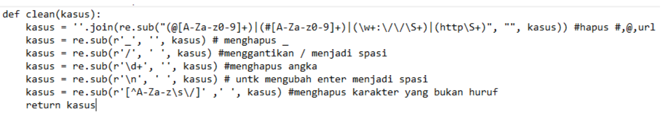
1. Nama lengkap	: Akhmad Saufi, S.H.;
2. Tempat lahir	: Selagalas;
3. Umur/Tanggal lahir	: 50 Tahun/20 Oktober 1969;
4. Jenis kelamin	: Laki-laki;
5. Kebangsaan	: Indonesia;
6. Tempat tinggal	: Lingkungan Selagalas RT/RW.002/269, Kelurahan Selagalas, Kecamatan Sandubaya, Kota Mataram;
7. Agama	: Islam;
8. Pekerjaan	: Swasta;

Terdakwa Akhmad Saufi, S.H., oleh Penyidik tidak ditahan;

Figure 2. Example of case verdict document

2.3 Data pre-processing

The preprocessing stage was conducted to alter the text to be more structured. This preprocessing stage was divided into several stages: Cleaning data to remove or clean sentences from elements that are not needed to reduce noise in the data such as HTML characters, retweets, usernames, hashtags, URLs, symbols, punctuation marks, and number, shown in Table 2.



```
def clean(kasus):
    kasus = ''.join(re.sub("(@[A-Za-z0-9]+)|(#[A-Za-z0-9]+)|(\w+:\/\/\S+)|(http\S+)", "", kasus)) #hapus #,@,url
    kasus = re.sub(r'_', '', kasus) # menghapus _
    kasus = re.sub(r'/', ' ', kasus) #menggantikan / menjadi spasi
    kasus = re.sub(r'\d+', '', kasus) #menghapus angka
    kasus = re.sub(r'\n', ' ', kasus) # untk mengubah enter menjadi spasi
    kasus = re.sub(r'[^A-Za-z\s\/]' ,' ', kasus) #menghapus karakter yang bukan huruf
    return kasus
```

Figure 3. Case decision document

Table 2. Cleaning process

Before Case Folding Process	After Case Folding Process
Pada desember Nug mengirim pesan broadcast pada media sosial Blackberry Messenger BBM nug menyebarkan foto bugil dari Yul ke semua contact BBMnya Nug menyebarkan materi pornografi ke orang orang dalam daftar contact BBM nya Nug melakukan penyebaran foto bugil Yul kepada orang orang yang ada di dalam daftar contact BBM nya	Pada desember Nug mengirim pesan broadcast pada media sosial Blackberry Messenger BBM nug menyebarkan foto bugil dari Yul ke semua contact BBMnya Nug menyebarkan materi pornografi ke orang orang dalam daftar contact BBM nya Nug melakukan penyebaran foto bugil Yul kepada orang orang yang ada di dalam daftar contact BBM nya

Case folding is the process of converting all text in a document to the same characters, both upper and lower case, to speed up the comparisons during the indexing process. At this stage, characters other than the alphabet will be removed and considered delimiters. The purpose of case folding is to make the characters to be uniform to speed up the indexing process. At this stage the system converts into all lowercase characters. The results of the case folding process can be seen in the Table 3.

Table 3. Case folding process

After Case Folding Process	After Cleaning Process
Pada 18 desemebr Nug mengirim pesan broadcast pada media sosial Blackberry Messenger BBM nug menyebarkan foto bugil dari Yul ke semua contact BBMnya Nug menyebarkan materi pornografi ke orang orang dalam daftar contact BBMnya Nug melakukan penyebaran foto bugil Yul kepada orang orang yang ada di dalam daftar contact BBM nya	Pada desember Nug mengirim pesan broadcast pada media sosial Blackberry Messenger BBM nug menyebarkan foto bugil dari Yul ke semua contact BBMnya Nug menyebarkan materi pornografi ke orang orang dalam daftar contact BBM nya Nug melakukan penyebaran foto bugil Yul kepada orang orang yang ada di dalam daftar contact BBM nya

Stop-word removal is the process of removing or eliminating stop words. Stop words are words that often appear in large numbers and are considered meaningless in text documents. Examples of stop words are "which", "and", "will" and others.

In this study, apart from using stop-words from the Python library, we also added a list of stop-words which included defendant, Twitter, account, status, uploading, posting, BBM, Facebook, witness, fb, Blackberry, Messenger, December, January, February, March, April, May, June, July, August, September, October, November, media and social. The results of the stop-word removal process can be seen in the Table 4.

