ASEAN Engineering
Journal

Full Paper

TOWARDS CURBING CYBER-BULLYING IN MALAYSIA BY AUTHOR IDENTIFICATION OF IBAN AND KADAZANDUSUN OSN TEXT USING DEEP LEARNING

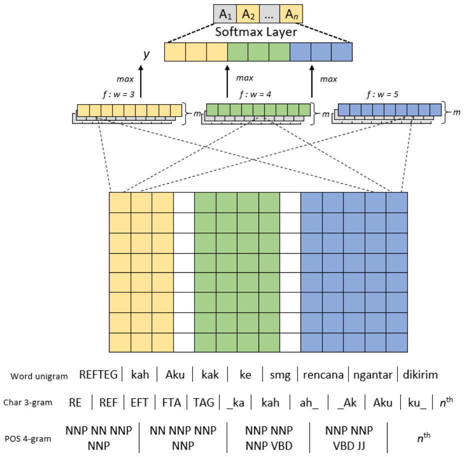
Nursyahirah Tarmizi[*], Suhaila Saee, Dayang Hanani Abang Ibrahim

Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 14300, Kota Samarahan, Sarawak, Malaysia

Article history
Received
23 September 2022
Received in revised form
03 January 2023
Accepted
04 January 2023
Published online
31 May 2023

*Corresponding author
17020134@siswa.unimas.my

Graphical abstract



Abstract

Online Social Network (OSN) is frequently used to carry out cyber-criminal actions such as cyberbullying. As a developing country in Asia that keeps abreast of ICT advancement, Malaysia is no exception when it comes to cyberbullying. Author Identification (AI) task plays a vital role in social media forensic investigation (SMF) to unveil the genuine identity of the offender by analysing the text written in OSN by the candidate culprits. Several challenges in AI dealing with OSN text, including limited text length and informal language full of internet jargon and grammatical errors that further impact AI's performance in SMF. The traditional AI system that analyses long text documents seems inadequate to analyse short OSN text's writing style. N-gram features are proven to efficiently represent the authors' writing style for shot text. However, representing N-grams in traditional representation like Tf-IDF resulted in sparse and difficult in grasping the semantic information from text. Besides, most AI works have been done in English but receive less attention in indigenous languages. In West Malaysia, the supreme languages that transcend ethnic boundaries are *Iban* of Sarawak and *KadazanDusun* of Sabah, which both are inherently under-resourced. This paper presented a proposed workflow of AI for short OSN text using two Under-Resourced Language (U-RL), *Iban* and *KadazanDusun* tweets, to curb the cyberbullying issue in Malaysia. This paper compares Tf-Idf (sparse) and SoA embedding-based (dense) feature representations to observe which representations best represent the stylistic features of the authors' writing. N-grams of word, character, and POS were extracted as the features. The representation models were learned by different classifiers using machine learning (Naïve Bayes, Random Forest, and SVM). The convolutional neural network (CNN), a SoA deep learning model in sentence classification, was tested against the traditional classifiers. The result was observed by combining different representation models and classifiers on three datasets (English, Iban, and KadazanDusun). The best result was achieved when CNN learned embedding-based models with a combination of all features. KadazanDusun achieved the highest accuracy with 95.76%, English with 95.02%, and Iban with 94%..

Keywords: Author Identification, authorship analysis, stylometry, social media, cyberbully**.**

1.0 INTRODUCTION

Online Social Network (OSN) is increasingly popular and has become an excellent socialising tool. OSN gets people connected without any boundary as of distance and time. OSN platforms like Twitter, Facebook, and Instagram facilitate the creation and sharing of various forms of content such as micro-blogs, videos, and photos. However, online activities via OSN are often being abused. Online violence and cyberbullying in OSN keep growing each day. According to Ghazali *et al.*, 2017, Malaysia have recorded an increment from 2013 (55.6%) to 2015 (62.3%) for cyber-bullying cases among the Malaysian youth [1].

Mohammad (2021) reported that three out of every ten Malaysian adolescents have been bullied, as stated by United Nations Children's Fund (UNICEF) [2]. Regrettably, Malaysia ranked sixth out of 28 countries in the global poll [3] and second in Asia [4] on cyber-bullying. Victims of cyberbullying often keep silent and go unreported because they do not want to be entangled in prolonged legal battles to seek relief. Besides, tracing the culprit of cyberbullying can be daunting as the anonymity issue in social media is a major drawback.

Current digital forensics solutions seem inadequate in tracing the real culprit of cyberbullying due to the anonymity issue. The perpetrators exploited the Privacy Enhancing Technologies (PET), for instance, the Onion Router network (TOR), to hide their identities. TOR technology provides a way to tunnel traffics through a series of the proxy server and make it harder to locate the originating IP address. Advanced network technology like TOR complicates network forensics to track down the culprits of cyberbullying as the real identity of the culprit is invisible behind the veil of anonymity. In such a case, the only clue left in tracing the real identity of the culprit is the text posted in the OSN.

Why text? In recent years, a steady increase of OSN users resulted in an abundance of text generated by the users. According to the statistics company, Statista, Twitter is currently among the most popular OSN worldwide, with 196 million of active users that post micro-blog, the so-called tweets [57]. In stylometry (a field of writing style study), a person's writing style is unique thus, text posted online can act as the digital fingerprint or so-called writeprint [5]. Hence, the writeprint can facilitate tracing the real identity of the culprit by analysing the writing samples. Authorship analysis is a process that analyses the writing style to draw a conclusion on its authorship.

Juola (2008) indicates that authorship analysis can be divided into three major tasks, namely, Author Identification (AI), Authorship Profiling (AP), and Authorship Verification (AV) [6]. AI predicts the most plausible author of the disputed text among the candidate authors. AP is focused on identifying demographic characteristics (such as age, education level, and gender) or psychological characteristics of the author. In contrast, AV provides an answer to whether the same author also writes the examined text. AI task seems suitable for determining the real culprit of cyberbullying by facilitating the social media forensics (SMF) to identify the author of the anonymous text message posted in OSN.

The goal of AI is to examine candidate authors' writing samples and predict the most likely author of an anonymous text [7]. AI is very benificial towards SMF in cybersecurity, social media analytics, and digital humanities. AI task can be viewed as a multi-class single-label text classification problem from the machine learning perspective [8]. AI includes classifying text based on its author, given a closed-set of candidate authors and their writing samples. Closed-set AI assumed that the author of the text under investigation is necessarily a member of a given well-defined set of candidate authors [6]. AI task involves the extraction of features with high discriminatory potential between the candidate authors, so-called stylistic features, followed by feature representation and classification process.

There are two types of data that are being analysed in AI , which are short text and long text. Long texts data composed of texts from novel excerpts and books for historical literature study. Traditional AI methods that analyse long and formal text are considered unreliable for OSN text as the accuracy will drop significantly with the unique structure of OSN text. In OSN platforms (i.e. Facebook and Twitter.) people often write less and express their feelings in a limited number of words [14]. For instance, Twitter posts (tweets) concise up to 240 characters, resulting in insufficient writing style information extracted from the text to classify the authors. OSN text contains an abundance of Internet jargon (e.g., LOL, g2g, btw) in expressing ones feelings and misspelled words, unlike formal text. Each OSN platform has its unique native language; for instance, in Twitter, the use of hashtags "#" at the beginning of a word is to search for specific content, alias "@" followed by a Twitter username is to direct message another Twitter user and retweet "RT" to repost other Twitter user's message. Distinct characteristics of OSN text impose a challenge in modeling a robust AI system for SMF. Thus, a viable and "non-traditional" attribution technique in capturing the diversity of OSN language is needed to tweak the performance of AI in OSN circumstances.

Moreover, most AI works have been done in English, but the subject has received less attention in indigenous languages. Malaysia is a diversified multi-ethnic country with many different indigenous languages spoken in East and West Malaysia, besides *Bahasa Melayu* as the official language. In East Malaysia of the Borneo island, the dominant indigenous languages spoken by the natives are *Iban* in Sarawak and *KadazanDusun* in Sabah [9]. Despite having a standard writing system, *Iban* and *KadazanDusun* are under-resourced languages (U-RL) as "the compendium of written corpora in each of these two languages is still thin compared to Malays" said [10]. It is afraid that these indigenous languages will slowly diminish through modernization and advancement in social standing [9].

Various parties have made numerous efforts, including local and abroad foundations and organizations, to sustain the indigenous languages in Malaysia through research, translation, documentation, training, and development [11], as these languages are the national identity and treasure. This work attempts to leverage the usage of indigenous languages through OSN parallel to the UNESCO (2021) strategy in the International Year of Indigenous Languages 2019 to elevate indigenous languages for sustainable development, good governance, reconciliation, and peacebuilding in the societies [12]. Therefore, the issues addressed in this paper regarding AI for SMF have led to three research questions which are:

i. How important are the native features of OSN text and tweets in AI for SMF?
ii. Which feature representation model befitted AI in SMF?
iii. Which classifier produces optimal performance in AI for SMF?

