Exploring Mechanisms for Detecting Violent Content in Sinhala Image Posts: Rationale with Unsupervised vs Supervised Techniques

U Dikwatta[1#], TGI Fernando[1], and MKA Ariyaratne[2]
[1] Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka
[2] Faculty of Information Technology and Communication Sciences, Tampere University, Finland

[#]umanda@sjp.ac.lk

ABSTRACT This research explores the different avenues in machine learning to classify Sinhala image posts. Image posts in social media are one big weapon that conveys information directly to people. Image posts contain both visuals and text. English based research work is common in this regard, but only a handful can be seen from other languages. The target language was a low-resource language, Sinhala. Unsupervised algorithms were used to classify image posts and supervised algorithms were involved classifying manually extracted text in image posts. The classification decides whether the posts are violent or nonviolent. The trained supervised models were tested with interpretability models to identify the words that cause the decision of violent or nonviolent. The findings reveal supervised algorithms perform better than unsupervised algorithms in classifying image posts. However, improved results can be obtained by increasing the size and the variety of the dataset.

INDEX TERMS Deep learning, machine learning, social media, violence detection

I. INTRODUCTION

People use social media as a powerful tool for communication. The birth of 2-way communication began with Web 2.0 and has been evolved, so now people who use social media can modify the contents as well as provide their thoughts towards enormous topics [30]. As with useful and entertaining content, social media also provides a platform for users who spread violent content that disperses violence to the physical world. One of the earliest examples of such behaviour can be pointed out through a case study done in 1997, which was based on an incident in Bangladesh where violence originated in social media [42]. Also in 2008, image posts about the resentment of immigrants in Italy were circulated through social media [45].

As images can talk to people way faster than the words, image posts have become very popular in social media to convey ideas; to spread good as well as the opposite. Hence, image posts shared on social media have become a powerful mechanism to disperse violence. They contain visual and text elements. One of the main concerns even from the early days of social media is to identify smart mechanisms for early detection of such poor posts and help to clean the social media platforms from such contents. Based on such concerns, there are many research attempts. However, most of such findings are based around image posts having text elements in the English language. A handful of research can be found from other languages, and we have listed them in section 2.

In natural language processing, languages are categorized by whether they are high or low resource. Low resource languages lack data that can be used for machine learning (ML) or other processing, and high-resource languages are rich in available data. With the birth of the Unicode character system, usage of law resources languages has accelerated in a noticeable way. That directly affects the usage of such languages in social media as well. Image posts with violence; what we are interested in this research can also be seen with text elements in such low resource languages.

In this work, we aimed to work on a low-resource language and chose ‘Sinhala’ as the preferred low-resource language. We concern text elements inside Sinhala image posts. As the literature revealed most of the text classification research works focused on “Sinhala” were based on Facebook comments or Tweets. In those works, they have used the row text form user comments. This research is different from the input from most of such research. Most of the research on the text classification connects with ML techniques since data are concerned as the main resource at hand in decision making [23]. Going along the same line, this research also focused on using unsupervised and supervised ML techniques to identify the desired image posts.

This research work has mainly four phases. First, we modified the dataset introduced in previous research to include a balanced dataset that provides violent and nonviolent image posts [20]. Then, we retrained the unsupervised models as an anomaly detection problem introduced in earlier research on the new dataset [20]. Next, we used supervised machine learning algorithms such as shallow learning and deep learning (DL) to classify the manually extracted text. Going beyond acquiring traditional training testing accuracies as goodness of measures, as the last phase, in this research we tried using explainable AI processes that allow human users to trust the results and output created by ML algorithms. By following the phases, the aim of the research, which is on exploring the capabilities of supervised and unsupervised machine learning techniques to detect violent context in Sinhala images posted was achieved.

For the convenience of the readers, this paper is arranged as follows. In the next section, we present the literature relevant to the current research. Section III lists used materials and methodologies in our work. In section IV, we present our research results and finally in section V we discuss our findings, present the drawn conclusions, and point out future possibilities.

II. RELATED WORKS

Violence detection in social media encompasses several distinct categories such as modality based, classification algorithm based, and language based. Modality-based violence detection has four distinct categories: text, images, videos, and multi-modal approaches that combine both textual and visual elements. However, this research primarily focuses on social media images, and we have not extensively discussed the techniques related to videos. In addition to the modality-based categorization, violence detection utilizes classification algorithms that employ ML and DL techniques. Furthermore, violence detection encompasses several language-based studies: English, Sinhala, and other low resource languages such as Arabic. The literature includes scholarly studies from the earliest publication in 2014 to the most recent. We present previous work by the language, the research work has been focused, from high level languages to low level languages.

A. Related Research on English Language

Hate speech is one way of spreading violence in social media. Different organizations, communities and social media sites have given different definitions to hate speech. Hate speech can be defined as a set of terms in a defined language that attacks a person or a group of people regarding religion, ethnicity, gender, sexual orientation. These hateful contents can persuade people to violence. Hate speech can be detected in several ways; however, many research studies were based on machine learning techniques as data is incorporated with the process. When it comes to ML, feature extraction is one of the main tasks in the process of decision making. Review works have pointed out, in ML, feature extraction for hate speech detection has been done using several approaches. Bag-of-words (BoW), term frequency - inverse document frequency (TF-IDF), rule based, n-grams, word embeddings, and topic classification methods are some of them. Further, the reviews discussed the contribution of shallow ML algorithms like support vector machines (SVM), naïve Bayesian (NB), logistic regression (LR), decision trees (DT) to the process of hate speech detection. Authors also point out the classification mechanisms used in detecting hate speech using deep learning algorithms. However, the comparisons among methods were not discussed as most of the newly created datasets are not published and publicly available [23, 55]. Anusha Chhabra and Dinesh Kumar Vishwakarma presented another review on multi-modal and multilingual social media hate content detection using shallow and deep learning ML models [10]. The findings of the survey point out; almost all the past reviews were conducted covering text-based hate speech detection studies, only two datasets were identified as multi-modal: text and image based, and DL approaches have outperformed shallow learning approaches.

Going deeper into actual works performed for the English language, one of the earliest works proposed a mechanism using paragraph2vec [31] and continuous bag-of-words (CBOW) [39] for on-line user comments in Yahoo Finance website to hate speech detection. LR was used as the classification algorithm. The proposed method has given higher Area Under the Curve (AUC) than existing BoW methods [21].

Tweets are an interesting and powerful communication mechanism among a lot of people. Hate speech is also a frequent content in tweets. In a research study, hate speech in Tweets was investigated as a multi-class problem with three classes: hate (strongest hate level), offensive and neither [15]. Five shallow ML algorithms; LR, linear SVM, NB, DT, random forest (RF) were used to build the models. The results showed that LR and linear SVM performed better than the other three algorithms. L2 regularization (Ridge Regression) combined with LR improved the accuracy of the normal linear regression model. Going beyond the shallow algorithms, in [5], has used deep neural networks (DNN) to detect hate speech in tweets. Convolutional neural network (CNN), long short-term memory (LSTM) and fastText were used as feature spaces. Precision, recall and F1 score were used to compare the results. DNN achieved better results than shallow machine learning methods that utilized Char n-gram, TF-IDF and BoW embeddings. Higher results were obtained for utilizing random embeddings trained with LSTM and using them in a gradient boosted decision trees (GBDT) algorithm for classification.

Another research used English as a language to test the performance of different feature extraction methods combined with Linear SVM model [36]. Character n grams, word n-grams, and skip grams were used as feature extraction methods to detect hate speech. The Character 4-gram method has shown better results than other methods and achieved an accuracy of 78%. However, previous research done by Malmasi et al., claimed better results than the 4-gram method using an oracle ensemble method with SVM [35]. Data is crucial for any machine learning process, so hate speech detection. Won et al.

has formed a protest image dataset [72]. They have employed a ResNet based model to violence detection in images and OpenFace based model [4] for emotion detection of people in violent scenes. They have found their model performs well in identifying violent scenes but does not perform well for emotion detection. As same, Sun et al. have created a new dataset that consists of still images related to violence and nonviolence [60]. They have used low level features as the multi views in their dataset along with features extracted from CNN. Low level features include dense scale invariant feature transform (DSIFT), histogram of oriented gradient (HOG) and local binary pattern (LBP). Authors have proposed new multi view maximum cross entropy discrimination.

Watanabe et. al. introduced a new feature extraction method that incorporates sentiment, semantic, unigrams and pattern feature to identify hate text in Twitter [70]. They have used ‘‘J48graft’’[69], SVM and RF as classifiers. ‘‘J48graft’’ has outperformed the other two classifiers. The new model ‘‘J48graft’’ is an extension of decision tree grafting algorithm that increases the performance of the original algorithm with respect to both bias and variance. Further Z. Zhang and L. Luo addressed the problem of “long trail” in hateful text in social media, specifically in Twitter [73]. This research has proposed a model that incorporates two DNN architectures with CNN and Gated Recurrent Unit (GRU). This method has surpassed state-of-the-art methods on a Twitter dataset and established a new benchmark for future research that involves identifying hate speech.