Table 4. Stop-word removal process

After Cleaning Process	After Stopword Removal Process
pada desember nug mengirim pesan broadcast pada media sosial blackberry messenger bbm nug menyebarkan foto bugil dari yul ke semua contact bbm nya nug menyebarkan materi pornografi ke orang orang dalam daftar contact bbm nya nug melakukan penyebaran foto bugil yul kepada orang orang yang ada di dalam daftar contact bbm nya	nug mengirim pesan broadcast nug menyebabkan foto bugil yul contact bbmnya nug menyebarkan materi pornografi orang orang daftar contact nya nug penyebaran foto bugil yul orang orang di dalam daftar contact nya

Table 5. Stemming process

After Stopword Removal Process	After Stemming Process
nug mengirim pesan broadcast nug menyebarkan foto bugil yul contact bbmnya nug menyebarkan materi pornografi orang orang daftar contact nya nug penyebaran foto bugil yul orang orang di dalam daftar contact nya	nug kirim pesan broadcast nug sebar foto bugil yul bbmnya nug sebar materi pornografi orang orang daftar contact nya nug sebar foto bugil yul dalam daftar contact nya

Stemming is the stage of changing a preposition into a basic word by removing the prefix and suffix in the word. The purpose of stemming is to group words derived from common stem data and base words.

In the process of this research, the author uses the stemming process to change affixed words into root words. The python library used in this study uses literature. The results of the stemming process can be seen in the Table 5.

Tokenization is the stage of parsing text in a paragraph, sentence, or page into pieces called tokens for later analysis. The purpose of tokenization is for the words in a paragraph, sentence or page to be converted into word units.

In this study, the dataset is broken down into words called tokens. The results of the tokenization process can be seen in the Table 6.

Table 6. Tokenization process

After Stemming Process	After Tokenization Process
nug kirim pesan broadcast nug foto bugil yul contact bbmnya nug sebar materi pornografi orang orang daftar contact nya nug sebar foto bugil yul dalam daftar contact nya	[nug, kirim, pesan, broadcast, nug, sebar, foto, bugil, yul, contact, bbmnya, nug, sebar, materi, pornografi, orang orang, daftar, contact, nya, nug, sebar, foto, bugil, yul, dalam, daftar, contact, nya]

2.4 Mutual information feature selection

After the preprocessing process is complete, the next process would be featuring selection. Feature selection is the stage of selecting terms or features that exist in the dataset to take relevant features to each class, while irrelevant features will be discarded. This process aims to reduce existing terms to be shorter. The mutual information (MI) method was used as the feature selection process in this study.

Mutual information (MI) is a feature selection method in a document. MI is a method that calculates the correlation between terms from one document to another document and sees the contribution of a term to make decisions on the classification process in a class [12]. Features are selected based on the MI values between terms and classes. The MI value can be obtained by calculating the frequency of term X in class A, the frequency of terms other than X in class A, the frequency of term X not in class A, the frequency of terms other than X not in class A and the sum of all existing terms. After the frequency value is obtained, the next step is to calculate the MI value of each term. The MI calculation can be seen from Eq. (1) below [27]:

$$I(K;C) = \sum_{e_{t\{10\}}} \sum_{e_{c\{10\}}} P(K = e_t C e_c) log \frac{P(K=e_t C=e_c)}{P(K=e_t)P(C=e_c)} \tag{1}$$

K is the term value and *C* is the class. *K* is a random variable with a value of e_t=1 and e_t=0. The value of e_t=1 is the frequency of the term X while e_t=0 is the frequency of the term other than X. *C* is a random variable with the value of 1 and 0. The value of e_c=1 is the feature frequency that is in class *C* while the value of e_c=0 is the feature frequency that is not in class *C*. If Eq. (2) is described, the results of the MI value calculation can be obtained through Eq. (2) as follows [28].

$$\begin{aligned} I(KC) = \frac{N_{11}}{N} log \frac{N.N_{11}}{N_1.N_1} + \frac{N_{01}}{N} + log \frac{N.N_{11}}{N_0.N_1} + \frac{N_{10}}{N} \\ + log \frac{N.N_{11}}{N_1.N_0} + log \frac{N.N_{00}}{N_0.N_0} \end{aligned} \tag{2}$$

Based on Eq. (2), N denotes the number of existing terms. Nuc is the number of documents that have e_t and e_c values. For example, N_{10} is a sentence containing term k (e_t=1) but not in class c (e_c=0). N_1=N_{11}+N_{10} is the number of documents containing term k (e_t=1) and the number of documents that are not in class c (e_c {10}). N=N_{00}+N_{01}+N_{10}+N_{11} is the number of documents. After calculating the MI value, the MI results are compared from one class to another and the scattered MI values will be stored. Then the features are sorted from the highest to the smallest value.