Thus, a competent AI system is required to combat cyberbullying in Malaysia by analysing and classifing Malaysian local languages in order to facilitate digital forensics investigations. This paper introduces AI workflow for SMF using U-RL datasets containing *Iban* and *KadazanDusun* tweets. N–grams of word, character, and POS are extracted from tweets since these properties are recognised to be useful in dealing with short and noisy text like [13 - 16]. Nonetheless, n-grams produced sparse representation and a lack of meaningful interpretation [17] .

This paper compares two text representation models, Tf-IDF and embedding-based models. Tf-IDF is a sparse matrix representation, while the embedding-based model is dense and distributed. AI's classic SoA machine learning methods including, SVM, Naïve Bayes, and random forest are evaluated against the newly emerging deep learning algorithm,CNN, (especially in text classification) to learn the feature vector. This paper observes different text representation models in representing the authors' writing style of the tweets and variations of SoA classifiers to classify the tweets in an optimum way.

Shrestha *et al.,* (2017) is the first to use a deep learning model in AI using CNN algorithm with character n-grams as the stylistic feature. The model was evaluated using 1,000 tweets per author, and there are 50 candidate authors. The model achieved an accuracy of 76.1% and outperformed the current SoA, SCAP. Rocha *et al.,* (2017) proposed that combining stylistic features, i.e., character, word, and POS n-grams, effectively attributes the authors of short OSN text [19]. Fourkioti *et al.,* (2019) reported that combining language models with characters and POS 3-grams as features achieved 54% accuracy on the tweets corpus. Nevertheless, the model was well-performed on the movie reviews corpus with 96%. The latter corpus is practical with language models as movie reviews are homogeneous with respect to the topic, while the tweets have a unique and noisy writing style with limited text length.

Jambi *et al.,* (2021) examined the potential of SoA classifiers (i.e., k-NN, Random Forest, and SVM), to predict the author of Arabic tweets. Several stylistic features, including character-based features, lexical features, and syntactic features, were used to quantify the tweets writing style. Lexical-based features are more potent than character-based and syntactic features, yet combining the stylistic features showed accuracy improvement. Random forest reported better than SVM and k-NN with a gradual reduction in accuracy when the number of authors increased. Besides prominent machine learning classifiers, deep learning models like neural networks are gaining attention in text classification [21] as well as in AI task [16, 22].

The rest of the paper is organised as follows: Section 2 describes the related works in three correlated fields: digital forensics, online social networks, and authorship identification. Section 3 outlines a proposed workflow of AI for OSN text using U-RL tweets as datasets. Section 4 presents the experimental settings to perform different experiments. Section 5 discusses the results obtained. Section 6 states the conclusion of the presented work.

2.0 REVIEWS OF AI WORKS IN SMF

Before the electronic text age, authorship analysis mainly focused on literary and historical text such as the bible, poems, prose, plays, articles, and essays to solve disputed authorship problems. The employment of computers on authorship analysis marked the beginning of authorship attribution in the late 1880s [23]. An early publication was the statistical analysis of word-length distribution in discriminating the authors of novels [24] and Shakespeare's poems [25]. Yule (1939) reported on the authorship analysis of the *De Imitatione Christi* book using sentence length distribution [26]. Ellegard (1962) in his stylistic study demonstrated the use of rare words and expressions in determining the authorship of *Junius* letters [28].

In the forensic investigation, the early publication was identified in 1963, where N.Tenow solved disputed authorship on a Swedish *Halender Slandering Letters* case using words, phrases, and syntactic constructions as the stylistic features [29]. Morton (1978) applied the cumulative sum (CUSUM) technique using sentence length and habit word occurrences to solve the disputed authorship of accused statements written by police [30]. Bailey (1979) applied authorship attribution to solve the authorship of *Patricia Hearst Letters* case [31]. Until the late 1980s, most works in authorship analysis employed univariate analysis such as frequency distribution of the average number of variables over a certain number of word length to solve authorship problems covering literary and forensics cases.

Shifting of techniques from univariate to multivariate analysis can be seen in the early of 1990's to solve disputed authorship [32, 33 ,34]. Burrows (1992) applied principle component analysis (PCA) to plot the most frequent words from *Jane Austen's* novels to solve the disputed authorship problem [35]. Most high frequency words were the function words such as 'to,' 'by,' 'from,' 'the,' extracted as features. Aside from the famous frequent words distribution [36, 37, 38], there are works started to explore word and character n-grams [39, 40] in attributing authorship through multivariate technique.

Computational stylistics in AI continue to grow in 2010 using machine learning methods including exponentiated gradient [41], Bayesian model [42], SVM [43], and Neural Network [44] to handle high dimensional features due to the increased number of authors and electronic documents. The emerging of online social network (OSN) has increased proportionally to cyber-crime activities like cyberbullying, hate speech, and fake news. Recently, works on authorship analysis in OSN text are actively explored to combat cybercrimes through digital text forensics and stylometry [45]. In AI for SMF, various employment of tweets as testbed in addressing OSN and short text issues.

Character n-grams are frequently employed in authorship attribution for short text because they tolerate to non-standard punctuation and typos [8]. Altakrori *et al.,* (2018) inferred character n-grams as features [45]. The proposed model trained by Random Forest gained the best accuracy of 55% [46]. Jambi *et al.,* (2021) used character-based and other stylometric features i.e. lexical and syntactic extracted from tweets. Character-based model reported to achieve high accuracy (68.5%) using SVM as classifier [21]. Combining different feature sets (i.e. combining character and word n-grams with POS tag n-grams) is a promising path due to the complex problem of identifying authors of OSN text [14, 19].

However, the drawback of n-grams is that they produce sparse features as the number of *n* increases [47]. Traditional feature representations, such as Tf-Idf, suffers from data sparsity and high dimensionality in representing n-grams and have difficulty in grasping the semantic meaning of texts [15]. Studies in AI starts to employ deep learning and embeddings to enhance the performance of AI system in attributing the author of OSN text [16, 48].

Theóphilo *et al.,* (2019) presented a deep learning technique for gauging small text messages posted in OSN on fake news issues [50]. The proposed model using character 4-grams with CNN achieved the highest accuracy in 400 epochs with 65% validation accuracy. Huang *et al.,* (2020) presented a novel method, joint of word and character n-grams trained by Bi-LSTM and CNN model on tweets [49]. The CNN-based model performed better than LSTM. Theóphilo *et al.,* (2021) in their extended work shows improvement in their model by implementing data augmentation techniques by amplifying the training samples to polish the model's performance. The proposed message augmentation model produced better result with 74.15% [16].

OSN text is idiosyncratic and evolving hence complicates the AI model in representing the authors’ writing style to maintain an optimum performance for SMF. However, a person writing style remains contant despite the source of the text might be varied. Deep learning and embeddings models seem to pave a promising path in enhancing the performance of AI to better representing and predicting the author of short OSN text to facilitate SMF in curbing cyberbullying issues.

3.0 METHODOLOGY

AI task can be viewed as a multi-class single-label text classification problem which consist three consecutive phases: Text Pre-processing, Text Representation and Classification. This paper specifically implied a closed-set attribution approach for AI, where the set of authors is predefined and supervised data is required. The setting fits the forensic applications where the writing samples are assumed as the validated samples of the candidate authors.

The dataset is composed of tweets extracted from Twitter and represented as a list of stylistic features-value pairs followed by the class label (which is the author). One of the main concerns in AI is the appropriateness and richness of the stylistic features. If the features are not well-represented, the classifier is unable to perform well in classification and prediction. Effective feature representation is needed to successfully capture the idiosyncrasies and grasp most of the information in the writing style. Thus, this study compares two feature representation techniques, Tf-Idf and embedding-based, to observe which representation model best represents the writing style of OSN text. As for classification, a classifier is trained with the representation models of the documents using Convolutional Neural Networks (CNN) as reflected in Figure 1.