Gang violence, one of the other types that create textual as well as visual modality, can be defined as criminal and non-political acts of violence committed by a group of people who regularly engage in criminal activity against innocent people. Research was also carried out around this area and multi-modal approaches were used to detect images with gang violence [7]. They used tweets to create the datasets that were annotated with psychosocial codes, aggression, loss, and substance use. Text features were detected using unigram, bigram, Part-of-Speech (POS), and CNN features. Regional-based convolutional neural network (R-CNN) was used to detect image features. Fusion methods: early fusion and late fusion were used as multi-modal feature extraction methods. Text feature classification showed better results for loss code where image features classification showed better results for aggression and substance codes. Fusion method has shown promising results in this research.

Amorim et al. introduced novelty detection in a temporal window using data fusion technique [3]. The objective of this approach is to detect comments that stand out from others within a given time frame considering both present and past comments. The dataset used in this study consists of posts from social media platform Twitter. Architecture comprises three key components: feature extraction from images and text, data fusion and unsupervised algorithm. Two distinct architectures were employed by rearranging the order of three key components. In the first architecture, input to the architecture is a data stream and MASK-RCNN [26] is the data fusion algorithm that converts the stream into textual representation. Then an autoencoder was employed to convert the textual representation into a vector and unsupervised algorithm was employed to classify the vectors. Second architecture, transform tweet images and texts into vectors using an autoencoder. Then an unsupervised algorithm identifies novelties using the vectors of images and texts. Finally, the AOM [1] fusion algorithm was employed to fuse the scores obtained from the unsupervised algorithm. Results depict that MASK-RCNN method outperforms AOM method.
Suryawanshi et al. proposed a novel system to detect offensive memes by leveraging multi-modal data: text and images [63]. Going beyond text and visual datasets, authors have curated a new dataset including memes that contain both text and images. They suggested an early fusion method that incorporates stacked LSTM, BiLSTM and CNN for text features and visual geometry group (VGG-16) for visual features. Results demonstrate that the multi-modal approach has outperformed methods that incorporate only a single mode in terms of precision, F score and recall.

In [41], a Bidirectional Encoder Representations from Transform (BERT)-based transfer learning method has been introduced for hate speech detection. The new method consists of different fine-tuning approaches, adding nonlinear layers, adding Bi-LSTM layers, and adding CNN layers. The fine-tuned BERT-based method has produced better results than other state-of-the-art methods such as character n-gram with LR [67, 15], CBOW with multi-layer perceptron feed forward neural network [68], and original BERT model. Further in a comparative study conducted on hate speech detection using 14 shallow and DL models with three commonly used datasets revealed that BERT-based models outperform other methods, and the TF-IDF-based classifier outperforms other DL models [34]. In another work, authors proposed an ensemble method that employs a combination of a fine-tuned BERT based model and a parallel recurrent model for multi aspect hate speech detection [37]. The proposed model was compared with pooled stacked Bi-LSTM, Bi-GRU models and ensemble models that combine the outputs of BERT, Bi-LSTM, and BI-GRU. The new model yielded better results compared to other methods. In a recent research work, authors have proposed a multi-modal fusion mechanism to combine both text and visual features for classifying fake news [65]. They have obtained a dataset along with their captions. A fine-tuned BERT model was used to classify text and higher results were obtained when compared with other DL models. Fine-tuned Xception network has obtained higher results for visual feature classification. Concatenate fusion techniques have obtained higher results than other fusion techniques. Fusion methods have achieved higher results than using text or visual solely.

To fulfil the lack of data sets containing fight images, authors in [2] have developed a new still fight image dataset collected from social media sites. They have used DL networks like VGG-16, residual network (ResNet50), ResNeXt50, and vision transformers (ViT Large 16) for the classification and ViT network has surpassed the results of other models. Most of the violent scene detection experiments were done for video-based datasets. As the next phase of the research, they have compared the results obtained for temporal models with frame-based models that were trained. Authors have done a cross-dataset experiment to evaluate which model generalizes well with all the datasets. Models that are trained for still images generalize better than the models trained for video-based datasets. We have studied a few but mostly relevant literature which were based on the English language. Compared to English, research work on other languages is limited.

B. Related Research on Sinhala Language

In one of the earliest works that touch Sinhala for the first time, English comments on a Sri Lankan website were investigated for hate speech [52]. NB, SVM, LR, DT, and k-means were tested with BoW and TF-IDF. NB with TF-IDF achieved a better F-score than other methods. In another previous study, researchers successfully identified racist Sinhala comments using a two-class SVM and n-gram approach, achieving over 70% accuracy [19]. The dataset comprised randomly selected Sinhala comments from social media platforms. However, the performance declined as the dataset size increased. Identifying abusive comments in Sinhala language was also tried in research [54]. SVM, Multinomial NB (MNB), and random forest decision tree (RFDT) were used as classifiers. BoW, word n-gram, character n-gram, word skip-gram were the feature extraction mechanisms. MNB showed better results than other classifiers. Character tri-gram and character four-gram showed better results than other feature extraction methods. Corpus-based approaches showed better results.

A multi-level and two-level hate speech classification was done for Sinhala social media comments [53]. Authors have mentioned the difficulty in finding a proper data source for Sinhala. CNN and SVM were used as classification algorithms. CNN has shown higher results for binary classification. SVM has shown higher results for multi-level hate speech classification. According to the authors, a lower F1 score is achieved due to the imbalance dataset.

For the first time in Sinhala language, images were used in [59] to classify Sinhala hate text in images. Several ML techniques were used to model the data. The text has been automatically extracted from images. MNB has shown better precision, recall, and F-measure than other ML techniques.

Adapter-based pre-trained multilingual models have been proposed for code mixed and code-switched text classification that includes Sinhala text [49]. The cross-lingual representation of robustly optimized BERT pre-training approach (XLM-R), with basic fine-tuning, has outperformed all other models. XLM-R with adapters has further improved the results. BERTifying Sinhala is an analysis carried out to evaluate the performance of XLM-R, Language-Agnostic BERT Sentence Embedding (LaBSE), and Language Agnostic SEntence Representations (LASER) in Sinhala text classification [18]. There, XLM-R has performed better than other models.

This summary covers the limited research carried out on violence detection in Sinhala. It highlights the pressing need for further research in low-level languages like Sinhala.

C. Related Research on Other Languages

Arabic, Bengali, Italy can be identified as other languages that have contributed more on hate speech detection research. In reference [43], the authors constructed a dataset for Arabic by gathering data from popular social media networks. They utilized this dataset for hate speech detection purposes. They have performed data filtering to clean the dataset. Dataset was annotated. Then the dataset was trained and tested with ML and DL models. Complement NB surpassed other ML models for accuracy, F1 score, recall and precision. RNN outperformed CNN in DL models. Regarding the previous datasets on Arabic, the dataset collected in this research has given higher accuracy. Further in [44], authors have developed an Arabic dataset for topic classification, sentiment analysis, and multi-label classification of on-line social media networks (OSNs). Removing tokens beyond a specific length, removing stop words and stemming were performed as preprocessing steps. BoW, n-gram, TF-IDF were used as feature extraction methods. Shallow ML algorithms were used. Authors have incorporated grid search to select the best set of hyper parameters. Chi-square feature selection and hyper parameter tuning has improved the results. n-gram (1,2) with linear support vector classification (LinearSVC) has obtained higher results in topic classification. LR with BoW has yielded higher results on sentiment classification while TF-IDF with LinearSVC showed higher results for multi-label classifiers. Authors have also found a relationship between hate speech and OSNs post topics. Their proposed mechanism yielded 83.7% accuracy in filtering Facebook posts.

In another study related to Arabic, proposed a new mechanism to detect contradictions in Arabic sentences, a special scenario of natural language inference (NLI) [29]. Authors have created a dataset consisting of more than 6,000 sentence pairs of Arabic language. Their dataset consists of three different classes: contradiction, entailment and neutral. They augmented the dataset by automatic translation using two existing datasets. Feature extraction models used were word embedding mechanisms and language level feature extraction methods. SVM, stochastic gradient descent (SGD), DT, adaptive boosting (AdaBoost), k-nearest neighbour (KNN) and RF were used as classification methods. They have evaluated the results on their original dataset and two translated datasets. Obtained results convince higher accuracy for RF classification that employs BoW vector with contradiction vector.

Regarding Bengali language, research was conducted to evaluate the performance of multi-class sentiment classification on Bengali text [25]. Authors proposed a system that employs CNN and LSTM architectures. They have built a Bengali text dataset of size 42,036 social media comments that has four different classes. Authors have selected MNB, LR, DT, RF, SGD, and SVC along with their word embedding mechanisms like TF-IDF and count vectorizer (CV). LSTM, Bi-LSTM, Bi-GRU and a model that employs both CNN and LSTM (C-LSTM) were used along with word embedding as DL architectures. C-LSTM has outperformed other baseline methods.

We have summarized the related work in the context of hate speech detection mainly with the involvement of shallow and complex machine learning techniques. For convenience, we categorize our findings language wise. The findings opened the avenues and pointed out the importance of conducting more research on low level languages such as Sinhala, which was tried to achieve in the current research.

