JCSDA, Vol. 1, No. 2, 59–75
DOI: 10.69660/jcsda.01022404
ISSN 2959-6912

Hate Speech Detection from Transliterated Amharic Social Media Comments Using Machine Learning and Deep Learning Approaches

Zeleke Abebaw

Department of Software Engineering
Artificial Intelligence and Robotics Center of Excellence
Addis Ababa Science and Technology University, Addis Ababa, Ethiopia
Corresponding author: zeleke.abebaw@aastu.edu.et

Andreas Rauber

Institute of Information Systems Engineering,
Technical University of Vienna, Vienna, Austria

Solomon Atnafu

Department of Computer Science, Addis Ababa University,
Addis Ababa, Ethiopia

The rise of transliterated script usage on social media has presented significant challenges to hate speech detection models, as such scripts often bypass models trained exclusively on formal language datasets. Existing Amharic hate speech detection studies predominantly focus on datasets written in formal Amharic scripts using machine learning approaches, leaving transliterated comments underexplored. This research addresses the gap by evaluating the impact of auto-transliterated and manually transliterated datasets, merged with an existing Amharic hate speech dataset, on the performance of machine learning and deep learning classifiers. The study employed a total of 3,000 datasets which is split into ratio of 80:20 for training and testing. The dataset consists of auto-transliterated, manually transliterated, formal Amharic script, and their combinations. The classifiers including Support Vector Machine, single and multichannel Convolutional Neural Networks were assessed. Experimental results show that the multichannel CNN outperformed single-channel CNN models on the existing Amharic dataset, achieving an F1-score of 0.810 compared to 0.783 and 0.769 for single channel and multichannel CNN, respectively. However, combining transliterated datasets with the existing dataset did not improve classifier performance, likely due to the inconsistencies in scrip transliteration and dataset domain dependencies. This study concludes that transliterated datasets should be treated separately for hate speech detection, and combining datasets from different domains and transliteration techniques negatively impacts classifier performance.

Keywords: Hate speech detection, transliteration, Amharic words, Latin script, single channel CNN, Multichannel, SVM

1. Introduction

The use of social media platforms, such as Facebook and Twitter, has significantly increased user participation in communication, particularly in political, economic, and social domains [1]. Users actively engage in activities like political discussions, online commerce, and social matters, primarily through informal writing forms such as comments, posts, and chats [2]. This informal writing can be produced in native

language scripts or transliterated scripts. Transliteration involves converting text from one script to another [3][4], such as transforming the Amharic term "**በጣም ጥሩ**" (pronounced "bt'äm t'ru," meaning "very good" in English) into its Latin script equivalent, "bettam ttiru." The use of transliterated scripts on social media presents significant challenges for automated hate speech detection models. These models, typically trained on formal language scripts, often fail to detect hate speech written in transliterated forms, enabling hate speech to bypass detection mechanisms. For instance, while a hate speech comment such as "destroy all blacks" in English or "**ሁሉን ጥቁር አጥፋ**" in Amharic might be detected by models trained on formal scripts, the transliterated version "hulun ttiqur attfa" would likely evade detection. This gap in model capabilities highlights the urgent need to address the challenges posed by transliterated scripts. Existing research on Amharic hate speech detection has primarily focused on datasets written in formal Amharic scripts, employing machine learning approaches [5]-[8]. However, there remains a significant research gap in detecting hate speech from transliterated Amharic comments, particularly those posted on social media platforms like Facebook. Notably, the impact of auto-transliterated and manually transliterated datasets on the performance of hate speech detection models has not been explored. In this study, we investigate this gap by using a publicly available Latin-to-Amharic script transliteration tool, RBLatAm, to convert Amharic comments written in Latin scripts back into their original Amharic script. These comments are further processed and utilized for hate speech detection using Support Vector Machine (SVM) and Convolutional Neural Network (CNN)-based classifiers. The main contributions of this research are as follows:

- A comparative evaluation of SVM and CNN-based classifiers on both formal Amharic hate speech datasets and transliterated datasets.
- An assessment of the impact of auto-transliterated and manually transliterated datasets on classifier performance.
- The creation of a corpus of 1,000 transliterated Amharic comments collected from Facebook pages.
- A manual annotation of the transliterated comments into hate speech and non-hate speech categories.

The remainder of this article is organized as follows: Section 2 covers the literature review and related works. Section 3 presents the methodology. Section 4 discusses the results and findings. Finally, Section 5 concludes the article and outlines directions for future research.