MI is a method that calculates the correlation between terms from one document to another document and looks at the contribution of a term to making decisions in the classification process of a class. The following is an example of using MI.

Sentences	Category	Length of Words
Kasus cemar nama baik laku oleh siswa lalu jaring sosial. Status devi komentar oleh banyak hujat rupa kata kata tidak baik.	27 paragraph 3	21
Kasus sara laku oleh florance lalu posting status path kata warga jogja miskin tolol tidak budaya teman teman jakarta bandung jangan mau tinggal jogja	28 paragraph 2	24

Based on the example above we do feature selection of "status" word in the category of 27 paragraph 3 which resulted below:

	$e_{c=27\ para\ 3}$ =1	$e_{c=27\ para\ 3}$ =0
$e_{t=status}$=1	1	1
$e_{t=status}$=0	20	23

Then based on the truth table above we can calculate MI value described as follows:

$$I(U,C) = \frac{1}{45}\log\frac{45.1}{(1+1)*(1+20)} + \frac{20}{45} + \log\frac{45*20}{(20+23)*(20+1)} + \frac{1}{45}\log\frac{45*1}{(1+1)*(1+23)} + \frac{23}{45} + \log\frac{45*23}{(23+1)*(23+20)} = 0.2053$$

After the MI calculation is carried out for all words and classes, the resulting MI values are compared for one class to another and the MI values spread out will be saved. Then the features are sorted from highest to smallest value.

2.5 Feature extraction

The next process would be feature extraction. Feature extraction is the process of converting words into numbers. Feature extraction was done by giving weights to the features that have been selected in the previous process. Term Frequency–Inverse Document Frequency (TF-IDF) is a feature weighting method that is most widely used to assign value to features in text processing. This method was used to calculate the feature weights in the document [13]. The weights in the TF-IDF calculation were used to evaluate words in a set of documents or corpus. The term frequency (TF) and inverse document frequency (IDF) were calculated to obtaining the weight value from TF-IDF.

Term Frequency (TF) measures the frequency of a term appearing in a document. Each document has a different term length in a document allowing a word to have a different frequency. The formula (3) was used to get the term frequency [29]:

$$TF\ (kd) = \frac{f(k)}{\max f\ (d)} \tag{3}$$

From the term frequency results in the TF calculation process, the Inverse Document Frequency (IDF) calculation would be carried out. IDF served to reduce the weight of a term if it was considered that the term is spread throughout the document, making it easier to find unique terms. IDF can be obtained through the following calculations:

$$IDF\ (k) = \log\frac{N}{DF} \tag{4}$$

The value of a term by using TF-IDF can be obtained by performing the following calculations [30]:

$$TF - IDF\ (kd) = TF\ (kd) * IDF\ (k) \tag{5}$$

where, *f(k)* is Word Frequency, *f(d)* is Frequency in documents, *TF- IDF (kd)* is the weight of a word in the whole document, k is word, d is Document, *TF (kd)* is the occurrence of frequency of a word k in a document d, *IDF (k)* is the inverse *DF* of k -word, *N* is the total number of documents, and *IDF (k)* is the number of documents containing the word. Based on the above formula, regardless of the value of *TF (kd)* if the value of *N=DF (k)* then the result will be 0 for the *IDF* calculation. For that, a value of 1 can be added on the *IDF* side.

Feature extraction is the process of changing words into numbers. Feature extraction is carried out by giving weight to the features that have been selected in the previous process. The weighting in this research uses Term Frequency-Inverse Document Frequency (TF-IDF). The values obtained from TF-IDF weighting are stored in matrix form where each row of the matrix is data while each column is a feature. This matrix will be filled with the weight value of each feature for the document or multiplying the TF value with the IDF value. The weight that has been obtained will be input to the system. The following is an example of a TF-IDF calculation which can be seen in the Table 7 with the N value or the number of documents being 50.