III. MATERIALS AND METHODS

Here, we present a detailed description of how our research has been conducted. As the first step of the study, we have composed a dataset that includes Sinhala violent and nonviolent images mainly collected from Facebook. We have employed two approaches to classify the dataset into two categories, nonviolent and violent. The first approach is clustering where images are fed to unsupervised algorithms. The second approach is to utilize manually extracted textual parts of the images to train supervised learning algorithms. To train supervised ML algorithms, the dataset was annotated as nonviolent and violent. To evaluate the results, we employed four metrics commonly used in ML studies: accuracy, precision, recall, and F1-score. One drawback on ML algorithms is that they are like black boxes, and we do not know why a ML model predicts a text as violent or nonviolent, which words in the text caused the decision. To find out which words caused the decision, in this research, we further employed explainable AI (XAI) methods such as local interpretable model-agnostic explanations such as (LIME) [50] and Shapley additive explanations (SHAP) [33] and integrated gradient (IG) [61]. The overall process of supervised learning is depicted in Figure 1.

A. Dataset collection

All images were manually downloaded from Facebook. We found Facebook groups and their pages that are specialized for different topics that are related to our study. We used such pages to download images and we have also used keyword search to download violent posts. We identified commonly used violent words and treated them as keywords. The final dataset consists of 3,463 nonviolent and 3,465 violent images. Figure 2 and Figure 3 depict a nonviolent image and a violent image respectively.

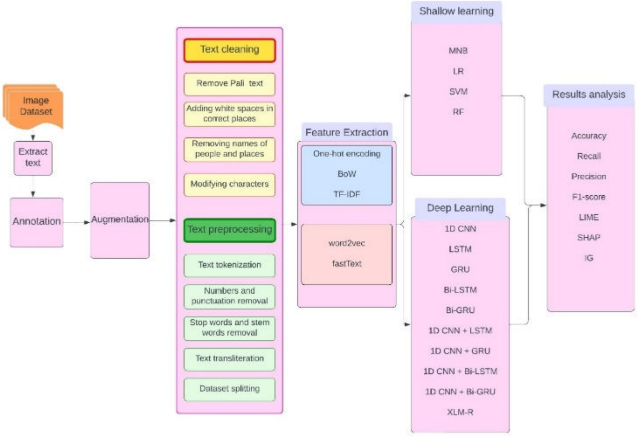


Figure 1. Supervised learning training process

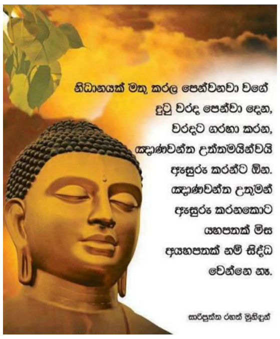


Figure 2. Image nonviolent content

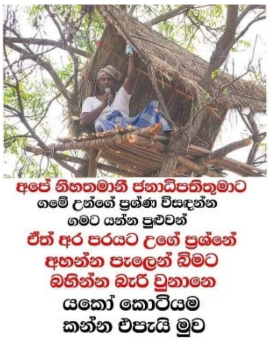


Figure 3. Image violent content

As the study was conducted mainly based on two types of ML algorithms, two types of data preparation were needed. For the unsupervised algorithms, mainly more nonviolent data were collected. For that basically a subset from [20] is used. 2,463 nonviolent image posts were used to train the unsupervised algorithms.

For the supervised algorithms, a different data handling process is employed. For that, a dataset was constructed by combining the data collected from previous research [20] with additional violent image posts. Textual portions of the images were extracted manually to perform the text classification. Unicode characters were utilized during extracting the text parts since Unicode characters provide device and platform-independent characters. For the supervised algorithms, data collection was followed by Data Annotation, Data Augmentation, Data cleaning, and Data Preprocessing.

1) Data/Images Annotation:

Two volunteer annotators annotated the dataset. The annotation process used the guidelines described in previous research [20]. If either the textual or visual component exhibited violence, the post was labelled as violent. A post can be identified as violent if it contains content that abuses a religion, race, or other beliefs; targets individuals or groups causing emotional harm or displays sexism. The posts that contain sarcasm were regarded as violent as it can inflict emotional distress on individuals or groups. Cohen's kappa was calculated to evaluate the agreement between two annotators, and scikit-learn library was used for the Cohen's kappa calculation [12]. The calculated Cohen's kappa value for the dataset was 0.9.

2) Data/ (Images, Text) Augmentation:

To enhance the accuracy of deep learning models, a larger dataset is expected. In the realm of ML, expanding a dataset using the existing samples is referred to as data augmentation [22, 58]. In the context of unsupervised learning our inputs were entire images. Therefore, we performed data augmentation techniques for the images such as random rotation, colour jittering and random horizontal flipping to generate additional variations of the existing images. We used random rotation to rotate images by a random angle, colour jittering to change the brightness, contrast, saturation, and hue of images randomly and horizontal flipping flip images horizontally for a given probability. When applying augmentation to our dataset, we followed a novel technique: we individually applied the three augmentation techniques to our training dataset, then concatenated the resulting datasets with the original dataset and achieved 9,852 as the size of the final training dataset.

For the textual components, a technique called back translation was employed to expand the dataset. 1,000 violent and 1,000 nonviolent texts were randomly selected for back translation. The initially selected text was translated into English and was subsequently translated back into Sinhala. Augmented data was re-evaluated to compare the original text with augmented text. Text that was augmented with a wrong meaning was manually corrected. Python translation was used in the translation process and the resulting text was written to a Microsoft Excel sheet. The augmented text and the text used to create augmented text were included only in the training dataset and for the testing, separate set of text were used. Following the augmentation, we achieved 8,907 as the size of our final dataset.

3) Data/Text Cleaning:

Text cleaning process in this study consists of several steps including removing Pali text, adding white spaces, removing names, and modifying characters. Pali text is a Middle Indo-Aryan language mainly used in Theravada-Buddhism. Buddhist monks in Sri Lanka use this language to chant prayers. Pali text included in most Buddhist posts was removed from the dataset. Usually, punctuations follow a white space in professional writing; however, the standard rules are not followed in most amateur posts. The tokenization process returns a different output when white spaces are not included in the correct places. Therefore, regular expressions were used to make sentences accurate.

In Sinhala alphabet, න, ණ, ල, and ළ are consonants. න and ණ, as well as ල and ළ have the same sound. Although the letters have the same sound, they cannot be used interchangeably. However, people who speak the language use these letters interchangeably due to a lack of knowledge of using them. It is difficult to memorize the places where these letters are used. Hence, these two letters are misused in Sinhala writing and in image posts. That is, න is used in cases where the letter ණ is expected, and vice versa. Similarly, ල is used in cases where the letter ළ is expected and vice versa. Therefore, all the text, including ළ, was modified to ල, and all the text, including ණ, was modified to න.

4) Data/Text Pre-processing:

Data/Text pre-processing is followed by a series of steps such as tokenization, removal of numbers and punctuations, stop words and stem words, text transliteration and dataset splitting.

The first step of text pre-processing is to split the text from white spaces. The split texts are called tokens. The research was conducted with "word_tokenize" in the natural language toolkit (NLTK) and "SinhalaTokenizer" from "sinling" [56].

Numbers and punctuation were removed from the dataset. Stop words and stem words prominently used in the Sinhala language were filtered out. Stop words such as ඒ, මේ, නම්, ඇති, එක, කර, හා, නෑ, වන, වූ, ද, බව, ගැන, කරයි, අතර, යන, ලෙස, නිසා are used. English meaning of these words respectively is "That, this, if, one, done, and, no, is, was, the, that, about, does, between, going, as, because". Stem words such as මම, මට, මටම, මටත්, මගේ, මගේම, මාගේ, මාගෙ, මාව, මාවම, අපි, අපිම, අපිවම, අපට, අපටම, අපව, අපවම, අපේ, අපේම are filtered out. "I, me, myself, mine, my own, we, ourselves, our" are the English translations of stem words.

Python Unidecode function was employed to obtain the transliterated text as a preprocessing step to check any improvement in the performance [74]. Unicode characters are fed into the Unidecode function and converted to ASCII characters.

The training process commences by initially partitioning the dataset into training and testing sets with a 4:1 ratio using the scikit-learn command to split the data [46]. Subsequently, we selected the models with higher results for further testing. In the subsequent step, the dataset was partitioned into training, validation, and testing subsets with 8:1:1 ratio. The selected models underwent further evaluation with the updated dataset partitioning. The PyTorch data loaders were created for training, validation, and testing datasets. For the unsupervised training process, the training set consists only of nonviolent images. A subset of nonviolent images from our new dataset was selected as the training dataset. As for the validation and testing datasets, 500 images from each category were selected.

B. Hardware, software, libraries, and technologies used.

PyTorch was used as the ML library and Jupyter Notebook as the platform. Experiments were conducted on an Nvidia RTX – 3090 64 GB server.

C. Evaluation metrics used in the research

Evaluation metrics used are accuracy, precision, recall, and area under the ROC curve (roc_auc_score) [28]. The confusion matrix is also used.

D. Unsupervised Learning

The dataset was used as it is to be fed into the unsupervised learning algorithms. There are many algorithms under unsupervised category. We have focused our study on autoencoders. Autoencoder is an unsupervised learning algorithm. The autoencoder architecture contains an encoder and a decoder. When an image is fed to the encoder, the decoder will attempt to regenerate the image. The loss function of autoencoders is defined as the difference between the original and the regenerated image (reconstruction loss). Autoencoders are trained using a specific type (nonviolent) of data, allowing them to learn patterns inherent within that dataset. Trained autoencoder can regenerate the type of data it has trained. If the type of data, we used in training is nonviolent then the autoencoder will give a lower reconstruction loss for nonviolent data in testing dataset, meaning that it recognized the nonviolent images properly. The autoencoder is not trained for violent images and unable to identify the pattern in violent images; therefore, a higher reconstruction loss is expected. Here, the autoencoder acted as an anomaly detection method where violent images act as the anomalies.