2. Literature review

2.1. *Hate Speech*

Since hate speech lacks one universal definition, various scholars have attempted their own but related definition. Davidson et al. [9] defined hate speech as "state-

ments that attack or delegitimize particular groups of people based on a demographic category race, gender, religion, sexual orientation". Brown et al. [10] also defined hate speech as "any kind of communication in speech, writing or behavior, that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, based on their religion, ethnicity, nationality, race, color, descent, gender or other identity factor". On the other hand, Ward et al. [11] defined hate speech as "any type of discourse in which speakers principally seek to condemn, humiliate, or inspire hatred against their targets". The targets could be based on again ones color, race, religion, gender. In all of these definitions of hate speech, the central focuses have been an attack on people core identities. Unless hate speech is detected and managed early, it would definitely has bad consequences such as displacement of communities due to hatred, disruption of business as a result of conflicts, even causing deaths for many innocent civilians such as women and children globally [12]. Several researchers have indicated the negative effects of hate speech. Hate speech creates psychological damage and self-hatred [13], silences women and minorities [14], causes illegal acts of discrimination [15], and creates a disordered society [14]. In general, hate speech threatens individual freedoms, human dignity, and equality while also inciting social tensions, upsetting the peace and order in the community, and jeopardizing peaceful coexistence. Hence, various bodies take different initiatives to minimize the spread of hate speech. While government bodies implement law enforcement, social media companies use machine-learning algorithms. The algorithms are dependent on the type of language scripts they have been trained on such as formal English language scripts, formal Amharic scripts, or transliteration of scripts.

2.2. *Transliteration*

Transliteration is the process of representing words of one language using the scripts of another language [16]. The concept of transliteration for Amharic texts does not have a universally agreed-upon standard. Transliteration involves representing Amharic sounds using the Latin alphabet, and in practice, different users may adopt varying approaches, leading to nonstandard transliteration. For example, the Amharic character "ሀ"might be transliterated as "ha", "he," or other variations depending on the writer's preference. This lack of standardization is particularly evident on social media, where informal and personalized transliteration practices are common, influenced by phonetics, individual habits, or familiarity with English pronunciations. In contrast, automated transliteration systems typically follow predefined rules, creating a form of standardized transliteration. These systems ensure consistency by mapping each Amharic character to a specific Latin representation, such as always converting "ሀ" to "ha." However, the informal, nonstandard transliteration often used in social media contexts does not adhere to such rules, making it more complex and unpredictable. Despite the prevalence of nonstandard transliteration, its patterns and impact remain underexplored in re-

Table 1. Social media comments transliterated by the RBLatAm[20]

Amharic comments in Latin characters	**Comments in original Amharic character**	**Translation** (google translate with manual modification)	**Class**
jimma yefqr yewubet ketema	**ጅማ የፍቅር የውበት ከተማ**	jimma city of romantic and beauty	notHate
eyemerereh watew gena tsebaberaleh be seber ziena	**እየመረረህ ዋጠው ገና ትሰባበራለህ በሰበር ዜና**	You will be bitter and broken by breaking news	notHate
znjero kersam hodam telalaki banda neh	**ዝንጅሮ ከርሳም ሆዳም ተላላኪ ባንዳ ነህ**	You are a greedy monkey	Hate
mnalebet trakter bgezulachaw	**ምናለበት ትራክተር ቢገዙላቸው**	Why don't you buy a tractor	notHate

search. While the idea of auto-transliteration standards is relatively clear due to the consistency enforced by algorithms, the phenomenon of nonstandard transliteration and its implications—particularly in tasks like natural language processing (NLP) and hate speech detection—has not been adequately discussed. Given the increasing use of such transliterations in online interactions, conducting research on transliteration dataset is essential, especially in the context of hate speech detection. Understanding how these informal transliteration practices influence model performance, accuracy, and bias in hate speech classifiers could significantly enhance the development of inclusive detection systems. By addressing this research gap, we can better handle the linguistic variability and complexity present in transliterated texts, ultimately improving the effectiveness of NLP models in detecting hate speech across diverse social media platforms. In the Latin-to-Amharic script transliteration tools, the works of [17]-[20] are notable examples. These tools helps to build Amharic datasets that can be used for various NLP applications such as sentiment analysis and hate speech detection tasks. However, not all of the tools are publicly available except the Latin-to-Amharic transliteration tool called Rule-Based Latin to Amharic (RBLatAm) [20]. We have used this tool to transliterate Amharic comment in Latin scripts. For example, the comment "bettam ttru asteyayet" is transliterated into original Amharic script as "**በጣም ጥሩ አስተያየት**", meaning "very good opinion" in English.

In this research, we investigate the potential impact of the auto-transliterated social media comments on the performance of hate speech detection models such as SVM and CNN-based classifiers. For this purpose, we have conducted comparative analysis using manually transliterated and auto-transliterated social media comments by merging them on an existing Amharic hate speech datasets. Table 1, show sample social media comments transliterated by the RBLatAm.