Table 7. Feature selection calculation

Term	TF	DF	IDF ($\log\frac{N}{DF}$)	TF-IDF($TF * IDF$)
foto	10	12	0.6198	6.1980
video	3	7	0.8539	2.5617
islam	3	1	1.6989	5.0967
rp	1	4	1.0969	1.0969
hina	1	1	1.6989	1.6989
tipu	2	6	0.9208	1.8416
uang	1	5	1.0000	1.0000
arisan	5	4	1.0969	5.4845
sexsi	1	3	1.2218	1.2218

2.6 Multi-class Support Vector Machine

Initially introduced by Vapnik, SVM can only classify data into two classes. Further researchers developed SVM so that it could classify data from more than two classes, and can solve nonlinear with high dimensional classification problems [31]. In implementing Multiclass SVM, there are several approaches to achieve it such as combining several binary

SVMs or combining all the data contained in several classes into an optimization form. As for the latter approach, the optimization to be solved is much more complicated. In solving multiclass problems, there are two common methods, namely the "one-against-all" method and the "one-against-one" method. If we considered our previous research using similar SVM [13], we found that using Multi-class SVM is better than using regular SVM. Even if we compare with Latent Semantic Indexing [16] which show a worse performance.

The one-against-all method is the simplest method in solving multiclass problems. This method solves the binary problem Ly to classify class j against all other classes with both positive class and negative samples. This method builds *i* amount of binary SVM models where k is the sum of all existing classes. The xth SVM model is trained with all class samples of positive-labeled x and the remaining negative-labeled x.

The one-against-one method builds a model using binary classification training ($K(K$-1)/2) where k is the number of classes. Each model receives each pair of classes from the training and must learn to distinguish the two classes given. At the time of testing, the classification ($K(K$-1)/2) was applied to the sample that was not visible and the class with a "+1" prediction for the highest number would be predicted using the combined classification.

The SVM binary classifier that was produced has been trained to determine whether a given class belongs to the first group or the second group of classes. The aforementioned procedure is iteratively performed on the second group, which has a number of classes beyond two, until each group consists of just one class. The procedure must be halted at that point. By using this approach, the multiclass SVM is converted into multiple SVM binary classifiers. Every Support Vector Machine (SVM) binary classifier is trained using a training matrix, whereby each row represents characteristics that have been retrieved as an observation from a certain class. Following the completion of the training phase, the multiclass Support Vector Machine (SVM) model demonstrates the capability to accurately choose the appropriate class for a given input features vector. In order to categorize an item, its input characteristics vector is sequentially given to each binary classifier from the first to the Nth classifier until a negative result is obtained [32]. The Multi-class SVM is really useful if we want to do character by character classifiers, and very suitable for our requirement. The reason to use this multiclass SVM is because we have several output in this case 6 output article in category 1-6 in Table 1. Thus, one against all is very precise to get the better result compare to one against one multiclass SVM.

The SVM algorithm will learn based on training data and determine groups that match the predetermined labels. If the data has dimensions that are difficult to recognize, the shape will be changed to a higher dimension to make it easier to find the best hyperplane.

The parameters used in SVM modeling in this research include:

1. Parameter Regulation

The parameter C tells the SVM optimizer how much it wants to avoid misclassification of each training example. For large values of C, the optimization will choose a smaller margin hyperplane if that hyperplane does a better job of getting all training points correctly classified. Conversely, a very small value of C will cause the optimizer to search for a larger margin separation hyperplane, even if that hyperplane classifies more points. For very small values of C, we should get misclassification examples, often even if your training data is linearly separated.

2. One against all

The multi-class SVM technique used is one-against-all. This method solves the binary problem L_y to classify class j against all other classes with positive class samples and negative other samples.

3. Loss function

Loss calculations in model training use the squared hinge technique.

4. Iteration

The maximum training iteration is 1000 iterations.

After the model is formed, the model will be used for the system testing stage. The testing process will be carried out by calculating the support vector in the data. Then each vector in the test data will be compared with the vector from the learning model created. The testing data will be tested by comparing the values on the label. After testing is complete, accuracy will be obtained from the success of the system for classifying data.