After training an autoencoder, we fed the validation dataset with both violent and nonviolent images to the trained autoencoder, obtaining the reconstruction loss of the images in the validation set. The reconstruction loss was acquired as a vector. Subsequently, we utilized an SVM to classify the reconstruction loss. Finally, the testing images were passed through the trained autoencoder to obtain their reconstruction loss as a vector. This vector was then fed into the trained SVM to evaluate the performance.

A previous study has found that an autoencoder utilizing GoogleNet transfer learning and convolutional layers give better results for violent and nonviolent image recognition than other autoencoders [20]. We have utilized the same autoencoders proposed in [20] to evaluate the results on our new dataset.

E. Supervised learning - Shallow learning

Before employing supervised learning-shallow learning on pre-processed data, the feature extraction step needs to be completed. For the feature extraction, feature engineering techniques were used.

1) Feature engineering:

The text must be represented in a numerical format to feed text to natural language processing (NLP) and ML algorithms; this is known as feature engineering. The text can be represented with a vector of numbers known as a vector space model. Popular vector space models are BoW, TF-IDF, and one-hot vector encoding. These models aim to obtain similar representations for similar tokens of text. All three methods have sparsity problems that are inefficient to handle in the computer memory and out-of-vocabulary problems.

First, the vocabulary that contains all tokens in the corpus was created. The vector size is |V| as V is the number of unique tokens in the corpus. In one-hot encoding each token is represented by a vector of length |V|, and a sentence is a combination of all vectors of the tokens in the sentence. As different sentences in the corpus have different lengths, vector size varies with each other. One-hot encoding ignores the similarity between words [66].

The order of words and context are not considered in BoW representation, and it considers a sentence or a document as a bag of words. Vocabulary is developed as in the one-hot representation, and the number of occurrences of each word in the sentence can be stored in the vector representation. BoW does not represent each word as a vector; it represents the whole document as a vector without considering the order of words. This representation has a fixed length for all documents in the corpus. Documents with similar words can be identified using BoW, though different words with similar meanings cannot be identified. Bag-of-n-grams can help obtain a semantic meaning between words [66]. “Countvectorizer” function in scikit-learn was used to implement the BoW method. TF-IDF is another text representation method with two terms: TF explains the importance of a word within a document, and IDF explains the importance of the same word concerning other documents in the corpus [66]. “TfidfVectorizer” in scikit-learn was used to implement TF-IDF.

2) Classification Algorithms:

Encoded data were fed into ML algorithms such as SVM, LR, NB, and RF. SVM computes the optimal hyperplane by maximizing the margin between support vectors and LR computes a line according to a sigmoid function [14, 38]. For LR, Gradient descent or maximum likelihood can act as the optimization algorithm [32, 51]. NB is based on the Bayes theorem that assumes all features are independent (of each other). NB is a generative algorithm where the posterior probability is calculated with a model that implements a joint distribution of X and Y. Equation 1 can be derived for the Bayes classifier; it can be categorized as Gaussian or multinomial, depending on the different distributions of P(x_i/y) [66].

$$\mathrm{P}(y/x_1, \dots, x_n)\ \alpha\ \mathrm{P(y)} \prod_{i=1}^{n} \mathrm{P}(x_i/y) \text{ ------------ Equation 1}$$

RF is a method that uses many uncorrelated decision trees to make predictions. More accurate predictions are received when each decision tree is independent of one another. RF implements bootstrap aggregation (bagging) that results in a crucial difference in the output by inputting a training set with minor changes [9]. The scikit-learn library was used for implementing shallow learning algorithms.

3) Sampling methods used in shallow algorithms

Although the data is divided into train and test, parameters in algorithms can be tweaked to give better results for the test set. A validation set was derived again from the train set to prevent the situation. Having three sets as train, validation, and test minimizes the data that can train the model. Cross validation, stratified sampling (an extension of cross validation), and re-sampling (a bootstrapping procedure) were used to solve the problem. The training set is divided into K folds. K-1 folds were used to train the model and the remaining *K*th fold was used to validate the model in cross-validation. The scikit-learn's "StratifiedKFold" was used in stratified sampling. Stratified sampling is an extension of cross-validation that uses stratified folds. Re-sampling that uses a bootstrapping method selects a sample with a pre-defined sample size. The model was trained on the selected sample and the model was tested on the data, which is not selected for the sample. The process can be repeated many times, and mean estimates can be obtained by averaging the values over the number of samples.

F. Supervised learning - Deep learning

1) Text padding and vocabulary creation:

The training set was tokenized into words, and a vocabulary was created for the training set. In the vocabulary, a unique ID was assigned to each word. The maximum length of sentences was selected depending on the number of tokens. Sentences were padded depending on the difference between sentence length and maximum length. The same vocabulary was used for the test set and assigned with IDs. Unknown tokens were assigned for words that were not in the vocabulary.

2) Feature engineering:

Pre-trained word embeddings were loaded after the vocabulary creation and text padding. A matrix was implemented with vocabulary size (as the row dimension) and embedding size (as the column dimension). Subsequently, distributed representations of text known as word2vec [40] and fastText [8] were used as the embedding mechanisms for deep learning algorithms. A Sinhala dataset created in previous research was also used to create new embeddings in conjunction with a random subset of the dataset collected in our research [48, 57]. However, the embedding models created using our dataset did not perform well. Text that was converted using the Unidecode library in Python and text without the conversion was also applied to generate word2vec and fastText models. However, by comparing the obtained accuracies, finally, pre-trained embeddings obtained from previous research were used [16, 57].

3) Classification Algorithms:

1D CNN [64], LSTM [27], GRU [11], bidirectional LSTM (BiLSTM) [24], and bidirectional GRU (BiGRU) [6] were utilized as deep learning algorithms. Ensemble methods, 1D CNN with LSTM, 1D CNN with GRU, 1D CNN with BiLSTM, and 1D CNN with BiGRU were also tested to evaluate the performance. Filter size, number of filters, number of layers, optimization algorithms, and number of epochs were modified to find the optimum result in 1D CNN. The size of the hidden layer, number of layers, and number of epochs were modified in the LSTM and GRU to find an optimum result. The learning rate was reduced to prevent overfitting. The output of 1D CNN layers with different filter sizes as 2, 3, 4, 5, 7, and 11 were concatenated. Figure 4 shows the architecture of 1D CNN. The output was sent through a fully connected layer to obtain the final output.

In Ensemble architectures, output obtained in 1D CNN was fed through recurrent models such as LSTM and GRU. The ensemble model, which combines 1D CNN and GRU, is depicted in Figure 5. Text with an embedding dimension 300 is fed to the model. Three 1D CNN filters are used to extract the features, followed by a max pooling layer. The outputs obtained from the three filters are concatenated. The concatenated output is reshaped and sent through a GRU layer. The output obtained from GRU is fed to a fully connected layer, resulting in the classification output.

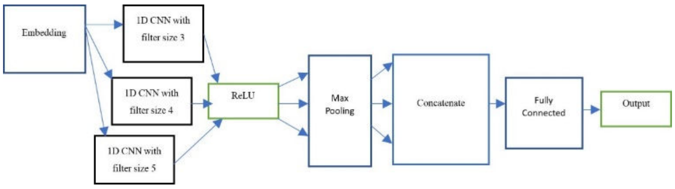

Embedding
1D CNN with filter size 3
1D CNN with filter size 4
1D CNN with filter size 5
ReLU
Max Pooling
Concatenate
Fully Connected
Output

Figure 4. 1D CNN architecture

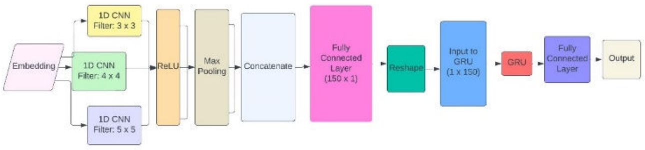


Figure 5. CNN GRU architecture

The cross-lingual representation of robustly optimized BERT pre-training approach (RoBERTa) (XLM-R) XLM-R was used as the BERT architecture, which is trained for 100 different languages and Sinhala is also included in these 100 languages [17]. Cross-lingual language model (XLM) was introduced to support 100 languages [13]. XLM uses Byte-Pair Encoding (BPE) to gain the sharing capability. In BPE, frequently used sub-word pairs are merged so they can easily represent an unknown word with sub-words that are already in the vocabulary. XLM-R, which works similarly to XLM, was trained according to the RoBERTa; RoBERTa uses a masked language model (MLM). The model was trained for the training set using "AdamW" optimization function and tested with a test set. The Hugging Face library which was implemented using PyTorch helped to access BERT interfaces [71]. The dataset in XLM-R was loaded and tokenized using a sentence piece tokenizer. All BERT algorithms expect sentences in the corpus to be tokenized in a distinct format. XLM-R requires similar formatting. The three special tokens used in BERT architecture are [CLS] as the classifier, [SEP] as the separator, and [PAD] as the padding. In XLM-R, the main tokens are <s> to indicate the beginning of a sequence, <\s> to indicate the separation of sequences and the end of a sequence, and <pad> as the padding. The method "encode_plus" returns the padded token list and attention mask. Attention mask indicates the separation between real tokens and padded tokens. "XLMRobertaTokenizer" was used as the tokenizer, and the "XLMRobertaForSequenceClassification" model was defined as the model for XLM-R [71].