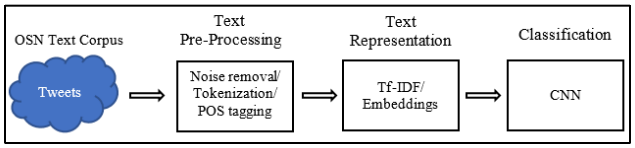


Figure 1 AI workflow

3.1 Tf-Idf as Sparse Representation Model

Term frequency-inverse document frequency (Tf-Idf) is used to weight the features, in this case, n-grams of tweets. Tf-Idf evaluates how vital is each N-gram to a tweet in the whole collection of tweets. The importance of an n-gram increases proportionally to the number of times it appears in the tweet but, is offset by its frequency in the corpus. Tf-Idf first computes the normalised Term frequency (Tf), which appears to be the number of times an n-gram appears in a tweet, divided by the total number of n-grams in the tweet.

$$Tf(n) = \frac{(Number\ of\ times\ ngram\ n\ appears\ in\ a\ tweet)}{(Total\ number\ of\ ngrams\ in\ the\ tweet)}$$

The inverse document frequency (Idf) is then computed as the logarithm of the number of tweets in the corpus, divided by the number of tweets where the specific N-gram appears.

$$Idf(n) = log\ \log\left(\frac{Total\ number\ of\ tweets}{Number\ of\ tweets\ with\ ngram\ (n)\ in\ it}\right)$$

N-grams have been proven successful in capturing detailed stylistic information on the author's lexical, syntactic, and structural preferences. N-grams also produced outstanding results in classifying authors of short text and multilingual text as they can indicate grammatical and orthographic tendencies without the need for linguistic background knowledge. The table 1 shows a tweet in *Iban* language with the corresponding Tf-Idf value of word unigram, character 3-grams and POS 1-4 grams:

Table 1 N-grams of Iban tweet with their corresponding Tf-idf values

Iban tweet:	“REFTAG kah Aku kak ke smg rencana ngantar dikirim”
Word Unigram:	'REFTAG' 'kah' 'Aku' 'kak' 'ke' 'smg' 'rencana' ‘ngantar' 'dikirim'
Tf-Idf value:	0.08, 0.26, 0.26, 0.3, 0.23, 0.41, 0.41, 0.43
Char 3-grams:	'_Ak' '_RE' '_di' '_ka' '_ke' '_ng' '_re' '_sm' 'AG_' 'Aku'
Tf-Idf value:	0.18, 0.06, 0.12, 0.19, 0.1, 0.12, 0.18, 0.2, 0.05, 0.18
POS {1:3}-grams:	'JJ' 'JJ NN' 'JJ NN NN' 'NN' 'NN NN' 'NN NNP' 'NN NNP NNP'
Tf-Idf value:	0.15, 0.17, 0.19, 0.26, 0.1, 0.14, 0.22

Based on Table 1, Tf-Idf values of word, character, and POS n-grams features were computed using sklearn, *TfidfVectorizor*. The *TfidfVectorizor class* calculates the Tf-Idf vector of n-gram present in the document. Every row in the sparse matrix represents the tweets, and columns represent the unique n-gram from all the tweets and the values in the data frame table represent the Tf-Idf value of that unique n-gram given in tweets.

However, the downside of Tf-Idf n-grams is that it suffers from data sparsity and high dimensionality. The number of vectors grows proportionally to the number of unique n-grams. On top of that, Tf-Idf representation has trouble grasping semantic meaning or word distances. For instance, words like “powerful,” “strong,” and “Paris” are equally distant, although semantically, “powerful” should be closer to “strong” than “Paris.” Distributed representation based on word learns semantic relationships. Distributed representation enables words with similar meanings have similar representations. Unlike the bag of words model, each unique words have different representations unless properly managed.

3.2 Distributed Embeddings Representation

Embedding-based representation represents words or n-grams that capture their meanings, semantic relationships, and the different types of contexts. There are four techniques: the embedding layer, Word2Vec, GloVe, and FastText algorithms to construct an embedding-based representation. The embedding layer technique is an n-gram embedding of a word, character, or POS that is learned in conjunction with a neural network model.

Word2Vec is a predictive embedding model that uses either continuous bag-of-words (CBOW) or continuous skip-gram model to train an embedding from a corpus [51]. GloVe algorithm extends Word2Vec by using matrix factorization to capture word meaning in vector space.

FastText is a predictive embedding model developed by Facebook that has a similar approach to GloVe and Word2Vec, but it is a better model that includes word and character levels [52] FastText n-grams enable capturing rare words that both Word2Vec and GloVe models cannot achieve. In this paper, the embedding layer technique is employed to represent the features as dense vectors.

Figure 2 illustrates the conversion process of n-grams into sequences of integers. The Keras *Embedding layer* requires integer inputs where each integer maps to a single token with a specific real-valued vector representation within the embedding. Keras *Embedding Layer* starts with random weights, and it will learn an embedding for each of the n-grams in the training dataset. These vectors are random at the beginning of training but later become meaningful to the network. The input n-grams are pre-processed beforehand to produce n-grams as padded sequences of integers.

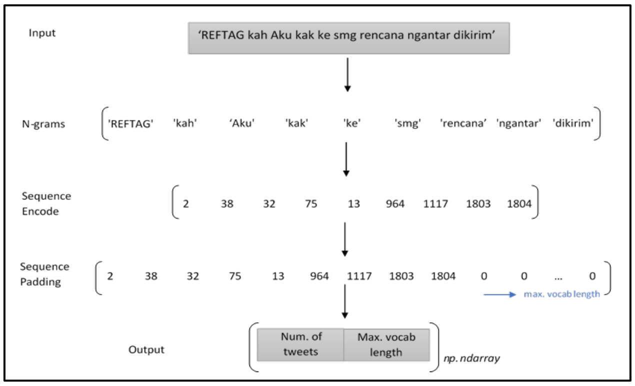


Figure 2 A process flow of transforming lists of n-grams into a sequence of integers

Training documents are encoded as sequences of integers using the *Tokenizer class* in the Keras API. The *texts_to_sequences()* function on the *Tokenizer* encodes the reviews in the training document. A dictionary of all tokens in the training dataset and a mapping from the tokens to distinct integers are constructed. Then, the sequence of integers is padded to the length of the longest tweets with zero padding by calling the *pad_sequences()* function. The padding process is mandatory as it is the requirement of Keras for efficient computation. After the sequence padding, the neural network classification model will use the *Embedding layer* as the first hidden layer. The *Embedding Layer* of Keras prompts the vocabulary size or the number of n-grams, dimension size, and the maximum length of the tweets.

3.3 Author Identification using Deep Learning Model

Recently, Convolutional Neural Networks (CNN) has gained attention in achieving powerful performance on AI task [16, 18, 50], In extending the work of [18], this paper proposed a Convolutional Neural Network (CNN) architecture using a sequence of different n-grams levels (word, characters, POS tags and mix) as input as depicted in Figure 2.

As shown in Figure 3, the input of padded sequence n-grams is passed through the Embedding Layer, Convolutional Layer, and lastly, to the fully connected *softmax* layer. The maximum length, *l*, of the n-gram sequences determines the input size. The Embedding layer receives the vocabulary size or the number of n-grams, the size of dimension, and the maximum length of the tweets. Then, the embedding layer produce a matrix, $C \in R^{d \times l}$, where the rows represent the sequence of n-grams, while the columns represent their embedding c_j of position j.

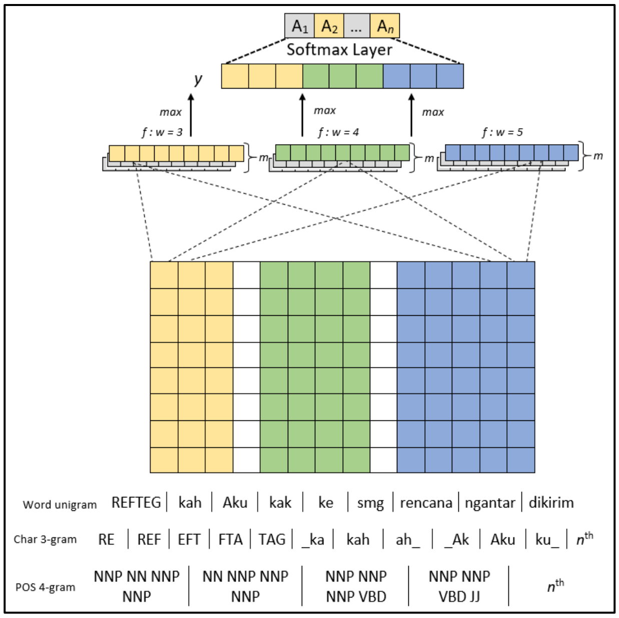


Figure 3 N-gram CNN architecture [18]

Next, in the convolutional layer, a convolutional filter, $H \in R^{d \times w}$, is applied to a portion of C, where w is the width of the filter. The resulting matrix, O, is used as input to a *relu* function g, along with a bias term b to produce feature representation f for the tweets:

$$O = H \cdot C[i : i + w - 1]$$

$$f = g(H \cdot C[i : i + w - 1] + b), f \in R\ l - w + 1$$

In the *1D* convolutional layer, different widths w are used to capture morpheme-to-word patterns. The *maximum pooling* layer condensed the feature maps f, to obtain y_k, the maximum value of each feature map f_k , where m is the number of feature maps.

$$yk = max\ i\ fk[i], k = 1 \ldots m$$

Pooling is necessary to represents the text's most crucial features, regardless of their position. After pooling and merging the feature representations y_k, a compact representation of the text is yielded. To convert the three-dimensional output to two-dimensional for merging, a *flatten* layer is employed.

