# A CONCEPTUAL ENHANCEMENT OF LSTM USING KNOWLEDGE DISTILLATION FOR HATE SPEECH DETECTION

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY

IN

**SOFTWARE ENGINEERING**

Submitted by:

**AKILENG ISAAC**

**Roll No: 2K19/SWE/17**

Under the Supervision of

**Dr ARUNA BHAT**

**DEPARTMENT OF SOFTWARE ENGINEERING**
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

**JUNE, 2021**

**DELHI TECHNOLOGICAL UNIVERSITY**
(FORMERLY DELHI COLLEGE OF ENGINEERING)
Bawana Road, Delhi-110042

## CANDIDATE'S DECLARATION

I, Akileng Isaac, Roll No 2K19/SWE/17, a student of M.Tech in Software Engineering, declare that the Project Report titled "A Conceptual Enhancement of LSTM Using Knowledge Distillation For Hate Speech Detection" which I submit to the Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirements for the award of the degree of Master of Technology is original and not copied from any source without proper citation. This work has not previously formed the basis for any Degree, Diploma Associateship, Fellowship, or other similar title or recognition.

Place: Delhi

Date:28/06/2021……………………

**Akileng Isaac**

**STUDENT**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

## CERTIFICATE

I hereby certify that the Project Report titled "A Conceptual Enhancement of LSTM Using Knowledge Distillation For Hate Speech Detection," which is submitted by Akileng Isaac, Roll No 2K19/SWE/17 Department of Software Engineering, Delhi Technological University, Delhi, in partial fulfilment of the requirement for the award of the degree of Master of Technology is a record of the project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or complete for any Degree or Diploma to this University orelsewhere.

Place: Delhi

Date: 30/06/2021

**Dr Aruna Bhat**

**SUPERVISOR**

# ABSTRACT

Hate speech is an issue to most governments and the public's concern due to the increased emergence of social media platforms and the increasing use of such media to disseminate hate speech to individuals, groups of persons, communities, or races. Hate speech is also by no means always on the rise due to the high rate of remote service usage such as communication, online studies, meeting, dating, etc. With the recent outbreak of COVID-19, there has been an increase in the number of users on different social media platforms. This increase in number has brought about an increase in issues such as hate speech, among others. Therefore without detection and analysis of hate speech, one cannot imagine social media to be free of malicious content. Deep Neural networks inspired by the human brain's work have continuously demonstrated their importance and relevancy in many different I.T. fields, particularly hate speech detection. This research aims to provide a detailed process of improving LSTM used for hate speech detection using knowledge distillation. The knowledge transfer is done from the more extensive network (teacher) to the smaller student network. The teacher has trained for five full epochs to output accuracy of 76.8%, the student network trained from the teacher network for three entire epochs attained an accuracy of 82.6%. Another student model cloned and trained from scratch for three full epochs instead of the teacher network achieves an accuracy of 75.4%.

# ACKNOWLEDGEMENT

First, I would like to extend my sincere gratitude to my mentor/supervisor, Dr Aruna Bhat, for her constant encouragement and insight. I thank her for suggesting the initial idea of applying knowledge distillation to improve the working on a deep neural network, which eventually formed the core of this dissertation. I greatly appreciate her constant motivation to understand the Machine Learning concepts that I wasn't familiar with. This work would not have been accomplished without her support.

The faculty members of Software Engineering played an extensive role in preparing me to pursue this research. Without knowing the various core courses they taught me, I would not have acquired the technical knowledge of undertaking such a project. Therefore, on this note, I sincerely thank them for preparing me for the journey that has successfully come to its completion.

I was fortunate enough to be part of a vibrant class of members who were a great pillar in my Masters' studies. I am grateful for the academic support they rendered to me during the two years of vigorous training. Working with classmates from various backgrounds also enabled me to develop a true sense of awareness of equality, diversity, and inclusion

I want to thank my family, especially my parents, Mr Akileng Johnson and Mrs Akileng Hellen, for supporting every step of my academic journey. I will always be grateful for the advice and support they have given me over the years.

I also take this honour to sincerely appreciate the Indian Government through Indian Council for Cultural Relations (ICCR), my sponsoring organization, for funding my Master's Education.

Above all, I thank the Almighty God for enabling me to accomplish this project.

## Table of Contents

## LIST OF TABLES

## LIST OF FIGURES

# LIST OF ABBREVIATIONS AND NOMENCLATURE

| | |
|---|---|
| LSTM | Long Short Term Memory |
| RNN | Recurrent Neural Network |
| GRU | Gated Recurrent Unit |
| KNN | K-Nearest Neighbor |
| NLP | Natural Language Processing |

# CHAPTER 1: INTRODUCTION

## 1.0 Chapter Summary

In this chapter, the introductory description of the topic of study has been presented. This is followed by the research question that guided the research and the objectives. The problem being addressed has been formulated in the next section.