With these deep learning techniques, early stopping was used as a promising technique to avoid overfitting and to find the most suitable model [47, 62].

G. *Explainable AIs*

Most ML models are black boxes; hence inner workings are not visible. Therefore, LIME and SHAP were used to describe the decisions taken by the black boxes [50, 33]. Using graphical pictures and details provided by the explainable APIs, texts that influenced the decision of the ML algorithms can be identified. 'LIME' model is a local approximation of the ML model. An instance in the dataset was selected, and the sample size in the LIME was initialized. The default sample size is 5,000, and better results can be obtained as the sample size increases. According to the sample size, the instance was perturbed by removing some of the tokens in the instance to create a sample. The sample obtained by perturbation was inputted to a custom prediction function that uses the trained ML model to calculate the prediction probability of each perturbation. The weights of the perturbed instances are calculated depending on the proximity to the original instance. LIME outputs the weights of each feature which helps to get a view of which features caused the decision given by the ML model. SHAP is based on game theory, and all features act as players in the game. SHAP calculates the average marginal contribution of a feature regarding all possible coalitions. Other than LIME and SHAP, IG is also used to describe deep learning models [61]. IG calculates the gradients of the output to its features. Initially, an instance was selected. The instance is interpolated starting from a baseline model. Then the gradient is calculated to check the changes in the features to the model prediction.

IV. RESULTS AND DISCUSSION

Here we present the obtained results for different ML algorithms we used, to identify hate speech related images. First, we will present the results of unsupervised learning algorithms, then shallow supervised learning algorithms and finally the results of deep learning algorithms.

A. *Results of unsupervised learning algorithms*

Table 1 presents the results for the autoencoders using the dataset mentioned in Section III A. Autoencoder with convolutional layers have shown better results than other autoencoders.

Table 1. Results for autoencoders

Model	Accuracy	Precision	Recall	F1 Score
GoogLeNet	0.657	0.6599	0.657	0.6555
Convolutional	0.727	0.7304	0.727	0.726

B. *Results of supervised learning algorithms*

Here, first we present the results of shallow ML algorithms, classifying the images using the text in the images. Results of shallow and deep ML algorithms were obtained using two main methods: using popular performance metrics and using explainable AI methods.

1) *Results of shallow ML algorithms using performance metrics:*

Using popular performance metrics for NB and LR algorithms, accuracy of the results was low, for one hot encoding feature extraction method, compared to other methods such as BoW and TF-IDF (see Table 2).

Table 2. Accuracy of one-hot encoding for NB and LR

	Classification Method	
	NB	LR
Accuracy	0.685	0.69

Initially, computations were performed on a dataset comprising 1,000 violent and 1,000 nonviolent images. Subsequently, the dataset size was expanded, and augmentation techniques were applied to further increase its size. Table 3 provides the results of different performance metrics for TF-IDF embedding for shallow algorithms; MNB, LR, SVM and RF before expanding the dataset in Unidecode format. Table 4 presents results after expanding the dataset but without augmentation and Unidecode format. Table 5 depicts the results of different performance metrics for the augmented and in Unidecode format. Table 6 describes the results of the BoW embedding with Unidecode and augmented data, and Table 7 describes the results for Unidecode and increased dataset but before the augmentation. According to these tables, NB classification has obtained higher results than other classification algorithms, RF showed lower results, and TF-IDF and BoW have obtained comparable results. The conversion of text to Unidecode format and expansion of the dataset has led to a noticeable improvement in the results. 91% accuracy was obtained for TF-IDF, BoW with NB classifier. The results of the BoW were slightly higher than TF-IDF.

Table 3. TF-IDF results before increasing the dataset but with Unidecode data

Metrics	Classification Methods			
	MNB	LR	SVM	RF
Accuracy	0.8725	0.875	0.875	0.6575
roc_auc_score	*0.9578*	*0.9477*	*0.9464*	0.7433

F1	0.8771	0.8775	0.878	0.5387
Precision	0.8505	0.8647	0.8612	0.8333
Recall	*0.9055*	0.8905	0.8955	0.398

Table 4. TF-IDF results after increasing the dataset but without Unidecode and augmented data

Metrics	MNB	LR	SVC	RF
Accuracy	0.829	0.8261	0.8217	0.7878
Precision	0.7851	0.8318	0.8125	0.7956
F1	0.8332	0.8159	0.8155	0.7735
Recall	0.8876	0.8006	0.8186	0.7526
ROC_AOC_Score	0.9238	0.9055	0.9074	0.8696

Table 5. TF-IDF results for the augmented and Unidecode dataset

Metrics	MNB	LR	SVC	RF
Accuracy	0.9074	0.8953	0.8961	0.813
Precision	0.8784	0.899	0.8925	0.8414
F1	0.9075	0.8908	0.8925	0.7962
Recall	0.9385	0.8827	0.8925	0.7556
ROC_AOC_Score	0.9721	0.9628	0.9601	0.8967

Further we have used NB as the classification algorithm with different sampling techniques and obtained the performance metrics (see Table 8). MNB with the BoW method has obtained better results for all the metrics in cross-validation (k-fold and stratified). However, employing cross-validation did not improve the previous result. Results depict that k-fold and stratified sampling have higher results than resampling.

2) Results of shallow ML algorithms using explainable methods:

We have used two text examples to describe the results obtained for XAI methods in shallow learning. Preprocessed and Unidecode text of Sentences 1 and 2 are shown in Table 9. Figure 6 and Figure 7 depict the LIME and SHAP outputs of sentence 1.

Table 6. BoW results for the augmented and Unidecode dataset

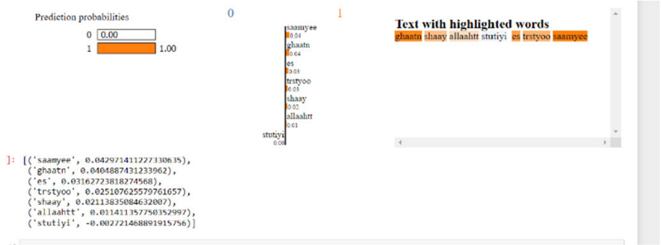
Metrics	Classification Methods			
	MNB	LR	SVM	RF
Accuracy	***0.9163***	0.8875	0.8534	0.815
roc_auc_score	***0.9735***	***0.9536***	***0.9323***	0.8927
F1	***0.915***	0.881	0.8462	0.7991
Precision	0.8914	***0.9106***	0.8634	0.8515
Recall	***0.9399***	0.8534	0.8296	0.7528

Table 7. BoW results for the Unidecode dataset but without the augmentation

Metrics	Classification Methods			
	MNB	LR	SVM	RF
Accuracy	***0.9047***	0.8732	0.8532	0.8381
F1	***0.9026***	0.8655	0.8453	0.8236
Precision	0.8825	0.8931	0.8639	0.8805
Recall	***0.9235***	0.8396	0.8276	0.7736

Table 8. Results of MNB classification for sampling methods with Unidecode and augmented data

Embedding	Sampling	Accuracy	F1	Precision	Recall
BoW	k-fold	***0.91***	***0.911***	0.884	***0.940***
TF-IDF	k-fold	***0.91***	0.893	0.867	***0.921***
BoW	Stratified	***0.91***	***0.903***	0.85	***0.956***
TF-IDF	Stratified	***0.91***	***0.913***	0.889	***0.939***
BoW	Resampling	0.862	0.867	0.846	0.889
TF-IDF	Resampling	0.862	0.865	0.860	0.871



```
]: [('saamyee', 0.042971411227330635),
    ('ghaatn', 0.0404887431233962),
    ('es', 0.03162723818274568),
    ('trstyoo', 0.025107625579761657),
    ('shaay', 0.02113835084632007),
    ('allaahtt', 0.011411357750352997),
    ('stutiyi', -0.002721468891915756)]
```

Figure 6. LIME results for sentence 1 - Shallow learning

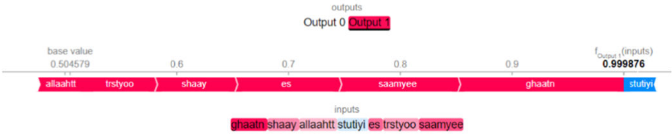


Figure 7. SHAP results for sentence 1 - Shallow learning.

LIME output with BoW as the embedding and NB as the classification algorithm has found Sinhala words ඝාතන (killings), සහාය (supported), අල්ලාහ්ට (Allah), ත්‍රස්තයෝ (terrorists), and සාමයේ (peace) caused to conclude that sentence 1 as violent. Sentence 2 can be identified as nonviolent. Violent words are highlighted (orange) in the text. Although Sinhala words සහාය (supported) and සාමයේ (peace) are nonviolent words, they were identified as violent. Orange colour indicates violent words, and others are nonviolent words.