Lastly, the dense feature matrix is passed to the fully connected layer of *softmax* layer. Table 1 describes the combination of hyper-parameters for the three layers of CNN. Additionally, *batch normalization* layer is added right after the convolutional layers to maintain the mean output by normalizing the output values using the mean and standard deviation of the batch of inputs.

Table 2 CNN architecture hyper-parameters

Layer	# of layers	Hype-parameters	
Embedding	1	*l*	max_length
		d	300
Convolutional 1D	3	*m*	[128,128,128]
		w	[3,4,5]
		pooling	Max1D
Fully connected	1	Num. of author	1290

In Table 2, the hyper-parameters, including *Adam* optimizer is used to compile the CNN model, and *sparse_categorical_crossentropy* is used to calculate the loss. Apart from that, *SparseCategoricalAccuracy* is used to calculate the accuracy for 100 epochs. The model is validated using *k-fold* cross-validation, where accuracy and F1-score are computed every ten folds using sklearn, *F1_score_micro*. Mean accuracy and mean micro F1-score are calculated as the final result.

4.0 EXPERIMENTAL SETTING

4.1 Description of the Dataset

Malaysia, notably Borneo, has a diverse range of indigenous languages. An indigenous language has stable group of speakers and a genetic link with other native languages in the same region[10]. *Iban* of Sarawak and *KadazanDusun* of Sabah have countless native speakers on Borneo Island. Although *Iban* and *KadazanDusun* are widely spoken among their native speakers, they are considered under-resourced languages (U-RLs). This is owing to the minimal online presence and absence of technological tools for speech and language processing [53] for *Iban* and *KadazanDusun* text. Hence, it is essential for Natural Language Processing (NLP) to process and analyse OSN text of these languages involved with cyberbullying activities for forensics investigation. Tweets in *Iban* and *KadazanDusun* languages were extracted as a U-RL dataset in this study to test the performance of AI for SMF on the U-RL text.

The process of attaining tweets is done using a feedlist of profane words regularly used by the native speakers of *Iban* and *KadazanDusun* on Twitter. The interest in tweets with vulgar or profane words is to mimic the inappropriate content posted by potentially offensive Twitter users. Tweets are streamed using the feedlist through the Twitter API class, *TwitterStreamListener*, a listener that handles tweets received from the stream. The result is a list of 50 recent tweets comprising profane phrases from the feedlist. Below are the examples of profane words in the feedlist for three languages in Table 3.

It is observed that some *Iban* and *KadazanDusun* words share the same word with the same meaning. For example, 'budu' (in Table 3) means 'stupid' in both languages. Moreover, there are hardly any profane words in *KadazanDusun*. The speakers of *KadazanDusun* usually express their cursing in the form of euphemisms or using taboo words [54]. Thus, the *KadazanDusun* feedlist comprises taboo words and general verbs to acquire tweets in this language.

Table 3 Profane words of English, Iban, and Kadazan-Dusun

English	*Iban*	*KadazanDusun*
asssucker	paloi	silaka
bitchy	basug	basug
dick	taru	mulau
fuckhole	tai nu	kapatai
vjayjay	toi gia	budu
whorebag	osonong	mimboros

Ten prolific users with their tweets are extracted for Iban and KadazanDusun, respectively. With the feedlist, the U-RL datasets of Iban and KadazanDusun short OSN text are produced. The datasets consist of 10 authors, and 500 tweets represent each author, as shown in Figure 4.

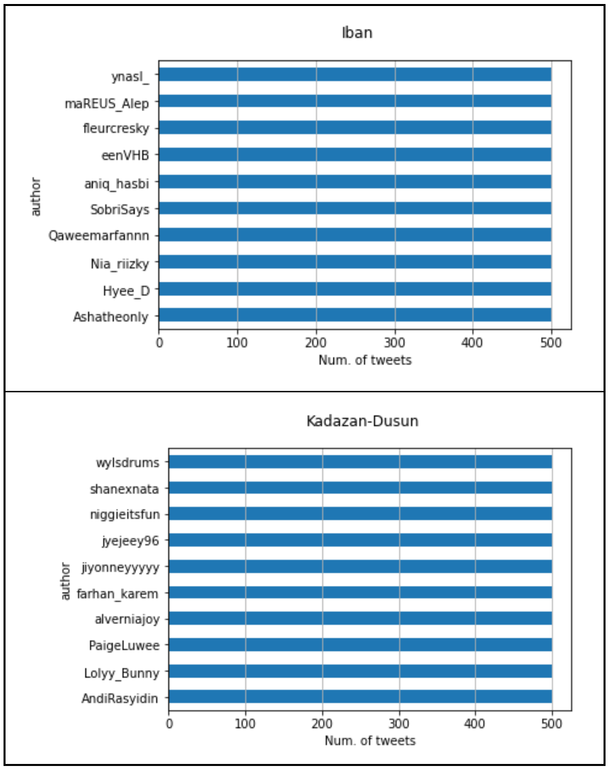


Figure 4 U-RL Datasets of *Iban* and *KadazanDusun*

Figure 4 illustrates two bar charts displaying the number of tweets assigned to each author. Both datasets have ten authors and the same number of tweets (500 tweets) for each author. However, the word and character frequency for each dataset are different. Figure below shows the frequency of tokens and characters between English, Iban, and KadazanDusun datasets.

Figure 5 depicts the total number of tokens and characters counted in 500 tweets for all the datasets. Although the number of tweets were balanced, the total number of tokens and characters were still imbalanced. Due to the imbalanced number of tokens and characters in each datasets, the performance of the AI system may affected as the stylistic features strongly depends on the tokens and characters.

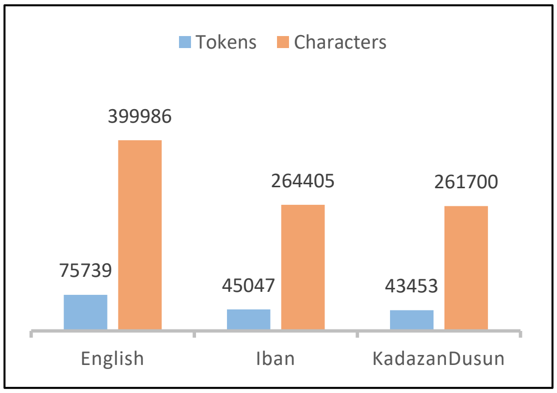


Figure 5 The frequency of tokens and characters for three different language datasets

All tweets are pre-processed by removing tweets that lesser than three words and eliminating English words. Then the tweets are normalised into standard tags, for instance, URLs are replaced with "URLTAG." Normalisation help to reduce the length of the sequence of n-grams when extracted as features due to the tweets instances like URL initially being longer.

4.2 Twitter Native Features

During pre-processing, a normalisation process is carried out to examine the importance of Twitter native features in enhancing the performance of AI for short OSN text. Normalisation substitutes sparse characters such as numbers, dates, and times in the tweets into standard tags. It helps to reduce the number of dimensions by replacing the original sparse characters with standard tags without eradicating the information. Besides, Twitter's native features, i.e., hashtags, user references, and URLs, are also replaced with standard tags. Thus, the datasets are manipulated during pre-processing to prepare three sub-datasets which comprise of:

i. *S+N* (stylometry + native features) dataset – where Twitter native features are included in tweets as standard tags (presence of native features along with text data)
ii. *S* (stylometry) dataset – where all native features are removed in the tweets (absence of native features)
iii. *N (*native feature) dataset – where all words in the tweets are removed, keeping only the native features (absence of text data)

Table 4 shows text examples in three sub-datasets created to examine the importance of Twitter's native features in AI for SMF.

Table 4 Profane words of English, Iban, and Kadazan-Dusun

Original tweet	"Yeah, no kidding! https://t.co/YGDcEreqS6"
***S+N* processed**	"Yeah, no kidding! URLTAG"
***S* processed**	"Yeah, no kidding!"
***N* processed**	"URLTAG"

4.3 N-grams as Features

The importance of identifying the relevant stylistic features for describing an author's signature or writing style is critical to resolving the authorship dispute. Stylistic features based on n-grams were proposed as the most accurate in attributing authors of OSN short text. Thus, this paper examined three different levels of n-grams as the stylistic features in attributing the writing style for short OSN text. Word, character, POS (Part-of-Speech), and the mixture of all three n-grams are extracted from the tweets.