2.7 Evaluation model

The evaluation method is the last process in text mining. The evaluation method is a calculation of the evaluation value to assess how well the results of the system are by comparing the system's accuracy to the actual results. The system evaluation used in this study are recall, accuracy, f-score, and precision, after we gain True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Below is the equation of every evaluation metrics:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$F - score = 2\,x\,\frac{Precision\;x\;Recall}{Precision + Recall} \tag{9}$$

3. RESULT AND DISCUSSION

3.1 Hardware and software specification

Hardware and software specifications used to build the system in this study are as follows:

1. Processor AMD A12-9720P Radeon R7, 12 Compute Cores 4C
2. 8GB of RAM
3. Hard drive capacity of 1024 GB
4. Windows 10 Pro operating system
5. Draw.io
6. Anaconda3 version 4.7.12
7. Python version 3.7.4
8. Jupiter notebooks
9. Qt Designer
10. Visual Studio Code
11. Python programming language libraries used: Pandas,

Nltk, Literary, Os, Io, Re, String, Sklearn, Pickle, Pyqt5

3.2 Training result

The training model aims to train the training data to find the best model before it is applied in the system. The training data will go through various processes, namely cleaning, case folding, stop word removal, stemming, and tokenization. After pre-processing, the process will continue to feature selection. The chosen method for the feature selection process in this study was Mutual Information. In this study, we selected 1500 out of 10164 features contained in the dataset. How to select is by calculating also MI value. MI value can be obtained using Eq. (2). After the MI value was obtained, the features will be sorted based on the MI value from the largest to the smallest. After the features are sorted, the features will be retrieved as many as the desired number of features. 1500 feature with highest MI value has been chosen. The example of the feature selection calculation can be seen in Table 8.

Table 8. Feature selection calculation

No	Term	MI Values
1	Rp	0.5198651970480264
2	Rupiah	0.49247978817830235
3	Juta	0.49247978817830235
4	Uang	0.44526896044325326
5	Ribu	0.443851000561505
6	Ratus	0.4425143934878197
7	Kirim	0.43274659681186717
8	transfer	0.42667210325835453
9	Nomor	0.42552067398594706
10	rekening	0.39507347104967405
...	....	....
10161	Merta	0.010607061615695086
10162	mmg	0.010607061615695086
10163	multi	0.010607061615695086
10164	munawir	0.010607061615695086

Table 9. Calculation results of TF, IDF, and TFIDF

No	TF	IDF	TFIDF
1	Perempuan	0.018072	2.879728
2	Tujuh	0.006024	2.636106
3	Telpon	0.006024	3.356652
4	Peran	0.006024	4.301114
5	Acara	0.008130	3.413811
6	Masjid	0.007477	4.167583
7	ambil	0.001965	2.781288
8	Screenshoot	0.001965	2.636106
9	Nomor	0.006593	1.682676
10	Terror	0.001418	4.637586

After the dataset features were selected, the next process would be feature weighting. Feature weighting aims to convert words into numbers. TFIDF method was applied for feature weighting in this study uses. For obtaining the results from TFIDF, the first thing to do would be to calculate the occurrence value of the term in the document (TF). The TF value was obtained by performing calculations using Eq. (3). Afterward, the Inverse Document Frequency (IDF) is calculated using Eq. (4). After the TF and IDF values were obtained then the TFIDF value could be calculated using Eq. (5). The calculation results of TF, IDF, and TFIDF values can be seen in Table 9.

After the dataset has been weighted, the next process is that the dataset is split into two with a ratio of 0.8 data used for training the classification model, while 0.2 data is randomly selected to become test data. After the dataset is split, training is carried out with the Support Vector Machine learning algorithm. SVM learning is done by finding the value of the support vector to find the best hyperplane value. The training data will be used as a trainer and the SVM algorithm will learn based on the training data and determine its grouping according to predefined labels. After the SVM algorithm learns to recognize the relationship pattern of each data with a label, a learning model will be produced. The learning model produced in this study stores the distribution values of the Support Vector Machine formula. For training model, the SVM Algorithm dataset learns the patterns of each sentence and learns the labels of each data, namely label 0 for article 27 paragraph 1, 1 for article 27 paragraph 3, 3 for article 27 paragraph 4, 3 for article 28 paragraph 1 and 4 for article 28 paragraph 2. After studying the pattern of fit from the data and labels, SVM will create a dividing line called a hyperplane.

After the classification model has been created, the model is then tested against 0.2 randomly selected data. The classification process will be carried out by calculating the support vectors in the dataset. Then each vector in the test data will be compared with the vector from the learning model created. After the test data has been tested, the results of the classification model training process will be obtained in the form of the level of accuracy produced by the SVM algorithm. The level of accuracy obtained in training this model was 95.45%, while the level of accuracy obtained in training the model without using MI feature selection was 86.36%.