3) *Results of deep learning algorithms using performance metrics:*

Table 10 presents the Sinhala text classification results using deep learning algorithms. The results were analysed with and without data augmentation. The presented outcomes are the best possible outputs obtained under different conditions: learning rate and number of epochs. Superior results were obtained for 1D CNN with a learning rate of 0.002. The learning rate was chosen as 0.000001 for other algorithms like LSTM and GRU. The GRU with CNN ensemble models (250 epochs) converge to a solution within fewer numbers of epochs than the CNN models (450 epochs).

XLM-R was evaluated for 500, 700 and 1,000 epochs with a learning rate of 0.000001, obtaining 93% accuracy, which is better than that of other models. XLM-R achieved over 90% for precision, recall, and F1-score, also outperforming other models. Subsequently, the GRU and CNN ensemble model, incorporating word2vec achieved 91% accuracy. Similarly, CNN with BiGRU utilizing word2vec, 1D CNN with word2vec and 1D CNN with fastText achieved 90% accuracy. In the context of 1D CNN, fastText with 300 embedding dimensions showed better results than word2vec embedding. Figure 8 illustrates the confusion matrix of XLM-R.

Figures 9 and 10 depict the loss and accuracy curves of nine deep learning models, respectively. CNN with GRU and CNN with BiGRU that incorporate word2vec, exhibit lower loss than LSTM and CNN models. Furthermore, CNN with GRU, incorporating word2vec, exhibit higher accuracy compared to other models.

Table 9. Sinhala text examples

Sentence No.	**Text**	**English Translation**	**Preprocessed Text**	**Unidecode Text**
Sentence 1	මේ ඝාතන වලට සහාය දුන් අල්ලාහ්ට ස්තුතියි ! අයි එස් ත්‍රස්තයෝ කියයි. සාමයේ ආගම මෙය ද?	Thanks to Allah, who supported these killings! IS terrorists say. Is this the religion of peace?	ඝාතන (killing) සහාය (support) අල්ලාහ්ට (Allah) ස්තුතියි (thanks) එස් (IS) ත්‍රස්තයෝ (terrorists) සාමයේ (peace)	ghaatn shaay allaahtt stutiyi es trstyoo saamyeet
Sentence 2	ඔබේ ගෙල වටා පැළඳිය හැකි හොඳම ආභරණය වන්නේ ඔබේ දරුවන්ගේ දෑතයි	The best jewelry you can wear around your neck is your children’s arms	ගෙල (neck) වටා (around) පැළඳිය (wear) හොඳම (best) ආභරණය (jewelry) දරුවන්ගේ (children’s) දෑතයි (arms)	gel vttaa paellndiy hondm aabhrnny druvngee daaetyi
Sentence 3	බෞද්ධයින්ට තඩි නොබා දෙපිල බෙදී මරාගන්නා අන්තවාදී මුස්ලිම් කල්ලි මඩිනු. ත්‍රස්තවාදයට නිදහසේ වැඩෙන්නට ඉඩදී බලා සිටින්නේ මේ රට තවත් ඉරාකයක් වෙනතුරුද? බෞද්ධ අන්තවාදයක් ගැන බොරු බේගල් ඇද නොබා මුස්ලිම් අන්තවාදය ගැන ඇත්ත පිළිගන්න.	Instead of punishing Buddhists, stop extremist Muslim gangs who divide and kill. Are they allowing terrorism to grow freely and waiting for this country to become another Iraq? Accept the truth about Muslim extremism without pulling false stories about Buddhist extremism.	බෞද්ධයින්ට (Buddhists) තඩි (punish) නොබා (not) දෙපිල (two groups) බෙදී (divide) මරාගන්නා (killing) අන්තවාදී (extremist) මුස්ලිම් (muslim) කල්ලි (gang) මඩිනු (stop). ත්‍රස්තවාදයට (terrorism) නිදහසේ (freely) වැඩෙන්නට (grow) ඉඩදී (let) සිටින්නේ (waiting) ඉරාකයක් (Iraq) වෙනතුරුද (until) බොරු (false) බේගල් (stories) ඇද නොබා (without telling) මුස්ලිම් (muslim) අන්තවාදය(extremism) ඇත්ත (truth) පිළිගන්න (accept)	bauddhyintt tddi nobaa depil bedii mraagnnaa antvaadii muslim klli mddinu. trstvaadytt nidhsee vaeddenntt idddii blaa sittinnee mee rtt tvt iraakyk venturud? bauddh antvaadyk gaen boru beegl aed nobaa muslim antvaady gaen aett pillignn.

Table 10. Results of deep learning algorithms in text classification

Method	Augmentation[1]	Accuracy	Precision	F1	Recall
CNN+word2vec 300 [2]	No	***0.9033***	0.8791	***0.9022***	***0.9265***
CNN+BiGRU+word2vec 300	No	0.8788	0.8855	0.8776	0.8763
CNN+BiGRU+word2vec 300	Yes	***0.9041***	***0.9055***	***0.9038***	***0.9032***
CNN+GRU+word2vec 300	No	0.8925	0.8926	0.8923	0.8921
BiGRU+word2vec 300	No	0.8911	0.8909	0.8911	0.8914
BiLSTM+word2vec 300	No	0.8853	0.8852	0.8851	0.885
CNN+BiLSTM+word2vec 300	No	0.8889	0.8892	0.8889	0.8897
GRU+word2vec 300	No	0.8939	0.8941	0.8937	0.8935
LSTM+word2vec 300	No	0.8853	0.8874	0.8847	0.8839
CNN+LSTM+word2vec 300	No	0.8918	0.8918	0.8916	0.8914
CNN+word2vec 300	Yes	***0.9001***	0.8586	***0.9019***	***0.9497***
CNN+GRU+word2vec 300	Yes	***0.9136***	***0.9137***	***0.9134***	***0.9132***
BiGRU+word2vec 300	Yes	0.8987	0.8986	0.8987	0.8989
GRU+word2vec 300	Yes	0.898	0.8988	0.8978	0.8973
CNN+fastText 300	Yes	***0.9048***	***0.9044***	***0.9012***	0.898
BiGRU+fastText 300	Yes	0.8872	0.8871	0.8871	0.8872
CNN+GRU+fastText 300	Yes	0.8899	0.8898	0.8898	0.8899
GRU+fastText 300	No	0.8687	0.8691	0.8683	0.868
CNN+fastText 300	No	0.8788	0.8833	0.8725	0.8621
BiGRU+fastText 300	No	0.8687	0.8685	0.8685	0.8684
CNN+GRU+fastText 300	No	0.8874	0.8873	0.8873	0.8872
CNN+fastText 450	Yes	0.898	0.9088	0.8927	0.8771
CNN+GRU+fastText 450 [3]	Yes	0.896	0.8965	0.8958	0.8955
XLM-R	Yes	***0.9203***	***0.9118***	***0.9182***	***0.9245***
XLM-R	No	***0.93***	***0.9371***	***0.9265***	***0.916***

[1] Augmentation: Yes - Refers to the dataset containing augmented data. No - Refers to the dataset without any augmentation.

[2] CNN stands for 1D CNN. 300 represents the embedding dimension. The "+" sign signifies the fusion of the "CNN" algorithm and the "word2vec" embedding mechanism with 300 embedding dimensions.

[3] 450 represents the embedding dimension.

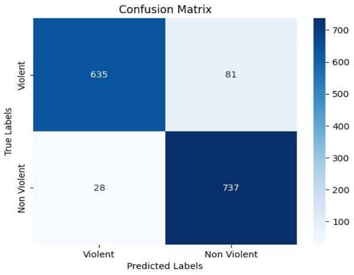


Figure 8. Confusion matrix of XLM-R

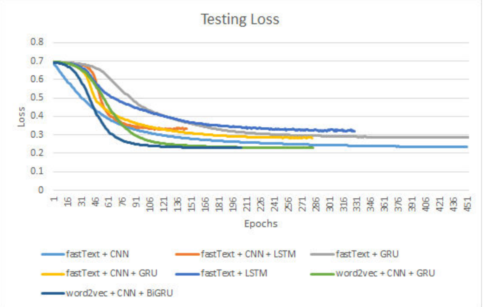


Figure 9. The loss of the model incurred on the test data.

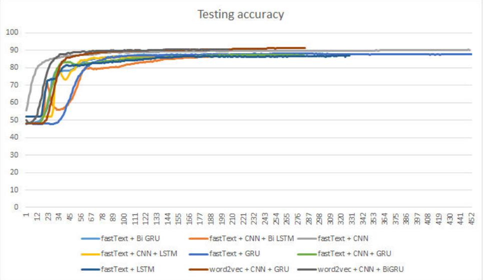


Figure 10. The accuracy achieved on the test data.

Table 11 presents the performance results of 1D CNN, CNN, and GRU ensemble model, as well as the XLM-R models after partitioning the dataset into train, validation, and test subsets. The results depict that the XLM-R model achieved superior results compared to the 1D CNN and GRU ensemble model.