Detecting hate speech on social media platforms, particularly Facebook, has become increasingly important due to its previously mentioned negative impacts. If left unaddressed, hate speech can escalate tensions, leading to harm at individual,

group, or societal levels. Numerous studies have focused on detecting hate speech from social media, including approaches involving transliterated text. This section provides a review of these research efforts. Sazzed et al. [21] have conducted abusive content detection written in transliterated Bengali words. In their research, the authors have provided a 3,000 annotated transliterated Bengali corpus. The corpus is classified into two classes abusive and non-abusive 1500 each to avoid class imbalance. As a baseline, SVM, RF, and LR with TFIDF feature extraction of character n-grams were used. In addition, the deep learning model of BILSTM. The authors concluded that the deep learning-based architecture BiLSTM achieves a substantially lower F1-score than LR and SVM due to its smaller size (only 3000 comments). In another study, Taawab et al. [22] classified 1,300 transliterated Bengali comments into abusive and non-abusive categories using machine learning (ML) and deep learning (DL) models. For classifying comments, several algorithms have been used including multinomial naive Bayes (MNB), logistic regression (LR), linear SVM, decision tree (DT), AdaBoost, random forest (RF), RBF SVM, gradient boosting, recurrent neural network (RNN), gated recurrent units (GRU), and long short-term memory (LSTM). Among the models, Logistic regression with countVectorizer outperformed the others with an F1 score of 85.70%. Furthermore, Jahan et al. [23] used transliterated Bengali text and text that was code-mixed in Bengali and English to classify social media users' comments into abusive and non-abusive categories. They employed unigrams, bigrams, the number of likes, emojis, and their categories as input features. To detect abusive speech, the authors used three machine-learning classifiers SVM, RF, and Adaboost. The best performance score was an accuracy of 72.14%. From the review works, the machine learning approach, SVM, has performed better than the deep learning approaches. This is could be due to smaller datasets. In this research, we aim to investigate the performance of SVM and variants of CNN models such as single-channel CNN and Multichannel CNN on the transliterated hate speech dataset merged with an existing dataset. We want to assess the impact of the transliterated hate speech dataset on model performance.

3. Methods

3.1. *Data Sets*

To address the research question, "To what extent does the transliteration system increase the performance of CNN-based hate speech detection?" we utilized both transliterated Amharic hate speech datasets and an existing Amharic hate speech dataset. The transliterated Amharic hate speech dataset was manually collected from the Ethiopian Broadcasting Corporation (EBC) Facebook page. Additionally, we merged the transliterated datasets with the existing Amharic dataset to evaluate the impact of dataset composition on classifier performance. Table 2 summarizes the dataset distribution across different classes and configurations.

- Manually Transliterated Dataset (MT): We collected 1,000 Amharic translit-

erated comments from the EBC Facebook page. Of these, 391 were labeled as Hate, and 609 were labeled as Not Hate. An 80:20 split was applied for training and testing, resulting in 80% of the dataset being used for training and 20% for testing. Manual transliteration was performed to compare the performance of hate speech detection systems using manually transliterated data against those utilizing auto-transliterated data.

- Auto-Transliterated Dataset (AT): The 1,000 manually transliterated comments were converted back into their original Amharic script using a Latin-to-Amharic transliteration tool. The class distribution and training/testing split for this dataset mirrored those used for the manually transliterated dataset.
- Amharic Hate Speech Dataset (AmHD): This dataset was obtained from a public repository on Zenodo. It contains 1,000 instances labeled as Hate and 1,000 instances labeled as Not Hate. The dataset was split into 800 training instances and 200 testing instances for each class, adhering to an 80:20 ratio.
- Amharic Hate Speech Dataset Merged with Manually Transliterated Dataset (AmHD+MT): To evaluate the effect of increasing dataset size on classifier performance, we merged the AmHD dataset with the manually transliterated dataset. The merged dataset contains a total of 3,000 instances, comprising 1,391 instances in the Hate class and 2,809 instances in the Not Hate class. The dataset was split into training and testing subsets using the same 80:20 ratio.
- Amharic Hate Speech Dataset Merged with Auto-Transliterated Dataset (AmHD+AT): Similarly, we merged the auto-transliterated dataset with the AmHD dataset to assess classifier performance on a larger dataset containing auto-transliterated instances. The class proportions and training/testing splits were consistent with those used in the manually transliterated merged dataset. This methodological approach allowed us to systematically evaluate the impact of transliteration and dataset merging on hate speech detection performance. The class distribution and training and testing data split of each dataset is shown in Table 2.

Table 2. Number of instances in each dataset

Data split	**Class**	**Datasets**				
		MT	**AT**	**AmHD**	**AmHD+ MT**	**AmHD+ AT**
Train	Hate	311	311	800	1,111	1,111
	notHate	489	489	800	1,289	1,289
Test	Hate	80	80	200	280	280
	notHate	120	120	200	320	320
Total		1,000	1,000	2000	3,000	3,000

3.2. *Data Preprocessing*

In the data preprocessing stage, punctuation, URLs, unused white space, and Amharic characters are all removed. Using the RBLatAm tool [18], we transliterated the Latin-based comments into the Amharic script. Since there are several habits to write the same Amharic words for some letters, such as the variants of "**ሀ፣ ሰ፣ አ**" etc., character regularization is also done after transliteration. However, we did not remove stop words for dimension reduction. Because it has a substantial meaning in the identification of hate speech. For instance, the phrase "**ሁሉንም እምነት አልባ ሰዎች ግደል**" ("kill all non-believers"). The stop word "**ሁሉንም**" ("all") is crucial in classifying the sentence as hate speech. Word based n-gram features can help us represent this concept.