3.3 Testing results

After the training modeling process was completed, the next step was to test the model using the remaining 0.2 partitions of the datasets. The obtained accuracy from model testing using MI is 95.45%, while the accuracy without using MI is 86.36%. In addition, the testing process was also performed on new test data that are not contained in the dataset. The new test data are stored in a txt file, which would then be tested in the system.

The increment of accuracy from using MI and without using MI is because of the use of mutual information selected from the feature selection process. This process will do weighting for every term with good mutual values. Then resulted TF-IDF is more narrow than any other normal TF-IDF. This also will give higher accuracy results in terms of SVM classification model.

The test was carried out by classifying all test data for cybercrime cases into 6 classes with a total of 35 test data. From these tests, the system testing results using MI can be seen in Table 10.

Based on Table 10, the system can classify 32 cases correctly and 3 cases incorrectly. After the system test results were obtained, the next process was to calculate the values of accuracy, recall, precision, and f-score. To obtain recall, accuracy, precision, and f-score values, it can be done by performing Eqs. (18), (19), (20), and (21) respectively. The results of these values can be seen in Table 11.

From Table 11, it is stated that the classification report results for accuracy, recall, precision, and f-score values are 91.42%, 90%, 82%, and 90.83% respectively. The model testing without using MI feature selection can classify data as much as 30 out of 35 data correctly and obtained an accuracy of 85.71%.

Table 10. System testing results

	27 Para 1	27 Para 3	27 Para 4	28 Para 1	28 Para 2	Other Cases
27 para 1	4	0	1	0	0	0
27 para 3	0	10	0	0	0	0
27 para 4	0	0	5	0	0	0
28 para 1	0	0	1	4	0	0
28 para 2	0	1	0	0	4	0
Other documents	0	0	0	0	0	5

Table 11. SVM classification results

Accuracy			**91.42%**
	Recall	**Precission**	**F-Score**
Article 27 paragraph 1	80%	100%	88.89%
Article 27 paragraph 3	100%	91%	95.29%
Article 27 paragraph 4	100%	71%	83.04%
Article 28 paragraph 1	80%	100%	88.89%
Article 28 paragraph 2	80%	100%	88.89%
Other documents	100%	100%	100%
Average	90%	82%	90.83%

3.4 Discussions

From the testing accuracy results, the use of Mutual Information as feature selection can improve the accuracy of the system created. The increase in system accuracy is due to the only features that are weighted are those that frequently appear in the dataset. When selecting the MI feature, it will discard words that don't appear often, so that when weighted with TF-IDF, there are no words that have a zero value.

Also from the classification results, the system can classify UU ITE cases fairly well. This is because the modeling process in this study was carried out by classifying data per case so that the algorithm reads the entire pattern of the cases, not individually. This causes in the system testing process, the input data is searched for a match in the model based on the pattern of cases that exist in the model. But the system created is not perfect. This is caused by several factors, namely the data used and the preprocessing process that is not quite right.

The data used in this study were 255 data for five classification classes of ITE Law types. The amount of data that is not large enough means that the patterns learned by the model are still small. Varied file sizes also affect the feature selection process to select features to be used in the next process. In addition, there are still many non-standard words in the dataset such as abbreviations, dates, currency amounts, and acronyms. Non-standard words affect the feature selection process, causing features to have the same meaning. Therefore, to solve this problem, word normalization is needed in the preprocessing process to convert non-standard words into standard words, to avoid that when the selection feature has features that have non-standard words. The system created in this study can only detect one label and sometimes cases of violations of the ITE Law are not only subject to one article. To overcome this problem, multi-label data labeling is needed so that it can detect types of cases that do not only have one type of article of the ITE Law.

4. CONCLUSION

Support Vector Machine (SVM) is able to classify types of law in cybercrime cases quite well. The accuracy results of the model evaluation using MI and without using MI were 95.45% and 86.36% using a dataset of 220 data. The accuracy value for testing data with and without MI feature selection were 91.42% and 85.71% from a total of 35 test data. These accuracy results are quite significant compared to previous model around 88-92%. The use of MI feature selection can increase the system accuracy because it only uses features that often appear in each class classified by the system. In the word weighting process, only the selected words will be weighted, while others will be discarded. The number of selected terms affects the accuracy results.