4.3.1 Word N-grams

Word n-grams consist of groups of one, two, or more sequences of words capable of capturing semantically meaningful information from a text in the form of a sequence of words. The limited length of text in tweets encourages users to be more sensible in their writing. Thus, it is reasonable to anticipate that the authors will use only the most concise vocab to write a tweet. Punctuation sequences also are included as they might be a significant component of the phrase. Word unigrams, bigrams, 3-grams, and (1-5)-grams are explored to analyse their impact on distinguishing the author's writing style in OSN text. All word n-grams are case-sensitive, and NLTK, *TweetTokenizer,* is deployed in tokenizing the tweets into words. *TweetTokenizer* is used because it was designed to analyse tweets, unlike other off-the-shelf NTLK tokenizers, thus producing better AI performance for OSN text.

4.3.2 Character N-grams

Apart from being language-independent features, character n-grams need minimal text processing and have proven very useful in AI. Character n-grams can capture unusual features in OSN text, such as emoticons (e.g., "=)," ";P"), unique use of punctuation (e.g., "!!!!," "@__@"), and Internet jargon (e.g., "LOL," "zzz"). Also, character n-grams can capture subtleties in style, hints of context, and handle noise. In the experiment, characters 3-grams and 4-grams are included as features. Sklearn, *Tfidfvectorizor* is used with the parameter, *analyzer* = '*char_wb*' to split the character n-grams from the text inside the word boundaries.

4.3.3 POS N-grams

POS N-grams are considered simple stylistic features related to the syntactic structure of texts. POS n-grams efficiently handle noise in stylistic measures that can affect the AI performance favorably. An off-the-shelf NLTK POS tagger is used to tag the text. There are 36 tags in Penn Treebank, and it is used to tag the tweets. POS 3-grams and 4-grams are extracted as features in attributing the author of short OSN text.

4.3.4 Different Combinations of N-grams

The combination of word, character, and POS n-grams is explored to observe the impact of mixed stylistic features in attributing the authors of tweets. The combinations involve concatenating features from each level of n-grams. Several tests will be run by mixing various degrees of character n-grams, POS n-grams, and word n-grams. After that, the best combination of the n-grams will be chosen based on the highest accuracy among the combinations.

4.4 Baseline Models

The proposed deep learning model of n-grams embedding with CNN is compared with three SoA machine learning algorithms, namely, Naïve Bayes, Random Forest, and Support Vector Machines (SVM). The algorithms are used to learn Tf-Idf n-grams and embedding n-grams models and compare the accuracy.

4.4.1 Naïve Bayes

Naïve Bayes constructs a probabilistic model for each authorship class from training data, y. The probabilities of features, $P(x_i)$, are computed by multiplying all the features to give the probability of test data. The highest probability, $\hat{y}$, among all authors (classes) is the most plausible author of the test data or anonymous tweet. The following equation shows the classification rule of Naïve Bayes [55]:

$$\hat{y} = argP(y) \prod_{i=1}^{n} \quad P(y)$$

Naïve Bayes methods may differ by the assumption they make regarding the distribution of $P(y)$. Multinomial Naïve Bayes is implemented [56]. The classifier lets us know that each $P(y)$ is a multinomial distribution.

4.4.2 Random Forest

A random forest classifier is an ensemble of many classification trees. It comprises a set of decision trees; each tree is trained using random subsets of features. Each tree consists of three types of structures: - leaves, interior nodes, and branches. The leaves of the tree represent the classes (authors), the interior nodes represent the features (vocabulary), and the branches represent the values held by the feature (which in this case is the Tf-Idf value).

To classify a tweet, the author of that tweet is tracked from the tree root until it reaches the leaf where the class is located. The construction of a tree is selected based on the nodes and branches that produce the best split. Although Random forest is less impacted by noise and reduces overfitting, the classifier requires more computational power and resources as it produces many trees. Additionally, the Random forest classifier also demands a longer time in training data as it generates many trees.

4.4.3 Support Vector Machines

SVC (C-Support Vector Classification) is used for classifying the tweets where the implementation is based on LibSVM [56]. LibSVM solves the multi-class problem; that is why it is used to handle multi-classes author identification problems. The multi-class support is handled according to a one-vs-one strategy. This strategy constructs a classifier for each pair of classes.

LibSVM is a faster implementation as it applies techniques such as caching and shrinking. Caching means that the earlier computed values are stored in memory so that re-computation is unneeded. While shrieking technique temporarily eliminates variables that have reached a predefined lower or upper bound. Therefore, these values are not used in subsequent computations.

When training an SVM classifier, several parameters can be set. One of the essential parameters in SVM is the type of kernel used. SVM kernel is a function that takes low dimensional input space and transforms it into higher dimensional space. It helps in converting the inseparable problem into a separable problem. A linear kernel is set for the experiment as this kernel is preferable when the number of features is extensive.

4.5 N-grams as Features

For training and testing, the *k-fold* cross-validation technique is implemented. The *k* cross-validation is chosen because the classification problem has limited data to work with; it can be challenging to provide enough data for disjoint training and testing sets. In the forensic context, only a relatively small amount of data is available to generate an authorship model. The *k* cross-validation provides more meaningful results using whole data in the dataset as both training and testing data.

5.0 RESULT AND DISCUSSION

5.1 N-grams as Features

This experiment aims to determine the significance of incorporating Twitter-native features in AI on short OSN text. The results are tabulated in Table 5, whereby the experiment is conducted by comparing the results of three sub-datasets, *S+N*, *S,* and *N* dataset to observe the impact of Twitter native features on the accuracy of AI system.

Table 5 The result of three sub-datasets of tweets comprising the absence and presence of Twitter native

		English		
	NB	***RF***	***SVM***	***CNN***
N	28.95	33.85	36.3	27.02
S	56.2	58.9	64.8	74.53
S+N	64.7	67.15	72.9	**92.28**
		KadazanDusun		
N	30.01	33.18	32.09	31.41
S	69.48	59.96	69.94	86.65
S+N	67.68	60.28	69.92	**88.68**

Table 4 shows that English and *KadazanDusun* datasets performed well with the presence of Twitter native features in the authors' writing via the *S+N* dataset. All classifiers produced the highest accuracy for English *S+N* dataset, while two out of four classifiers, Random Forest and CNN, excel in classifying the *KadazanDusun* in *S+N*. SVM has a comparative result between

KadazanDusun S and *S+N* datasets with only a 0.02% difference. The *N* dataset for both languages resulted the worst. Low accuracy in *N* dataset is due to the scarce information in the author's writing style by including only the standard tags of Twitter native features.

In contrast, the *S* dataset containing the text with the absence of Twitter native features resulted in average performance compared to *N* and *S+N* datasets that contain Twitter native features. The *S+N* dataset performance improved as Twitter native features play a crucial role by adding information to the authors' writing style. However, when the features were excluded, the accuracy dropped gradually. The result in Table 4 can be concluded that OSN native features play a vital role in accelerating the AI system performance on the OSN text. The rest of the experiments are done with the *S+N* dataset, which will be elaborated in the following subsections.

5.2 Baseline Results based on Tf-Idf and Machine Learning

This section discusses the performance of AI based on different n-grams levels to evaluate the usefulness of the n-grams. Tf-Idf representation and machine learning algorithm as classifier act as the baseline model to learn the n-grams. Naïve Bayes, RandomForest, and SVM were selected as classifiers as they are robust with sparse data and can optimally handle vectors with high dimensionality. Three language datasets, English, Iban, and Kadazan-Dusun were used to evaluate the baseline models. The results consist of word, character, and POS n-grams. Mix n-grams were constructed by combining individual n-grams. Table 6 illustrates the accuracy of the Tf-Idf n-grams, using Naïve Bayes, Random Forest, and SVM as classifiers on three language datasets.