3.3. *Feature Representation*

Machine-learning algorithms cannot learn classification rules unless the raw texts are transformed into numerical features. As a result, feature extraction is a crucial stage in text classification. In order to express the raw text in numerical representations, this step is utilized to extract the essential elements from it. In this experiment, we use both character and word-based n-grams for the SVM classifier and word2vec feature-engineering techniques for the CNN-based classifiers.

3.3.1. *N-gram based feature selection*

Based on prior research for text classification, we use character and word-based n-grams as features and pass their TFIDF (term frequency-inverse document frequency) values to the SVM model, which is used as a baseline. We conduct a comparative analysis while taking into account various n values in the model. In this experiment, we test the SVM classifier using a unigram (n=1), a bigram (n=2), and a combination of the two for the word-based n-grams. For the character-based n-gram, we use (n=1,2,3,4). During the experiment for word-based n-gram increasing the n does not add any improvements for the classifier while in the character-based n-gram increasing the n value adds improvements. (See Table 4).

3.3.2. *Word2vec feature learning*

Recently, there are efforts are made to create word2vec models for the Amharic language such as FastText [24]. However, the model is not sufficient for hate speech detection issues because hate speech contents posted by users have its own unique characteristics that are not seen in the standard texts. For instance, the word "Thank you," which is frequently used by Facebook users, is abbreviated as "10Q." Similarly to this, users of Amharic social media leave out the letters "**አሀ***" for "**አህያ**" (in English "Donkey") while writing hateful sentiments for the insulting expressions. Therefore, we employ the continuous bag of words (CBOW) word-embedding model to produce features for our hate speech detection system in order

to take the meaning of such words into account in the feature space and prevent out-of-vocabulary issues.

3.4. *Support Vector Machine (SVM)*

Support vector machines (SVMs) are supervised learning models that evaluate data for regression and classification [25]. An SVM training algorithm creates a model that categorizes new samples into one of two categories when given a series of training examples, converting it into a non-probabilistic binary linear classifier. SVM maximizes the distance between the two classes by mapping training examples to points in space. Then, based on which side of the gap they fall, new samples are projected into that same area and predicted to belong to a class. A method for two-class and multi-class classification is the LIBSVM kernel for support vector classification (SVC) [26]. SVM with a linear classifier and TF-IDF parameters are used in the experiment as a baseline. As a baseline, we use a linear Support Vector Machine (SVM) classifier because it has shown effective performance in previous studies using TFIDF-weighted bag-of-word features [27]. Further, we use the grid search optimization strategy to select the best parameters for each dataset. We use the python sci-kit-learn library to implement the classification model.

3.5. *Convolutional Neural Network (CNN)*

CNN models were initially developed for computer vision, but they have now been proven to be successful for NLP and produce outstanding results [28]. CNN is made to learn features automatically and is adaptable. The three basic building blocks of CNN are convolution, pooling, and fully connected layers. The third, a fully connected layer, transfers the extracted features into a final output, such as classification, while the first two, convolution and pooling, do feature extraction [29]. For the CNN-based experiments, we have used the CNN models proposed in [7] since it is a continuation of hate speech detection research from social media using transliterated Amharic comments.

3.5.1. *Single channel CNN-1 (SC-CNN-1)*

The first single channel CNN (SC-CNN-1) model is described as having a kernel size of 1, 2, 3, 4, and 5 for each of the five datasets, an embedding size of 100, a convolutional filter size of 8, 16, and 32, and an activation of ReLu with a dropout rate of 0.5. The maximum pooling size is 2. Two classes are represented by the Dense 2 output layer, which has a sigmoid activation function. Table 3.3 displays each one of the model's distinct configurations. The model is trained using a validation split of 0.1, 16-epoch iterations, and a 20-batch size.

3.5.2. *Single channel CNN-2 (SC-CNN-2)*

The second single-channel CNN (SC-CNN-2) model is stated as having an embedding size of 100, convolutional filter sizes of 8, 16, and 32, and activation of ReLu with a dropout rate of 0.5 for each of the five datasets. A pooling size of two is the maximum. The Dense 2 output layer, which has an activation function with a sigmoid, represents two classes. Every single model's unique configuration is shown in Table 5. The model is trained with 20 batches, 16 epochs, and a validation split of 0.1.

3.5.3. *Multichannel CNN (MC-CNN)*

We perform experiments by defining the MC-CNN model by concatenating the two single-channel CNN models mentioned above in order to compare the behavior of the unified feature of the multichannel CNN model parameters to the single-channel CNN models in each of the three hate speech datasets. With two Conv layers and an embedding layer with an embedding size of 100, we create the MCCNN model. ReLu activates the Conv layer. Dense 2 is the output layer, and it has two classes represented by a sigmoid activation function. We employ the same validation split of 0.1, epochs of 16, and batch sizes of 20 for training the MC-CNN model. Each of the five datasets' filters and kernel sizes is defined in Table 5.