The amount of data used in each class for further research must be increased and the types of classes classified are not only six classes, so that all types of items can be classified. In future research, it is necessary to normalize words in the preprocessing process to change non-standard words into standard words, so that at the time of selection features there are features that have multilabel non-standard word so that it can classify data that has more than one type of article. Potential challenge in implementing these improvements is higher accuracy then before, we can consider using any language-based deep learning approach such as BERT or Indo-BERT, which in theory it will increase training time but with outperformed accuracy.

ACKNOWLEDGMENT

Author thanks to Universitas Sumatera Utara for the TALENTA Research Grant. Special thanks also for our partnership with others.

REFERENCES

[1] Westerlund, M. (2019). The emergence of deepfake technology: A review. Technology Innovation Management Review, 9(11): 39-52.

[2] Fernandez, C.B., Hui, P. (2022). Life, the Metaverse and everything: An overview of privacy, ethics, and governance in Metaverse. 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 272-277.

[3] Phillips, K., Davidson, J.C., Farr, R.R., Burkhardt, C., Caneppele, S., Aiken, M.P. (2022). Conceptualizing cybercrime: definitions, typologies and taxonomies. Forensic Sciences., 2(2): 379-398. https://doi.org/10.3390/forensicsci2020028

[4] Yemanov, V., Dzyana, H., Dzyanyi, N., Dolinchenko, O., Didych, O. (2023). Modelling a public administration system for ensuring cybersecurity. International Journal of Safety and Security Engineering, 13(1): 81-88. https://doi.org/10.18280/ijsse.130109

[5] Lukings, M., Habibi Lashkari, A. (2022). Cybersecurity and cybercrimes. Understanding Cybersecurity Law and Digital Privacy, 59-96. https://doi.org/10.1007/978-3-030-88704-9_3

[6] Pawar, S.C., Mente, R.S., Chendage, B.D. (2021). Cyber crime, cyber space and effects of cyber crime. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 7(1): 210-214. https://doi.org/10.32628/CSEIT217139

[7] Riadi, I., Yudhana, A., Fanani, G.P.I. (2023). Mobile forensic tools for digital crime investigation:

Comparison and evaluation. International Journal of Safety and Security Engineering, 13(1): 11-19. https://doi.org/10.18280/ijsse.130102

[8] Ishak, N. (20223). Guarantee of information and communication technology application security in indonesia: regulations and challenges? Audito Comparative Law Journal, 4(2): 108-117. https://doi.org/10.22219/aclj.v4i2.26098

[9] Safiranita, T., Waluyo, T.T.P., Calista, E., Ratu, D.P., Ramli, T.S. (2021). The Indonesian electronic information and transactions within Indonesia's broader legal regime: urgency for amendment? Jurnal HAM, 12(3). http://doi.org/10.30641/ham.2021.12.533-552

[10] Lubis, M., Maulana, F.A. (2010). Information and electronic transaction law effectiveness (UU-ITE) in Indonesia. In Proceeding of the 3rd International Conference on Information and Communication Technology for the Moslem World (ICT4M) 2010, pp. C-13-C–19. https://doi.org/10.1109/ICT4M.2010.5971892

[11] Lubis, F.S., Lubis, M., Hakim, L. (2022). Investigation of netizen sentiment analysis toward the controversy of information and electronic transaction law. 2022 Seventh International Conference on Informatics and Computing (ICIC), pp. 1-7. https://doi.org/10.1109/ICIC56845.2022.10006918

[12] Morina, M., Azemi, F., Eren, M., Zejneli, I., Papajorgji, E. (2023). Crime scene in cybercrime criminal offenses: evidence management and processing. acad. j Academic Journal of Interdisciplinary Studies, 12(2): 179. https://doi.org/10.36941/ajis-2023-0041

[13] Rahmat, R.F., Faza, S., Adnan, S., Situmorang, D.T.E., Gunawan, D., Lini, T.Z. (2021). News articles classification for electronic information and transaction law in Indonesia using support vector machine. 2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), pp. 106-110. https://doi.org/10.1109/DATABIA53375.2021.9650285

[14] Wardani, N.K., Afriansyah, A. (2020). Indonesian legal challenges regarding electronic contracts in international trade. 3rd International Conference on Law and Governance (ICLAVE 2019), pp. 23-30. https://doi.org/10.2991/aebmr.k.200321.004

[15] Suryono, R.R., Budi, I., Purwandari, B. (2021). Detection of fintech P2P lending issues in Indonesia. Heliyon, 7(4): e06782. https://doi.org/10.1016/j.heliyon.2021.e06782

[16] Saputra, P.Y., Yunianto, D.R., Arissandy, S.S. (2019). Pencarian pasal pada UU ITE berdasarkan kasus cyber crime dengan metode Latent Semantic Indexing (LSI). Seminar Informatika Aplikatif Polinema, pp. 126-130.