*** Referring to the rest of the tables below, Word n-grams represents as capital W; Character n-grams as capital C; POS n-grams as capital P; English represent as capital E, Iban as capital I, and KadazanDusun as capital K.*

Table 6 The result of three sub-datasets of tweets comprising the absence and presence of Twitter native

	Naïve Bayes			Random Forest			SVM		
	E	*I*	*K*	*E*	*I*	*K*	*E*	*I*	*K*
Characters N-grams									
C3	60.18	62.92	62.42	**59.38**	57.00	56.46	**72.12**	67.42	66.64
C4	63.60	**65.08**	62.74	59.20	57.72	55.12	70.64	69.66	67.36
POS N-grams									
P3	**38.32**	23.56	27.00	38.16	25.32	26.54	40.18	27.16	27.64
P4	34.70	25.02	27.52	32.12	25.28	26.78	35.36	26.44	27.52
P{1-3}	37.16	22.18	26.80	**42.72**	27.20	28.28	**45.24**	29.26	28.84
Mix N-grams									
W1+C4+P4	**67.82**	66.62	66.40	**64.18**	58.72	57.42	74.82	68.82	67.94
W1+C4+P{1-3}	62.74	67.24	65.62	63.52	58.82	56.94	**75.36**	71.28	69.38
Word N-grams									
W1	**69.64**	69.40	68.94	**66.76**	56.98	56.88	**73.3**	68.38	66.20
W2	61.62	43.34	38.24	49.96	30.20	29.12	61.26	43.46	40.64
W3	41.66	17.82	19.98	27.7	16.14	22.50	43.48	20.66	24.68
W{1-5}	69.08	67.60	65.72	63.00	54.18	54.22	70.26	65.42	64.54

The performance of n-grams will be discussed by level (ref. Table 6). At the word level, the accuracy dropped sharply when the value of *n* increased from 1 to three. Word unigram and word {1-5} yielded a comparative result, but unigram outperformed in all three classifiers. Although the accuracy is quite similar, word {1:5} n-grams produce a sparser vector which is not practical in terms of space and processing time compared to word unigram. The larger the matrix, the more time taken for the classifier to learn the vectors. Compared to the U-RL datasets, English has better accuracy because it has more text and vocab (refer Fig. 5), which means it has a greater number of features. Among the classifiers, SVM yielded the best results in all three datasets.

At the character level, character or char 4-grams showed the best individual n-grams features, followed closely by char 3-grams features. Table 5 shows that char 3-grams offer a competitive accuracy, where char 3-grams through Random Forest and SVM produce the highest accuracy for English dataset. As for the U-RL datasets, char 4-grams is preferable in yielding better accuracy than char 3-grams.

At the POS level, the result shows a fair accuracy with an average of 30% accuracy, where the highest is 45.24% and POS {1-3} gram as features. It can be said that POS n-grams are incompetent individually to discriminate the authors of short OSN text, but it is useful when combined with other feature sets through Random Forest and SVM classifier, especially on the U-RL datasets.

As for mix n-grams, two types of mix n-grams are assessed due to competitive results performed by POS 4-grams and POS {1-3}. Both mix n-grams show competing results with a slight increase by the W1+C4+P4 feature set. In terms of efficiency, the aforementioned mix n-grams set is favorable due to lesser matrix size than the later mix n-grams.

Overall, for U-RL datasets, word unigram was learned well by Naïve Bayes. While mix n-grams performed better using Random Forest and SVM. The strategy of using n-grams covering word, character, and POS level offers a satisfactory result on U-RL datasets, and the result is comparable to English datasets because n-grams are language-independent and require less NLP processing. From the result, n-grams were proven to perform well in attributing authors of short OSN text not only in an established language like English but also in U-RL datasets.

5.3 The Effectiveness of Embedding-Based Method in AI for SMF

Due to the sparsity issue by Tf-Idf, the embedding-based method was evaluated using machine learning classifiers to observe the effectiveness of representing the n-grams in dense and distributed form. Using the best results of n-grams (W1, C4, P{1-3} and a combination of them) from the previous experiment, embedding n-grams with fixed 300-dimensions were learned by Naïve Bayes, Random Forest, and SVM on the U-RL dataset, and the result was plotted in Figure 6.

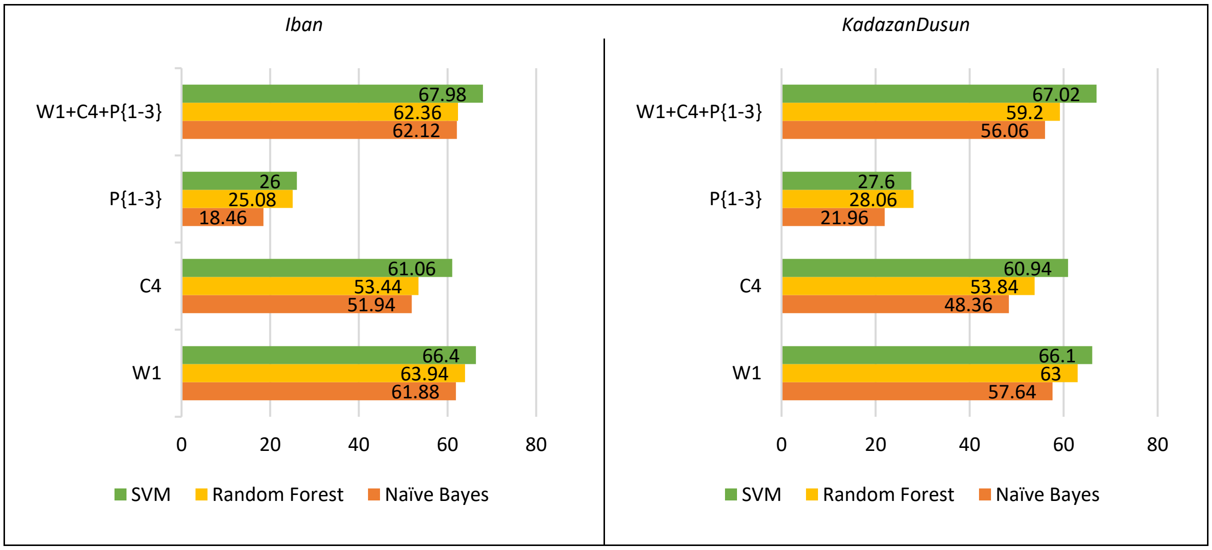


Figure 6 Embedding-based n-grams result using machine learning algorithms on U-RL datasets

The result presented in Figure 6 reveals that the embedding n-grams are in good agreement with SVM on both U-RL datasets compared to Random Forest and Naïve Bayes. The embedding of mix n-grams (W1+C4+P{1-3}) with SVM produced the best result for both datasets with an average of 67%. Word unigrams excel as the individual n-grams feature compared to the character and POS n-grams, likewise Tf-Idf representation, but with lower accuracy. Embedding n-grams learned by machine learning algorithms is relatively lower compared to Tf-Idf n-grams as shown in Figure 6.

Figure 7 shows the comparison of Tf-Idf and embedding mix n-grams learned by SVM. The graph depicted a big difference in accuracy when machine learning algorithms were used to learn Tf-Idf n-grams representation where embedding n-grams are less accurate. Even though embedding representations are known to be distributed and yield dense vectors, machine learning algorithms seem incompatible to learn the embeddings representation well. From the result, it seems that machine learning classifiers performed well in learning the Tf-Idf representation as they accustomed in handling sparse vectors.

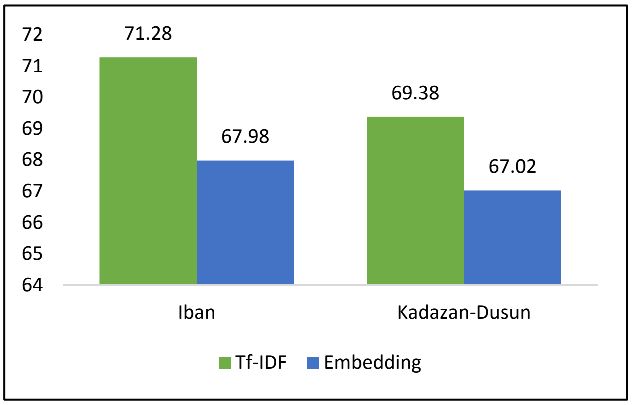


Figure 7 Comparing text representation methods on U-RL datasets

On the other side, embeddings are well known with deep learning where pre-trained embeddings techniques like Word2Vec and GLove employed deep learning to predict the surrounding words [57]. Previous studies proved that embedding representation is notably successful using deep learning models [18, 22, 56]. Compared to machine learning classifiers, deep learning tend to perform better in learning the embeddings.

5.4 A Proposed Deep Learning Model for Short OSN Text on U-RL Dataset using Mix Embedding N-grams and CNN

The proposed model of mix embedding n-grams (W1+C4+P{1-3}) with CNN was evaluated against three other baseline models including mix Tf-Idf n-grams with SVM, mix embedding n-grams with SVM, and Shrestha *et al.,* (2017) which used character n-grams with CNN [18]. The proposed model and baselines were evaluated on the language datasets, and the results were compared, as depicted in Figure 8 and Figure 9.