3.5.4. *Evaluation*

The performances of the proposed model classifiers using the test dataset are evaluated by recording the statistics of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Three performance metrics are used to evaluate the classifiers. These are recall, precision, and F-measures [30].
Recall (R): the proportion of actual positives, which are predicted positive.

$$Recall = \frac{TP}{TP+FN} \tag{1}$$

"Precision (P): the proportion of predicted positives which are actually positive."

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

"F-measure (F1): the harmonic mean of precision and recall."

$$F-measure = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \tag{3}$$

3.6. *Experimental results and discussion*

In these experiments, we have used five datasets. Manually transliterated Amharic hate speech datasets(MT), Auto-transliterated Amharic hate speech datasets (AT), existing Amharic hate speech datasets (AmHD), the combination of manually transliterated and existing Amharic hate speech (AmHD+MT) and the

auto transliterated combined with the existing Amharic hate speech dataset (AmHD+AT). Each class distribution and training and testing data split are shown in Table 3.

Table 3. Datasets used in the experiment

Data split	**Classes**	**MT**	**AT**	**AmHD**	**AmHD+MT**	**AmHD+AT**
Train	Hate	311	311	800	1,111	1,111
	notHate	489	489	800	1,289	1,289
Test	Hate	80	80	200	280	280
	notHate	120	120	200	320	320
Total		1,000	1,000	2000	3,000	3,000

Table 3 provides a detailed breakdown of the datasets used in the experiment, including their composition and split into training and testing subsets across different configurations. The datasets consist of manually transliterated (MT), auto-transliterated (AT), formal Amharic script (AmHD), and combinations of formal script with transliterated datasets (AmHD+MT and AmHD+AT). Each dataset is categorized into two classes: Hate and NotHate, and is divided in an 80:20 ratio for training and testing purposes. The training data for the hate class includes 311 instances for both MT and AT datasets, 800 instances for the AmHD dataset, and 1,111 instances for the combined AmHD+MT and AmHD+AT datasets. Similarly, the notHate class contains 489 instances for MT and AT, 800 instances for AmHD, and 1,289 instances for both combined datasets. For the testing subset, the hate class comprises 80 instances for MT and AT datasets, 200 instances for AmHD, and 280 instances for the combined datasets. The notHate class in the testing set includes 120 instances for MT and AT, 200 instances for AmHD, and 320 instances for AmHD+MT and AmHD+AT. In total, the MT and AT datasets each contain 1,000 instances, the AmHD dataset comprises 2,000 instances, and the combined datasets (AmHD+MT and AmHD+AT) include 3,000 instances each. This table underscores the dataset configurations and their respective sizes, which were critical for evaluating the performance of the hate speech detection models.

3.6.1. *Results of the SVM classifier*

The results of the SVM classifier on the "MT," "AT," "AmHD," "AmHD+MT," and "AmHD+AT" datasets, utilizing both character-based and word-based n-gram features, are summarized in Table 4. For the "MT" dataset, character 2-grams (ch-2g) achieved the highest F1 score of 0.72, indicating their effectiveness in handling the inconsistencies present during script transliteration. In the "AT" dataset, character 3-grams (ch-3g) yielded the best performance with an F1 score of 0.65,

showcasing their ability to capture the nuances of auto-transliterated texts.

Table 4. F1-score of the SVM classifier on each datasets

Features	**MT**	**AT**	**AmHD**	**AmHD+MT**	**AmHD+AT**
ch-1g	0.65	0.61	0.82	0.76	0.66
ch-2g	**0.72**	0.62	**0.90**	0.86	0.84
ch-3g	0.71	**0.65**	0.89	0.86	**0.86**
ch-4g	0.69	0.63	0.87	**0.88**	0.84
w-1g	0.71	0.61	0.87	0.87	0.84
w-2g	0.57	0.48	0.86	0.69	0.77
combined	0.73	0.65	0.86	0.86	0.83

The "AmHD" dataset, which originates from a specific domain, exhibited the highest classification performance across all datasets. Here, character 2-grams (ch-2g) achieved an F1 score of 0.90, demonstrating their strength in identifying hate speech patterns. For the "AmHD+MT" dataset, the classifier performed best with character 4-grams (ch-4g), achieving an F1 score of 0.88. This suggests that the increased context captured by longer n-grams benefits the detection of hate speech in merged datasets. Similarly, for the "AmHD+AT" dataset, character 3-grams (ch-3g) delivered the best results with an F1 score of 0.86. Although the combined datasets (AmHD+MT and AmHD+AT) were larger, the results highlight the challenges posed by nonstandard script transliteration variations, which may affect the effectiveness of the classifier. These findings underscore the importance of selecting appropriate n-gram features for SVM classifiers, particularly in handling datasets with transliteration inconsistencies.

3.6.2. *Results of the Single channel CNN*

The performance of the single-channel CNN model was evaluated using various filter sizes and kernel dimensions across five datasets: auto-transliterated (AT), manually transliterated (MT), the existing Amharic hate speech dataset (AmHD), the combination of AmHD with MT (AmHD+MT), and the combination of AmHD with "AT" (AmHD+AT).