[17] Hakim, L., Kusumasari, T.F., Lubis, M. (2018). Text mining of UU-ITE implementation in Indonesia. Journal of Physics: Conference Series, 1007(1): 12038. https://doi.org/10.1088/1742-6596/1007/1/012038

[18] Yıldırım, S., Jothimani, D., Kavaklıoğlu, C., Başar, A. (2018). Classification of" hot news" for financial forecast using NLP techniques. 2018 IEEE International Conference on Big Data (Big Data), pp. 4719-4722. https://doi.org/10.1109/BigData.2018.8621903

[19] Zheng, Q., Tian, X., Yang, M., Su, H. (2019). The email author identification system based on Support Vector Machine (SVM) and analytic hierarchy process (AHP). IAENG International Journal of Computer Science, 46(2): 178-191.

[20] Çelenli, H.İ., Öztürk, S.T., Şahin, G., Gerek, A., Ganiz, M.C. (2018). Document embedding based supervised methods for Turkish text classification. 2018 3rd International Conference on Computer Science and Engineering (UBMK), pp. 477-482. https://doi.org/10.1109/UBMK.2018.8566326

[21] Paul, A.K., Shill, P.C. (2016). Sentiment mining from bangla data using mutual information. 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), pp. 1-4. https://doi.org/10.1109/ICECTE.2016.7879569

[22] Kusumaningrum, R., Wiedjayanto, M.I.A., Adhy, S. (2016). Classification of indonesian news articles based on latent Dirichlet allocation. 2016 International Conference on Data and Software Engineering (ICoDSE), pp. 1-5. https://doi.org/10.1109/ICODSE.2016.7936106

[23] Romsaiyud, W., Nakornphanom, K.N., Prasertsilp, P., Nurarak, P., Konglerd, P. (2017). Automated cyberbullying detection using clustering appearance patterns. 2017 9th International Conference on Knowledge and smart Technology (KST), pp. 242-247. https://doi.org/10.1109/KST.2017.7886127

[24] Alami, S., Elbeqqali, O. (2015). Cybercrime profiling: Text mining techniques to detect and predict criminal activities in microblog posts. 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1-5. https://doi.org/10.1109/SITA.2015.7358435

[25] Suliani, I., Asnal, H., Suryati, L., Efendi, R. (2022). Sentiment analysis for abolition of national exams in Indonesia using support vector machine. Engineering Letters, 30(4).

[26] Mothukuri, R., Basaveswararao, B., Bulla, S. (2020). Judgement classification using hybrid ANN-Shuffled frog leaping model on cyber crime judgement database. Revue d'Intelligence Artificielle, 34(4): 445-456. https://doi.org/10.18280/ria.340409

[27] Cheng, P., Hao, W., Dai, S., Liu, J., Gan, Z., Carin, L. (2020). Club: A contrastive log-ratio upper bound of mutual information. International Conference on Machine Learning, pp. 1779-1788.

[28] Zhao, S., Wang, Y., Yang, Z., Cai, D. (2019). Region mutual information loss for semantic segmentation. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

[29] Ridho Lubis, A., Nasution, M.K.M., Salim Sitompul, O., Muisa Zamzami, E. (2021). The effect of the TF-IDF algorithm in times series in forecasting word on social media. Indonesian Journal of Electrical Engineering and Computer Science, 22(2): 976. https://doi.org/10.11591/ijeecs.v22.i2.pp976-984

[30] Lubis, A.R., Nasution, M.K.M., Sitompul, O.S., Zamzami, E.M. (2020). A framework of utilizing big data of social media to find out the habits of users using keyword. Proceedings of the 8th International Conference on Computer and Communications Management, New York, NY, USA: Association for Computing Machinery, pp. 140-144. https://doi.org/10.1145/3411174.3411195

[31] Jing, C., Zhao, H.P., On, C.K., Moung, E.G., Anthony, P. (2022). Face recognition based on deep convolutional support vector machine with bottleneck attention.

IAENG International Journal of Computer Science, 49(4).

[32] Oujaoura, M., Minaoui, B., Fakir, M., El Ayachi, R., Bencharef, O. (2014). Recognition of isolated printed tifinagh characters. International Journal of Computer Applications, 85(1). https://doi.org/10.5120/14802-3005