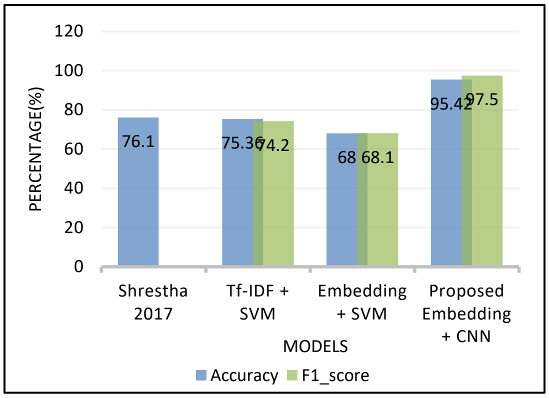


Figure 8 Accuracy comparison between different AI models for OSN short text using mix n-grams as features

Figure 8 shows the result of the proposed model and baselines in terms of accuracy and F1-score. The proposed embedding mix n-grams with CNN (Embedding + CNN) model worked exceptionally well and outperformed the baselines on the English tweets dataset. The proposed model using combinations of embeddings (W1+C4+P{1-3} n-grams) improves the previous work [18] which used embeddings of character n-gams by 19%. Based on the findings, it appears that embedding representation performs significantly better with deep learning than machine learning. The findings also demonstrate that combining n-grams of word, character, and POS can improve AI system performance for short OSN text instead of individual n-grams like characters.

The proposed model is then evaluated on U-RL tweets datasets to observe how the proposed model performed in the U-RL datasets compared to English. Figure 9 shows the accuracy comparison of the proposed model on the established (English) and U-RL (*Iban* and *KadazanDusun*) tweets.

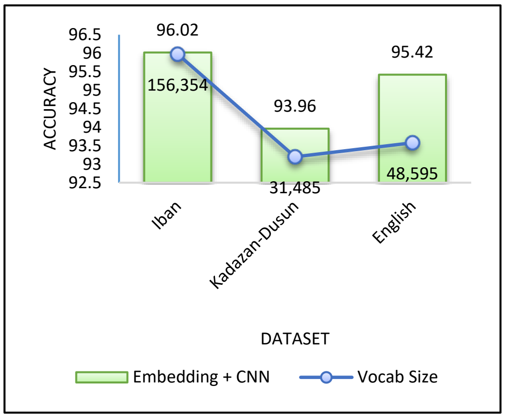


Figure 9 Proposed AI Accuracy vs Vocab Size for different datasets

Figure 9 illustrates the accuracy of the proposed model, CNN mix n-grams embedding, as well as the vocabulary size of the n-grams of three distinct language datasets. The result reveals that the *Iban* dataset outperformed English and *KadazanDusun* because its vocabulary size is bigger than the later datasets. When a dataset has a lot of vocab, the more features it has, the more accurate the model is.

The findings indicate that the vocabulary size also influences the performance of the AI model for short OSN text. Furthermore, the proposed model was observed to be capable of producing consistent performance not only on the established language but also on U-RL tweets. This finding is consistent with previous work [19], where the study suggested that a combination of n-grams feature sets will lead to better performance in attributing authorship for short OSN text.

6.0 CONCLUSION

In recent years, information security, copyright disputes, and public safety have increased the importance of Authorship Identification (AI). This paper presents the evaluation of a proposed AI model for SMF evaluated using U-RL and English tweets as datasets. The evaluation comprises comparing two text representation methods and machine learning against deep learning classifiers in classifying authors of tweets. In addition, this study investigates the effect of Twitter's natural features in facilitating the author's identification of OSN content and the effectiveness of different n-grams features as the stylistic features of OSN text. The finding suggested that Twitter native features are essential in boosting the AI accuracy for short OSN text like tweets. Dense representation appears to be more adaptable to deep learning than machine learning. The proposed model, CNN with mix n-grams embedding, performed exceptionally well on the U-RL datasets containing a large number of n-grams. Based on the findings, combinations of different n-grams levels yield better results than singleton n-grams. In the future, different embedding methods will be explored, such as training the data using Word2Vec or GLoVE method before classification to make better predictions in AI for OSN short text.

Acknowledgement

I would like to express my sincere gratitude to our supervisor, Dr. Suhaila, for her valuable guidance and support throughout the research process. Her expertise and insights was invaluable in shaping my research and helping me to overcome challenges.

References

[1] Ghazali, A.H.A., Abdullah, H., Omar, S.Z., Ahmad, A., Samah, A.A., Ramli, S.A. and Shaffril, H.A.M., 2017. Malaysian youth perception on cyberbullying: The qualitative perspective. International Journal of Academic Research in Business and Social Sciences, 7: 87-98. DOI:10.6007/IJARBSS/v7-i4/2782

[2] Mohammad, N. 2021. Let's Put A Stop To Cyber Bullying, The Faceless Beast, [Online]. Available: https://www.bernama.com/en/thoughts/news.php?id=1979465. Accessed: Aug 2022

[3] Cook. S. 2021. Cyberbullying facts and statistics for 2018 - 2021, [Online]. Available:.https://www.comparitech.com/internet-providers/cyberbullyingstatistics/. Accessed: Aug 2022

[4] The Star. 2022. Malaysia is 2nd in Asia for youth cyberbullying, [Online]. Available: https://www.thestar.com.my/news/nation/2022/01/14/malaysia-is2nd-in-asia-for-youth-cyberbullying. Accessed: Aug 2022

[5] Jiexun L., R. Zheng, and H. Chen. 2006. From Fingerprint to Writeprint. Communications of the ACM, 49(4): 76-82. DOI: https://doi.org/10.1145/1121949.1121951

[6] P. Juola. 2008. Authorship attribution. Foundations and Trends in Information Retrieval. 1(3): 233-334. DOI: https://doi.org/10.1561/1500000005

[7] Zhang, C., X. Wu, Z. Niu, and W. Ding. 2014. Authorship identification from unstructured texts. Knowledge-Based Systems, 66: 99-111. DOI: https://doi.org/10.1016/j.knosys.2014.04.025

[8] Stamatatos,E. 2008. Author Identification: Using Text Sampling to Handle the Class Imbalance Problem. Information Processing and Management. 44(2): 790-799. DOI: https://doi.org/10.1016/j.ipm.2007.05.012

[9] Ghazali, K. 2012. National Identity and Minority Languages. UN Chronicle, 47(3): 17-20. DOI: https://doi.org/10.18356/f3ee6e9c-en

[10] Omar, A. 2014. Processing Malaysian Indigenous Languages: A Focus on Phonology and Grammar. Open Journal of Modern Linguistis. 4(5): 728-738. DOI: https://doi.org/10.4236/ojml.2014.45063

[11] Tajuddin, M. S. A. 2019. Permanent Mission of Malaysia to the United Nations High-Level Plenary Meeting of the United Nations General Assembly on the Global Launch of the International Year of Indigenous Languages. [Online]. Available:https://www.kln.gov.my/web/usa_un-new-york/home/-/asset_publisher/ZJfQEzYEsqRQ/blog/statement-by-mr-mohdsuhaimi-ahmad-tajuddin-charge-d-affaires-permanent-mission-ofmalaysia-to-the-united-nations-high-level-plenary-meeting-of-theu?inheritRedirect=false. Accessed: Aug 2022.

[12] UNESCO. 2021. The International Year of Indigenous Languages: mobilizing the international community to preserve, revitalize and promote indigenous languages. 82-83. ISBN :978-92-3-100484-1.

[13] Igawa, R. A., A. M. G. d. Almeida, B. B. Zarpelao, and S. Barbon. 2015. Recognition of Compromised Accounts on Twitter. SBSI 2015 Proceedings of the annual conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective.1: 9-14. DOI:10.5753/sbsi.2015.5885

[14] Banga, R., and P. Mehndiratta. 2017. Authorship attribution for textual data on online social networks. 2017 Tenth International Conference on Contemporary Computing (IC3). 1-7. DOI : https://doi.org/10.1109/IC3.2017.8284311

[15] Fourkioti, O., S. Symeonidis, and A. Arampatzis.2019. Language Models and Fusion for Authorship Attribution. Information Processing & Management. 56(6): 102061.DOI : https://doi.org/10.1016/j.ipm.2019.102061

[16] Theophilo, A., R. Giot, and A. Rocha. 2021. Authorship Attribution of Social Media Messages. IEEE Transactions on Computational Social Systems. 10(1): 1-14, 2021. DOI: https://doi.org/10.1109/TCSS.2021.3123895

[17] Posadas-Durán, J.P., H. Gómez-Adorno, G. Sidorov, I. Batyrshin, D. Pinto, and L. Chanona-Hernández. 2017. Application of the distributed document representation in the authorship attribution task for small corpora. Soft Computing. 21: 627-639. DOI: https://doi.org/10.1007/s00500-016-2446-x

[18] Shrestha, P., S. Sierra, F. A. González, P. Rosso, M. Montes-y-Gómez, and T. Solorio. Convolutional Neural Networks for Authorship Attribution of Short Texts. 15th Conference of the European Chapter of the Association for Computational Linguistics, Spain. 669–674. DOI : https://doi.org/10.18653/v1/E17-2106

[19] Rocha, A., W. J. Scheirer, C. W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A. R. B. Carvalho, and E. Stamatatos. 2017. Authorship Attribution for Social Media Forensics. IEEE Transactions on Information Forensics and Security. 2(1): 5-33. DOI: https://doi.org/10.1109/TIFS.2016.2603960

[20] Jambi, K. M., I. H. Khan, M. A. Siddiqui and S. O. Alhaj. 2021. Towards Authorship Attribution in Arabic Short-Microblog Text. IEEE Access. 9: 128506-128520. DOI: https://doi.org/10.1109/ACCESS.2021.3112624

[21] Chen, Y. 2015. Convolutional Neural Network for Sentence Classification. Thesis (Master), University of Waterloo, Ontario. DOI: https://doi.org/10.48550/arXiv.1408.5882

[22] Khatun, A., A. Rahman, M. S. Islam and Marium-E-Jannat. 2020. Authorship Attribution in Bangla literature using Character-Level CNN. 2019 22nd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh. 1-5. DOI: 10.1109/ICCIT48885.2019.9038560.