## 1.1 Introduction

In today's increasing technological advancements, especially in the communication and information exchange platform, there is an increase in expressions of hate and violence towards an individual or a specific group of people. [10]

Hate speech over the years has been a growing concern that can't be overlooked anymore. Whether it is towards an individual or group of people, the need for hate speech detection is vastly growing and needed. The recent pandemic has seen many people hooked onto different social platforms to communicate with their loved ones, make money, provide entertainment, etc. The growth and diversity of many ethnic groups have made hate speech detection even more necessary.

It is also evident that hate speech detection and analysis are needed and essential aspects of today's social media era such as Facebook, Twitter, YouTube, etc. These platforms play an indispensable role in the lives of different users ranging from studies, entertainment, business advertising, marketing, communication, etc., which makes them an easy target for discrimination, hatred, and violence. [10]

Different approaches have been used in various studies, such as Lexicon based approach to extract features from text data which were to be used in the machine learning models; both supervised learning, and unsupervised learning algorithms were used to build the classifier models and then results of the models were compared and analyzed. Hashing vectorizer and Term Frequency Inverse Document Frequency (TF-IDF) were also used together with different supervised machine learning models (Naive Bayes Classifier, Decision Tree, Support Vector Machine) and unsupervised techniques K-Nearest Neighbors.

In other unsupervised machine learning techniques, using graph, sentiment, and emotion analysis techniques, clustering and analysis of posts on prominent Facebook pages was done and consequently identify the pages that automatically promote hate speech in the comment sections regarding sensitive topics.

Another approach in unsupervised learning uses a combination of a popular topic modelling technique, i.e. Latent Dirichlet Allocation (LDA) and an unsupervised machine learning technique, i.e. self-organizing maps (SOM), to analyze hate speech spread over social media. This method was later compared to K-means clustering used after the application of LDA.

Previous studies have been done using different deep neural networks such as Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM). The use of neural networks has shown promising results in hate speech detection and room for even better results and improvements. Since neural networks require a lot of training time, computational power, large datasets, etc., improving this has been vital.

Knowledge distillation has, over the years, been a primary focus for improving the work of neural networks. The use of this concept has proven improvement in results. This research utilizes this same concept to improve the LSTM neural network's working in hate speech detection. We create both the teacher network and the student network, which the teacher will train. The student's initial clone is made before the teacher trains it, and then it's trained from scratch. The performance results obtained from these different models are then compared.

## 1.2 Research Questions and Objectives

### Research Questions

(1) Can the performance of a neural network be improved?

(2) Can we apply the concept of knowledge distillation to a neural network for hate speech detection?

### Objectives

(1) To perform a systemic survey on hate speech detection using different techniques.

(2) To improve the performance of a neural network using knowledge distillation.

(3) To apply the concept of knowledge distillation to LSTM.

(4) Evaluate the model based on accuracy and perform the training and testing on different models: teacher network, student network, and the student network trained from scratch.

## 1.3 Problem Formulation

Neural networks have several challenges, such as the time taken to train the network, the processing power required for training or testing, the complexity of more extensive networks, to mention but a few. Based on the above, there is a need to have a relatively more minor network to reduce the challenges mentioned above and improve its performance, in this case, in hate speech detection.

Previous studies on the use of Knowledge distillation of different areas of applications of neural networks have shown significant improvement in the use of this concept.

Hate speech is one area where neural networks have performed significantly and are among the world's biggest concerns today, giving a basis for researching this area/topic.

Additionally, there hasn't been any precise mechanism of applying knowledge distillation to hate speech detection using any neural networks, even in their most straightforward implementation.

Therefore, this research provides a comprehensive approach to improving a neural network's performance (LSTM) for Hate speech detection trained and tested over a publicly available dataset.

# CHAPTER 2: BACKGROUND

## 2.0 Chapter Summary

This chapter gives brief notes about the different concepts one needs to know while reading this research as this will provide a better understanding and insist throughout this research.

## 2.1 Hate Speech

Speech or any textual message that incites violence directed towards an individual or specific group of persons is deemed hate speech. Hate speech can be in various areas like religion, gender, colour, citizenship, disability, race, sexual orientation, etc.

In a rather legal context, prohibited writing, words, performance, or behaviour that can insult or incite violence from either party (victim or offender) is considered and regarded as hate speech. [1]

## 2.2 RNN

RNN is "a type of artificial neural network designed to recognize patterns in sequences of data such as words and handwriting."

Unlike the feed-forward networks whose outputs are unrelated, RNN works with sequential inputs, making it a variation of the feed-forward networks. They store the information from the previous timestamp and feed it as an input for the next step. From Figure 1, the data from 't-1' together with input 't' is provided as input to the next step, and similar information from input 't' and input 't+1' is fed to the next step.