Table 11. Performance results on different dataset splits

	Accuracy	Precision	Recall	F1-score
fastText + 1D CNN	0.8862	0.9024	0.8626	0.8821
word2vec + CNN + GRU	0.8977	0.8992	0.8973	0.8975
XLM-R	0.93	0.94	0.91	0.92

3) *Results of deep learning algorithms using explainable methods:*

Sentences 1, 2, and 3, as depicted in Table 10, are utilized in the context of deep learning. Sentences 1 and 3 are identified as violent, whereas Sentence 2 is identified as nonviolent. Violent words identified by LIME using CNN, and GRU ensemble are හම්බයො (similar word for Muslims), අත්අඩංගුවට (arrested), මරනයට (to death), නපුන්සකයෝ (eu nuchs), බොර (fake), උද්ඝෝෂණයක් (campaign), පගාව (revenge), විනාශ (destruction), කුඩු ජාවාරමේ (drug dealing), මුස්ලිම් (Muslim), ත්‍රස්ත (terror), දූෂනය (corruption), ඉස්ලාමයේ (Islam), අන්තවාදය (extremism), බේබදු (drunkenness), බැනල (scolded), හොරු (thieves), අවජාතක (bastards), වංචාව (fraud), පිළිකුලෙන් (disgusted), බලු (dogs), නීතිය (law), මුසල්මානුවන්ට (similar word for Muslims), හලාල් (Halal), කුඩුකාරයෝ (drug addicted), අන්තවාදීන් (extremists), හොරකං (thieves), and මාට්ටු (caught). The identified nonviolent words are, ආදරෙන් (with love), ඉවසීම (patience), හිතේ (heart), බැඳීමකින් (bond), සිනහවට (smile), කඳුලු (tears), පැලඳිය (dress), හොඳම (best), දරුවන්ගේ (children's), ගෙල (neck), සතුටින් (happy), දෙමාපියන් (parents), and මානසික (mental). Although some nonviolent words were identified as violent in Sentence 1 and Sentence 2 by shallow machine learning, in deep learning they were identified correctly. Figure 11 and Figure 12 show the LIME output of Sentence 1 and Sentence 2 respectively.

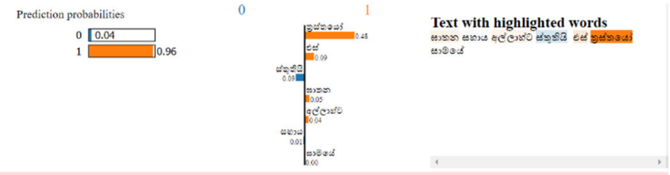


Figure 11. LIME results for sentence 1 using deep learning.

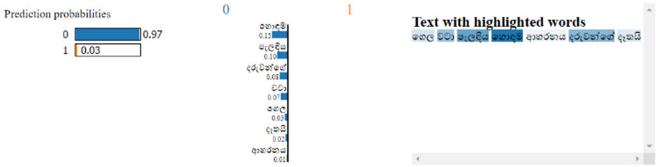


Figure 12. LIME results for sentence 2 using deep learning.

SHAP has produced slightly different results than LIME. Figure 13 shows sentence 1. The red colour indicates violent words. According to the figure, සාමයේ (peace) is identified as a violent word.

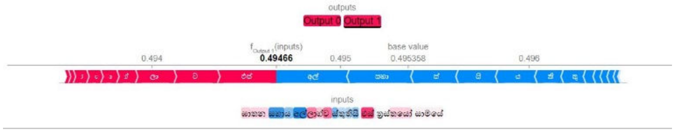
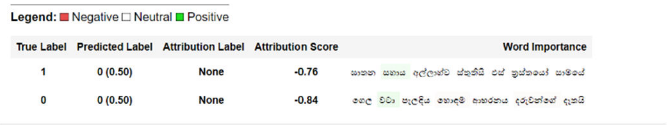


Figure 13. SHAP output of sentence 1

IG and LIME have produced different outputs. Red and green colors indicate violent and nonviolent words, respectively. Figure 14 shows sentences 1 and 2 for IG. According to thoutput, sentence 1 is identified as nonviolent (the predicted label is 0). Violent words are not highlighted in the text either.



True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	0 (0.50)	None	-0.76	ඝාතන සහාය අල්ලාහ්ට ස්තුතියි එස් ත්‍රස්තයෝ සාමයේ
0	0 (0.50)	None	-0.84	ගෙල විවා පැලඳිය හොඳම ආභරනය දරුවින්ගේ දැකයි

Figure 14. IG results for sentences 1 and 2

V. CONCLUSION

In the context of classifying images posted based on hate speech, unsupervised learning algorithms achieved 73% accuracy. Increased dataset size, along with characters encoded using Unidecode, has resulted in a 92% accuracy for shallow machine learning algorithms. Comparable results were obtained for BoW and TF-IDF, with slightly higher results for BoW. Models that employ GRU have achieved 91% accuracy. Models with 1D CNN achieved 90% accuracy, and the XLM-R algorithm obtained 93% accuracy. RNN architectures that employ LSTM have shown lower results than models that incorporate GRU and 1D CNN. However, LSTM with the CNN model obtained 89% accuracy.

In most cases, data augmentation has improved the results of models employing the GRU architecture. Among the models that employ GRU, slightly better results have been obtained for word2vec than for fastText. LIME has shown better interpretation than SHAP and IG. Supervised learning of text classification produced better results than unsupervised learning for identifying violent Sinhala image posts. This research can be further enhanced by extracting the text from image posts automatically using a text extraction method.

REFERENCES

[1] Aggarwal C.C., Sathe S.: Theoretical foundations and algorithms for outlier ensembles. In: Acm sigkdd explorations newsletter, vol. 17(1), pp. 24--47, 2015.

[2] Aktı S., Ofli F., Imran M., Ekenel H.K.: Fight detection from still images in the wild. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 550--559. 2022.

[3] Amorim M., Bortoloti F.D., Ciarelli P.M., Salles E.O., Cavalieri D.C.: Novelty detection in social media by fusing text and image into a single structure. In: IEEE Access, vol. 7, pp. 132786--132802, 2019.

[4] Amos B., Ludwiczuk B., Satyanarayanan M., et al.: Openface: A general-purpose face recognition library with mobile applications. In: CMU School of Computer Science, vol. 6(2), p. 20, 2016.

[5] Badjatiya P., Gupta S., Gupta M., Varma V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th international conference on World Wide Web companion, pp. 759--760. 2017.

[6] Bharti: Sentimental with Multi Layer Bi directional RNN using PyTorch, 2020. URL {https://medium.com/@bhartikukreja2015/sentimental-with-multi-layer-bi-directional-rnn-using-pytorch-4f386297a0fc}.

[7] Blandfort P., Patton D.U., Frey W.R., Karaman S., Bhargava S., Lee F.T., Varia S., Kedzie C., Gaskell M.B., Schifanella R., et al.: Multimodal social media analysis for gang violence prevention. In: Proceedings of the International AAAI conference on web and social media, vol. 13, pp. 114--124. 2019.

[8] Bojanowski P., Grave E., Joulin A., Mikolov T.: Enriching word vectors with subword information. In: Transactions of the association for computational linguistics, vol. 5, pp. 135--146, 2017.

[9] Breiman L.: Random forests. In: Machine learning, vol. 45, pp. 5--32, 2001.

[10] Chhabra A., Vishwakarma D.K.: A literature survey on multimodal and multilingual automatic hate speech identification. In: Multimedia Systems, pp. 1--28, 2023.

[11] Chung J., Gulcehre C., Cho K., Bengio Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: arXiv preprint arXiv:1412.3555, 2014.

[12] Cohen J.: A coefficient of agreement for nominal scales. In: Educational and psycho logical measurement, vol. 20(1), pp. 37--46, 1960.

[13] Conneau A., Lample G.: Cross-lingual language model pretraining. In: Advances in neural information processing systems, vol. 32, 2019.

[14] Cramer J.S.: The origins of logistic regression. In: , 2002.

[15] Davidson T., Warmsley D., Macy M., Weber I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of the international AAAI conference on web and social media, vol. 11, pp. 512--515. 2017.

[16] Demotte P., Senevirathne L., Karunanayake B., Munasinghe U., Ranathunga S.: SEN CAT Tool for Sinhala Sentiment Analysis, 2020. URL https://sencat.lk/.

[17] Devlin J., Chang M.W., Lee K., Toutanova K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: arXiv preprint arXiv:1810.04805, 2018.

[18] Dhananjaya V., Demotte P., Ranathunga S., Jayasena S.: BERTifying Sinhala--A Comprehensive Analysis of Pre-trained Language Models for Sinhala Text Classification. In: arXiv preprint arXiv:2208.07864, 2022.

[19] Dias D.S., Welikala M.D., Dias N.G.: Identifying racist social media comments in sinhala language using text analytics models with machine learning. In: 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 1--6. IEEE, 2018.

[20] Dikwatta U., Fernando T.: Violence Detection of Sinhala Image Posts with Autoencoders. In: 2021 10th International Conference on Information and Automation for Sustainability (ICIAfS), pp. 275--280. IEEE, 2021.

[21] Djuric N., Zhou J., Morris R., Grbovic M., Radosavljevic V., Bhamidipati N.: Hate speech detection with comment embeddings. In: Proceedings of the 24th international conference on world wide web, pp. 29--30. 2015.

[22] Edunov S., Ott M., Auli M., Grangier D.: Understanding back-translation at scale. In: arXiv preprint arXiv:1808.09381, 2018.

[23] Fortuna P., Nunes S.: A survey on automatic detection of hate speech in text. In: ACM Computing Surveys (CSUR), vol. 51(4), pp. 1--30, 2018.

[24] Graves A., Schmidhuber J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. In: Neural networks, vol. 18(5-6), pp. 602--610, 2005.