The results, summarized in Table 5, highlight the impact of filter and kernel size on the model's ability to detect hate speech effectively. For the "AT" dataset, the model achieved its best performance with a filter size of 8 and a kernel size of 4, attaining an F1 score of 0.436. In the MT dataset, the optimal configuration was a filter size of 16 and a kernel size of 1, resulting in an F1 score of 0.393. On the AmHD dataset, the CNN performed best with a filter size of 32 and a kernel size of 4, achieving an F1 score of 0.783, the highest recorded score across all datasets and configurations.

Table 5. F1-score of the single channel CNN based classifier on each datasets.

Filters	**Kernel size**	**AT**	**MT**	**AmHD**	**AmHD+MT**	**AmHD+AT**
8	1	0.313	0.302	0.589	0.590	0.591
	2	**0.375**	0.333	0.741	0.623	0.591
	3	0.368	0.298	0.737	0.590	0.551
	4	**0.436**	0.338	0.712	0.639	0.555
	5	0.278	0.280	0.707	0.584	0.563
16	1	0.256	**0.393**	0.726	0.592	0.530
	2	0.300	**0.333**	0.734	0.579	0.595
	3	0.376	0.271	0.736	0.614	0.608
	4	0.350	0.283	0.755	0.620	0.556
	5	0.260	0.243	0.752	0.558	0.523
32	1	0.363	0.240	0.718	0.617	0.585
	2	0.341	0.316	0.757	0.630	**0.661**
	3	0.353	0.317	0.755	0.626	**0.662**
	4	0.327	0.319	0.**783**	**0.632**	0.562
	5	0.269	0.250	0.**769**	**0.644**	0.553

When combining the datasets to analyze the effect of increased data size, the model's performance varied. For the "AmHD+MT" dataset, the best F1 score of 0.644 was observed with a filter size of 32 and a kernel size of 5. Similarly, on the "AmHD+AT" dataset, the highest F1 score of 0.662 was achieved with a filter size of 32 and a kernel size of 3. These results suggest that while combining datasets can improve overall performance, the model's sensitivity to filter and kernel size remains critical for achieving optimal results. In summary, the single-channel CNN model's performance across datasets and configurations demonstrates that selecting appropriate filter and kernel sizes significantly influences the detection accuracy. The AmHD dataset stands out, where the model's F1 scores of 0.783 (filter size 32, kernel size 4) and 0.769 (filter size 32, kernel size 5) indicate its effectiveness in identifying hate speech in a datasets where all the scripts are normal and standard. Table 5 provides a detailed breakdown of the F1 scores for all configurations and datasets.

3.6.3. *Results of the multichannel CNN*

The performance of the SC-CNN-1, SC-CNN-2, Multi-Channel CNN (MC-CNN), and Support Vector Machine (SVM) models was evaluated across five datasets: auto-transliterated (AT), manually transliterated (MT), the existing Amharic hate speech dataset (AmHD), the combined "AmHD" and "MT" dataset (AmHD+MT), and the combined "AmHD" and "AT" dataset (AmHD+AT). The F1 scores of these models are summarized in Table 6.

Table 6. Results of the single, multichannel CNN and SVM models

Models	**AT**	**MT**	**AmHD**	**AmHD+MT**	**AmHD+AT**
SC-CNN-1	0.375	0.393	0.783	0.632	0.661
SC-CNN-2	0.436	0.333	0.769	0.644	0.662
MC-CNN	0.280	0.350	**0.810**	0.530	0.630
SVM	**0.720**	**0.650**	**0.900**	**0.880**	**0.860**

The SC-CNN-1 model achieved its best performance on the AmHD dataset, recording an F1 score of 0.783, while on the AmHD+MT and AmHD+AT datasets, it performed with F1 scores of 0.632 and 0.661, respectively. Similarly, the SC-CNN-2 model showed its highest performance on the AmHD dataset with an F1 score of 0.769, slightly lower than SC-CNN-1. However, it achieved relatively better results on the AmHD+MT and AmHD+AT datasets with F1 scores of 0.644 and 0.662, respectively. The MC-CNN model demonstrated strong performance, particularly on the AmHD dataset, achieving an F1 score of 0.810. However, its performance was lower on the AT, MT, AmHD+MT, and AmHD+AT datasets, with F1 scores of 0.280, 0.350, 0.530, and 0.630, respectively. Finally, the SVM model outperformed all CNN-based models across all datasets. It achieved the highest F1 score of 0.900 on the AmHD dataset, showing its effectiveness in handling the features of this dataset. On the AmHD+MT and AmHD+AT datasets, SVM recorded F1 scores of 0.880 and 0.860, respectively, further highlighting its robustness.

4. Discussion and analysis

The primary objective of this research was to evaluate the impact of auto-transliterated dataset size on model performance, compare the performance of various machine learning models—such as the MC-CNN model versus single-channel CNN models—and analyze the performance of the SVM classifier relative to CNN variants on both auto-transliterated and manually transliterated hate speech datasets. The experiments were conducted on five datasets: "AT," "MT," "AmHD," "AmHD+MT," and "AmHD+AT."