[23] Dreher, J. J. 1970. The Computer-Linguistic Detective of Authorship. The Journal of Asian Studies. 29(4): 883-887. DOI: https://doi.org/10.2307/2943094

[24] Mendenhall, T. C. 1887. The Characteristic Curves of Composition. American Association for the Advancement of Science. 9(214): 237-249.DOI: https://doi.org/10.1126/science.ns-9.214S.237

[25] Mendenhall, T. C. 1901. A menchanical solution of a literary problem. Popular Science Monthly. 60: 97-105.

[26] Yule, G. U. 1939. On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. Biometrika. 30(3/4): 363-390. DOI: https://doi.org/10.1093/biomet/30.3-4.363

[27] Ellegård, A. 1962. A statistical method for determining authorship: the junius letter. Gothenburg, Sweden: Acta Universitatis Gothoburgensis. 1769-1772. DOI:10.2307/411928

[28] Sarndal, C. E. 1967. On Deciding Cases of Disputed Authorship. Journal of the Royal Statistical Society. 16(3): 251-268. DOI: https://doi.org/10.2307/2985921

[29] Morton, A. Q. 1978. Literary Detection : How to Prove Authorship and Fraud in Literature and Documents. New York: Scribner.

[30] Bailey, R. W. 1978. Authorship Attribution in a Forensic Setting. Advances in Computer-aided Literary and Linguistic Research: 87-106.

[31] Burrows, J. F. 1987. Word-patterns and story-shapes: The statistical analysis of narrative style. Literary & Linguistic Computing. 2(2): 61-70. DOI: https://doi.org/10.1093/llc/2.2.61

[32] Burrows, J. F. 1989. An ocean where each kind...': Statistical analysis and some major determinants of literary style. Computers and the Humanities. 23: 309-321. DOI: https://doi.org/10.1007/BF02176636

[33] Holmes, D. I. 1998. The Evolution of Stylometry in Humanities Scholarship. Literary and Linguistic Computing. 13(3): 111-117. DOI: https://doi.org/10.1093/llc/13.3.111

[34] Burrows, J. F. 1992. Not unles you ask nicely: The interpretative nexus between analysis and information. Literary and Linguistic Computing. 7(2): 91-109. DOI: https://doi.org/10.1093/llc/7.2.91

[35] Greenwood, H. H. 1995. Common word frequencies and authorship in Luke's Gospel and Acts. Literary and linguistic computing. 10(3): 183-187. DOI: https://doi.org/10.1093/llc/10.3.183

[36] Holmes , D. I. and R. S. Forsyth. 1995. The Federalist revisited: New directions in authorship attribution. Literary and Linguistic Computing. 10(2): 111-127. DOI: https://doi.org/10.1093/llc/10.2.111

[37] Mealand, D. L. 1995. Correspondence analysis of Luke. Literary and linguistic computing. 10(3): 171-182. DOI: https://doi.org/10.1093/llc/10.3.171

[38] Kjell, B. 1994. Authorship determination using letter pair frequency features with neural network classifiers. Literary and Linguistic Computing. 9(2): 119-124. DOI: https://doi.org/10.1093/llc/9.2.119

[39] Hoorn, J. F., S. L. Frank, W. Kowalczyk, and F. v. d. Ham. 1999. Neural network identification of poets using letter sequences. Literary and Linguistic Computing. 14(3): 311-338. DOI : https://doi.org/10.1093/llc/14.3.311

[40] Argamon, S., M. Šarić, and S. S. Stein. 2003. Style mining of electronic messages for multiple authorship discrimination: first results. Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York. 475–480. DOI : https://doi.org/10.1145/956750.956805

[41] Kešelj, V., F. Peng, N. Cercone, and C. Thomas. 2003. N-gram-based author profiles for authorship attribution. Proceedings of the conference pacific association for computational linguistics (PACLING), Nova Scotia. 3: 255-264.

[42] Abbasi A. and H. Chen. Applying authorship analysis to extremist-gro up web forum messages. IEEE Intelligent Systems, 20(5): 67-75. DOI : https://doi.org/10.1109/MIS.2005.81

[43] Zheng, R., J. Li, H. Chen, and Z. Huang. 2006. A framework for authorship identification of online messages: Writing-style features and classification techniques. Journal of the Association for Information Science and Technology. 57(3): 378-393. DOI : https://doi.org/10.1002/asi.20316

[44] Potthast, M., P. Rosso, E. Stamatatos, and B. Stein. 2019. A Decade of Shared Tasks in Digital Text Forensics at PAN. In Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany. Part II 41: 291-300. DOI: https://doi.org/10.1007/978-3-030-15719-7_39

[45] Almishari, M., E. Oguz and G. Tsudik. 2014. Fighting Authorship Linkability with Crowdsourcing. In Proceedings of the second ACM conference on Online social networks. 69-82. DOI: https://doi.org/10.1145/2660460.2660486

[46] Neocleous, A. and A. Loizides. 2021. Machine Learning and Feature Selection for Authorship Attribution: The Case of Mill, Taylor Mill and Taylor, in the Nineteenth Century. IEEE Access. 9: 7143-7151. DOI: https://doi.org/10.1109/ACCESS.2020.3047583

[47] Barlas, G. and E. Stamatatos. 2021. A transfer learning approach to cross domain authorship attribution. Evolving Systems. 12(3): 625-643. DOI: https://doi.org/10.1007/s12530-021-09377-2

[48] Huang, W., R. Su, and M. Iwaihara. 2020. Contribution of improved character embedding and latent posting styles to authorship attribution of short texts. In Web and Big Data: 4th International Joint Conference, APWeb-WAIM 2020, Tianjin, China. Part II 4: 261-269. DOI : https://doi.org/10.1007/978-3-030-60290-1_20

[49] Theóphilo, A., L. A. Pereira, and A. Rocha. Needle in a haystack? harnessing onomatopoeia and user-specific stylometrics for authorship attribution of micro-messages. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal

Processing (ICASSP), Brighton. 2692-2696. DOI: 10.1109/ICASSP.2019.8683747. DOI: https://doi.org/10.1109/ICASSP.2019.8683747

[50] Le, Q. and T. Mikolov. 2014. Distributed Representations of Sentences and Documents. In International conference on machine learning. DOI: https://doi.org/10.48550/arXiv.1405.4053

[51] Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov. 2017. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics. 5: 135-146. DOI: https://doi.org/10.1162/tacl_a_00051

[52] Besacier, L., E. Barnard, A. Karpov, and T. Schultz. 2014. Automatic Speech Recognition for Under-Resourced Languages: A Survey. Speech Communication. 56: 85-100. DOI: https://doi.org/10.1016/j.specom.2013.07.008

[53] Apin, P. and K. A. Wahab. Tabu bahasa dalam masyarakat Dusun di Daerah Ranau, Sabah. Jurnal Melayu. 14(2): 224-239.

[54] Kowsari, K., K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown. 2019. Text Classification Algorithms: A Survey. Information. 10(4): 150. DOI: https://doi.org/10.3390/info10040150

[55] Savoy, J. 2020. Machine Learning Models. Machine Learning Methods for Stylometry: Authorship Attribution and Author Profiling. Chapter 6: 109-151. DOI: https://doi.org/10.1007/978-3-030-53360-1_6

[56] Wenjing, R. S., Huang, and M. Iwaihara. 2020. Contribution of Improved Character Embedding and Latent Posting Styles to Authorship Attribution of Short Texts. In Web and Big Data: 4th International Joint Conference, APWeb-WAIM 2020, Tianjin, China. Part II 4: 261-269. DOI: https://doi.org/10.1007/978-3-030-60290-1_20

[57] Chowdhury, H. A., Imon, M. A. H., and Islam, M. S. 2018. A comparative analysis of word embedding representations in authorship attribution of bengali literature. In 2018 21st international conference of computer and information technology (ICCIT). 1-6. DOI: 10.1109/ICCITECHN.2018.8631977