[25] Haque R., Islam N., Tasneem M., Das A.K.: MULTI-CLASS SENTIMENT CLASSIFICATION ON BENGALI SOCIAL MEDIA COMMENTS USING MACHINE LEARNING. In: International Journal of Cognitive Computing in Engineering, 2023.

[26] He K., Gkioxari G., Dollár P., Girshick R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961--2969. 2017.

[27] Hochreiter S., Schmidhuber J.: Long short-term memory. In: Neural computation, vol. 9(8), pp. 1735--1780, 1997.

[28] Hossin M., Sulaiman M.N.: A review on evaluation metrics for data classification evaluations. In: International journal of data mining & knowledge management process, vol. 5(2), p. 1, 2015.

[29] Jallad K.A., Ghneim N.: ArNLI: Arabic Natural Language Inference for Entailment and Contradiction Detection. In: arXiv preprint arXiv:2209.13953, 2022.

[30] Kaplan A.M., Haenlein M.: Users of the world, unite! The challenges and opportunities of Social Media. In: Business horizons, vol. 53(1), pp. 59--68, 2010.

[31] Le Q., Mikolov T.: Distributed representations of sentences and documents. In: International conference on machine learning, pp. 1188--1196. PMLR, 2014.

[32] Le Cam L.: Maximum likelihood: an introduction. In: International Statistical Review/Revue Internationale de Statistique, pp. 153--171, 1990.

[33] Lundberg S.M., Lee S.I.: A unified approach to interpreting model predictions. In: Advances in neural information processing systems, vol. 30, 2017.

[34] Malik J.S., Pang G., Hengel A.v.d.: Deep learning for hate speech detection: a comparative study. In: arXiv preprint arXiv:2202.09517, 2022.

[35] Malmasi S., Tetreault J., Dras M.: Oracle and human baselines for native language identification. In: Proceedings of the tenth workshop on innovative use of NLP for building educational applications, pp. 172--178. 2015.

[36] Malmasi S., Zampieri M.: Detecting hate speech in social media. In: arXiv preprint arXiv:1712.06427, 2017.

[37] Mazari A.C., Boudoukhani N., Djeffal A.: BERT-based ensemble learning for multi aspect hate speech detection. In: Cluster Computing, pp. 1--15, 2023.

[38] Meyer D., Wien F.: Support vector machines. In: The Interface to libsvm in package e1071, vol. 28, p. 20, 2015.

[39] Mikolov T., Chen K., Corrado G., Dean J.: Efficient estimation of word representations in vector space. In: arXiv preprint arXiv:1301.3781, 2013.

[40] Mikolov T., Chen K., Corrado G., Dean J.: Efficient estimation of word representations in vector space. In: arXiv preprint arXiv:1301.3781, 2013.

[41] Mozafari M., Farahbakhsh R., Crespi N.: A BERT-based transfer learning approach for hate speech detection in online social media. In: Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8, pp. 928--940. Springer, 2020.

[42] Naher J., Minar M.R.: Impact of social media posts in real life violence: A case study in Bangladesh. In: arXiv preprint arXiv:1812.08660, 2018.

[43] Omar A., Mahmoud T.M., Abd-El-Hafeez T.: Comparative performance of machine learning and deep learning algorithms for Arabic hate speech detection in osns. In: Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020), pp. 247--257. Springer, 2020.

[44] Omar A., Mahmoud T.M., Abd-El-Hafeez T., Mahfouz A.: Multi-label arabic text classification in online social networks. In: Information Systems, vol. 100, p. 101785, 2021.

[45] Orru P., et al.: Racist discourse on social networks: A discourse analysis of Facebook posts in Italy. In: Rhesis, vol. 5(1), pp. 113--133, 2015.

[46] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., et al.: Scikit-learn: Machine learning in Python. In: the Journal of machine Learning research, vol. 12, pp. 2825--2830, 2011.

[47] Prechelt L.: Early stopping-but when? In: Neural Networks: Tricks of the trade, pp. 55--69. Springer, 2002.

[48] Ranathunga S., Liyanage I.U.: Sentiment analysis of sinhala news comments. In: Transactions on Asian and Low-Resource Language Information Processing, vol. 20(4), pp. 1--23, 2021.

[49] Rathnayake H., Sumanapala J., Rukshani R., Ranathunga S.: Adapter-based fine-tuning of pre-trained multilingual language models for code-mixed and code-switched text classification. In: Knowledge and Information Systems, vol. 64(7), pp. 1937--1966, 2022.

[50] Ribeiro M.T., Singh S., Guestrin C.: ``Why should i trust you?'' Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135--1144. 2016.

[51] Ruder S.: An overview of gradient descent optimization algorithms. In: arXiv preprint arXiv:1609.04747, 2016.

[52] Ruwandika N., Weerasinghe A.: Identification of hate speech in social media. In: 2018 18th international conference on advances in ICT for emerging regions (ICTer), pp. 273- -278. IEEE, 2018.

[53] Samarasinghe S., Meegama R., Punchimudiyanse M.: Machine learning approach for the detection of hate speech in sinhala unicode text. In: 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 65--70. IEEE, 2020.

[54] Sandaruwan S.T., Lorensuhewa S.A.S., Munasinghe K.: Identification of abusive sinhala comments in social media using text mining and machine learning techniques. In: The International Journal on Advances in ICT for Emerging Regions, vol. 13(1), 2020.

[55] Schmidt A., Wiegand M.: A survey on hate speech detection using natural language processing. In: Proceedings of the fifth international workshop on natural language processing for social media, pp. 1--10. 2017.

[56] Senarath Y.: A language processing tool for Sinhalese, 2004. URL https://sinling. ysenarath.com/.

[57] Senevirathne L., Demotte P., Karunanayake B., Munasinghe U., Ranathunga S.: Sentiment analysis for sinhala language using deep learning techniques. In: arXiv preprint arXiv:2011.07280, 2020.

[58] Shorten C., Khoshgoftaar T.M.: A survey on image data augmentation for deep learning. In: Journal of big data, vol. 6(1), pp. 1--48, 2019.

[59] Silva E., Nandathilaka M., Dalugoda S., Amarasinghe T., Ahangama S., Weerasuriya G.T.: Machine Learning-Based Automated Tool to Detect Sinhala Hate Speech in Images. In: 2021 6th International Conference on Information Technology Research (IC ITR), pp. 1--7. IEEE, 2021.

[60] Sun S., Liu Y., Mao L.: Multi-view learning for visual violence recognition with maximum entropy discrimination and deep features. In: Information Fusion, vol. 50, pp. 43--53, 2019.

[61] Sundararajan M., Taly A., Yan Q.: Axiomatic attribution for deep networks. In: International conference on machine learning, pp. 3319--3328. PMLR, 2017.

[62] Sunde B.M.: Early Stopping for PyTorch, 2020. URL https://github.com/ Bjarten/early-stopping-pytorch.

[63] Suryawanshi S., Chakravarthi B.R., Arcan M., Buitelaar P.: Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In: Proceedings of the second workshop on trolling, aggression and cyberbullying, pp. 32--41. 2020.

[64] Tran C.: A Complete Guide to CNN for Sentence Classification with PyTorch, 2020. URL https://chriskhanhtran.github.io/posts/ cnn-sentence-classification/.

[65] Uppada S.K., Patel P.: An image and text-based multimodal model for detecting fake news in OSN?s. In: Journal of Intelligent Information Systems, pp. 1--27, 2022.

[66] Vajjala S., Majumder B., Gupta A., Surana H.: Practical natural language processing: a comprehensive guide to building real-world NLP systems. O'Reilly Media, 2020.

[67] Waseem Z., Hovy D.: Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: Proceedings of the NAACL student research workshop, pp. 88--93. 2016.

[68] Waseem Z., Thorne J., Bingel J.: Bridging the gaps: Multi task learning for domain transfer of hate speech detection. In: Online harassment, pp. 29--55, 2018.

[69] Watanabe H., Bouazizi M., Ohtsuki T.: Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. In: IEEE access, vol. 6, pp. 13825--13835, 2018.

[70] Webb G.I.: Decision tree grafting from the all-tests-but-one partition. In: Ijcai, vol. 2, pp. 702--707. 1999.

[71] Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., Cistac P., Rault T., Louf R., Funtowicz M., Davison J., Shleifer S., von Platen P., Ma C., Jernite Y., Plu J., Xu C., Scao T.L., Gugger S., Drame M., Lhoest Q., Rush A.M.: HuggingFace's Transformers: State-of-the-art Natural Language Processing, 2019. URL http://dx.doi.org/10. 48550/ARXIV.1910.03771.

[72] Won D., Steinert-Threlkeld Z.C., Joo J.: Protest activity detection and perceived violence estimation from social media images. In: Proceedings of the 25th ACM international conference on Multimedia, pp. 786--794. 2017.

[73] Zhang Z., Luo L.: Hate speech detection: A solved problem? the challenging case of long tail on twitter. In: Semantic Web, vol. 10(5), pp. 925--945, 2019.

[74] Šolc T.: Unidecode, lossy ASCII transliterations of Unicode text, 2022. URL https: //github.com/avian2/unidecode.

ACKNOWLEDGMENT

The authors wish to thank Apple Research and Development Center, Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Nugegoda, Sri Lanka, for providing the facilities needed to conduct the research.

AVAILABILITY OF SUPPORTING DATA

Data are available at the public repository https://github.com/umandaDik/Data.