When comparing the performance of single-channel models (SC-CNN-1 and SC-CNN-2) with that of the multichannel CNN (MC-CNN) model, the results showed that the MC-CNN outperformed the single-channel models on the AmHD dataset. Specifically, the MC-CNN achieved an F1-score of 0.810, compared to 0.783 and 0.769 for SC-CNN-1 and SC-CNN-2, respectively. This performance is likely due to the domain-specific nature of the AmHD dataset and normal Amharic scripts, which focuses on religion, ethnicity, and racial hate speech. These focused domains enable classifiers to detect hate speech with higher accuracy. In contrast, datasets collected from broader, less specific domains introduce variability that challenges classifier performance. The MC-CNN model performed better because it could learn richer features from the combined channels than from individual single-channel features.

The second experiment examined the effect of increasing the size of the transliterated Amharic hate speech dataset on classifier performance. Experimental results indicated that combining auto-transliterated hate speech datasets with existing datasets did not improve performance for either CNN models or SVM classifiers. This outcome may stem from the domain-specific nature of the datasets: the transliterated datasets primarily originate from the political domain, which lacks consistent terminology for classification. Despite the larger data sizes of "AmHD+AT" and "AmHD+MT" compared to "AmHD," all models (single-channel CNNs, MC-CNN, and SVM) demonstrated better performance on the smaller, domain-specific "AmHD" dataset. This suggests that combining transliterated datasets negatively impacts classifier performance due to variability in data source domains. Consequently, transliterated datasets should be treated separately and focused on specific domains for effective hate speech detection. Additionally, the SVM classifier consistently outperformed CNN variants across all five datasets, aligning with findings from previous studies[21]. Research has shown that machine learning models often excel with smaller datasets, while deep learning models require larger datasets to achieve superior performance[22]. Although increasing dataset size generally enhances deep learning model performance, this was not observed in our study. This discrepancy is likely attributable to the small size of the original dataset and domain differences among transliterated datasets, which can adversely affect model performance.

In practical applications of hate speech detection using transliterated datasets, separate treatment of datasets is essential. Our findings indicate that merging transliterated hate speech datasets with existing datasets does not improve model performance and, in fact, may degrade it. Therefore, transliterated datasets should be domain-specific and analyzed independently for optimal performance.

5. Limitations

The transliterated social media comments used in this study were collected from a wide range of sources, encompassing diverse domains. In contrast, earlier datasets on Amharic hate speech predominantly focused on specific topics such as politics, religion, and ethnicity. This variation in domain topics could affect classifier performance, as terms from certain domains may be underrepresented while others dominate, leading to class imbalances that can confuse the classifiers. Therefore, future studies should prioritize collecting datasets from consistent domains, whether written in formal Amharic scripts or transliterated scripts, to ensure a more balanced dataset and enable more accurate evaluation of model performance.

6. Conclusions

This research aimed to detect hate speech from transliterated Amharic social media comments using both machine learning and deep learning approaches. We evaluated the performance of hate speech detection models on auto-transliterated, manually

transliterated, and combined datasets that merged transliterated data with existing Amharic hate speech datasets. Our findings demonstrate that the SVM classifier with a character n-gram of two outperformed the CNN-based models. Additionally, merging the transliterated datasets with existing Amharic hate speech datasets to increase dataset size did not lead to performance improvements for either SVM or CNN-based classifiers. This lack of improvement is likely due to the datasets originating from different domains, which introduces inconsistencies that challenge the classifiers. Based on these experimental findings, we recommend that hate speech detection using transliterated datasets should be conducted separately, ensuring datasets are domain-specific to achieve optimal model performance.

References

1. Brusilovskiy E, Townley G, Snethen G, Salzer MS. Social media use, community participation and psychological well-being among individuals with serious mental illnesses. Computers in Human Behavior 2016;65:232–40. https://doi.org/10.1016/J.CHB.2016.08.036.
2. Sumikawa Y, Jatowt A. Analyzing History Related Posts in Twitter. International Journal on Digital Libraries 2021;22:105–134. https://doi.org/https://doi.org/10.1007/s00799-020-00296-2.
3. Fernando B, Gilbert FD, Pius von D, Mark C. TRANSLIT: A Large-scale Name Transliteration Resource. 12th Conference on Language Resources and Evaluation, Marseille: European Language Resources Association (ELRA); 2020, p. 3265–3271.
4. Yaqob. D. No Title. Transliteration on the Internet: The Case of Ethiopic 1997;1:17–41.
5. Mossie Z, Wang J-H. Social Network Hate Speech Detection for Amharic Language. 4th International Conference on Natural Language Computing (NATL 2018), Dubai, UAE, 2018, p. 41–55. doi.org/10.5121/csit.2018.80604.
6. Mossie Z, Wang JH. Vulnerable Community Identification using Hate Speech Detection on Social Media. Information Processing and Management 2020;57:102087. https://doi.org/10.1016/j.ipm.2019.102087.
7. Zeleke A, Andreas R, Solomon A. Design and Implementation of a Multichannel Convolutional Neural Network for Hate Speech Detection in Social Networks. Revue d'Intelligence Artificielle n.d.;36:175–83. https://doi.org/https://doi.org/10.18280/ria.360201.
8. Zeleke A, Andreas R, Solomon A. Multi-Channel Convolutional Neural Network for Hate Speech Detection in Social Media. 9th International conference on the Advancement of Science and Technology, M. L. Berihun (Ed.): ICAST 2021, LNICST 411: ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2022 Published by Springer Nature Switzerland AG 2022.; 2022, p. 603–618, doi.org/10.1007/978-3-030-93709-6_41.

9. Davidson SA and T. Identifying hate speech in social media. XRDS:Crossroads, The ACM Magazine for Students n.d.;24:56–9. https://doi.org/10.1145/3155212.
10. Brown A. What is hate speech? Part 1: The Myth of Hate. Law and Philosophy 2017;36:419–68. https://doi.org/10.1007/s10982-017-9297-1.
11. Ward KD. Free Speech and the Development of Liberal Virtues: An Examination of the Controversies Involving Flag-Burning and Hate Speech. U Miami L Rev 1998;733.
12. Bayer J, Bárd P. Hate speech and hate crime in the EU and the evaluation of online content regulation approaches. Policy Department for Citizens' Rights and Constitutional Affairs 2020.
13. Clark KB. Dark Ghetto: Dilemmas of Social Power. Wesleyan University Press; 1989.
14. Waldron J. Hate Speech and Political Legitimacy. Cambridge University Press; 2012. https://doi.org/10.1017/CBO9781139042871.022.
15. Faggian DD and A. Where do angry birds tweet? Income inequality and online hate in Italy. Cambridge Journal of Regions, Economy and Society 2021;14:483–506. https://doi.org/10.1093/cjres/rsab016.
16. Spilioti T. From transliteration to trans-scripting: Creativity and multilingual writing on the Internet. Discourse, Context Media 2019;29. https://doi.org/https://doi.org/10.1016/j.dcm.2019.03.001.
17. Munye M, Atnafu S. Amharic-English Bilingual Web Search Engine. Proceedings of the International Conference on Management of Emergent Digital EcoSystems, Addis Ababa, Ethiopia: Association for Computing Machinery; 2012, p. 32–9. https://doi.org/10.1145/2457276.2457284.
18. Kore M, Goyal V. Machine Transliteration for English to Amharic Proper Nouns. International Journal of Computer Science Trends and Technology (IJCST) 2017;5:23–31.
19. [19] Tedla T. AmLite: Amharic Transliteration using Key Map Dictionary. ArXiv 2015.
20. Abebaw Z. Latin-to-Amharic script transliteration. Zenodo 2022. https://zenodo.org/record/7317713#.Y7WiAHZBy1s.
21. Sazzed S. Abusive content detection in transliterated Bengali-English social media corpus. Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching, pages Linguistics, Association for Computational Linguistics; n.d., p. 125–130.
22. A. Al Taawab, L. Tasnia MD and MHKM. Transliterated Bengali Comment Classification from Social Media 2022:365–71. https://doi.org/10.1109/R10-HTC54060.2022.9929514.
23. M. Jahan, I. Ahamed MRB and SS. Abusive Comments Detection in Bangla-English Code-mixed and Transliterated Text. 2019 2nd International Conference on Innovation in Engineering and Technology (ICIET), 2019, p. 1–6. https://doi.org/10.1109/ICIET48527.2019.9290630.

24. Eshetu A, Teshome G, Abebe T. Learning Word and Sub-word Vectors for Amharic (Less Resourced Language). International Journal of Advanced Engineering Research and Science 2020;7:358–66. https://doi.org/10.22161/ijaers.78.39.
25. Boser BE, Guyon IM, Vapnik VN. Training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, 1992, p. 144–52. https://doi.org/10.1145/130385.130401.
26. Chang CC, Lin CJ. LIBSVM: A Library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2011;2:1–27. https://doi.org/10.1145/1961189.1961199.
27. Gaydhani A, Doma V, Kendre S, Bhagwat L. Detecting Hate Speech and Offensive Language on Twitter Using Machine Learning: An N-Gram and TFIDF Based Approach. CoRR 2018;abs/1809.0.
28. Collobert R. Deep learning for efficient discriminative parsing. Journal of Machine Learning Research, vol. 15, 2011, p. 224–32.
29. Georgakopoulos S V., Tasoulis SK, Vrahatis AG, Plagianakos VP. Convolutional neural networks for toxic comment classification. ACM International Conference Proceeding Series (2018), 2018, p. 1–6. https://doi.org/10.1145/3200947.3208069.
30. Seliya N, Khoshgoftaar TM, Van Hulse J. A study on the relationships of classifier performance metrics. Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, 2009, p. 59–66. https://doi.org/10.1109/ICTAI.2009.25.