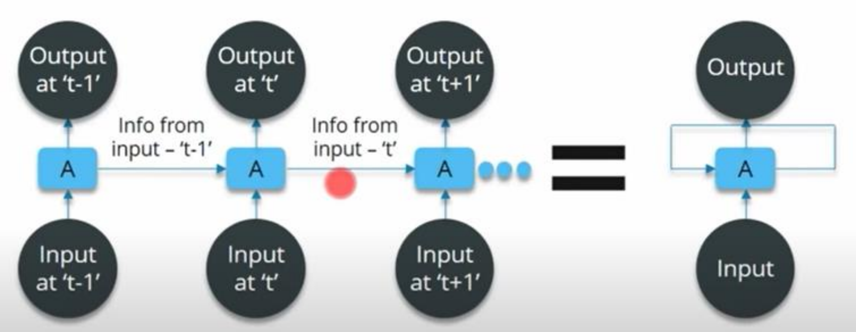


*Fig. 1: Structure of RNN [6]*

RNNs are good & work well with dealing with sequential data. However, training becomes a challenge when using the backpropagation algorithms due to the vanishing gradient problem [6] [7]. During the training of the network, this problem arises when the gradient propagated grows exponentially or decays. Some traditional RNN units such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) demonstrate valuable solutions to this problem.

## 2.3 LSTM

Long Short-Term Memory has a unit used or acts as an accumulator of information of a particular state [6] [8]. This makes it a variation of the traditional RNN. Three gates (input gate, forget gate, output gate) control the unit's operations (memory cell).

The input gate activates, the information is accumulated at the cell. The forget gate controls the data from the previous data, which gets ignored once the gate activates. Then finally, the output of the unit/cell is governed by the output gate.

This gate concept is instrumental in preventing the vanishing gradient problem faced by the traditional RNN [8]. The use of multiple layers at each timestamp in LSTM has proven much more effective than the ordinary RNN. [1][5]

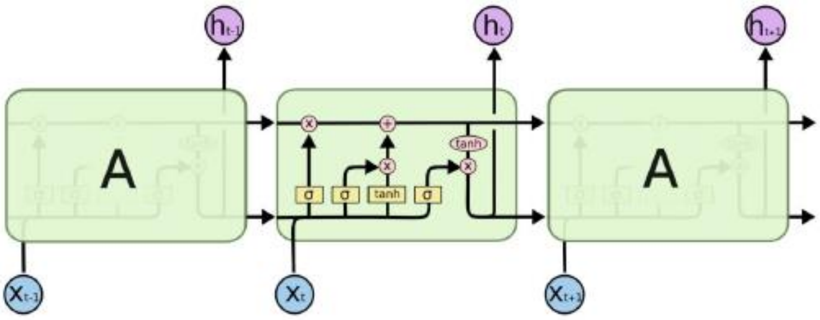


*Fig. 2: Structure of LSTM Network [1]*

## 2.4 Knowledge Distillation

Knowledge distillations help distil the information or knowledge from an extensive neural network to a much smaller network. This process has been inspired by how humans live and keep learning every day, such as learning from the teacher.  The teacher is the most informed entity that helps transfer the knowledge to the student.

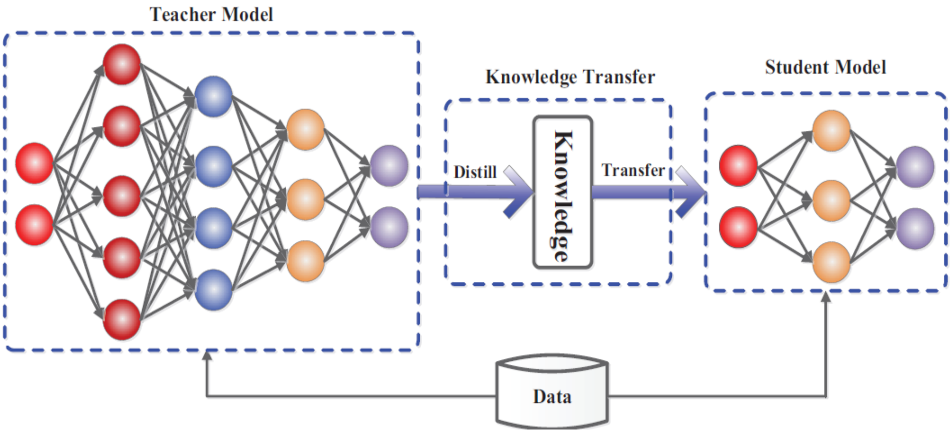


*Fig. 3: Teacher-student structure for knowledge distillation [4]*

The student model continuously learns from the teacher model while eliminating the teacher model's unnecessary features. This refinement of knowledge makes the student model more accurate and performs better than the teacher model.

# CHAPTER 3: LITERATURE REVIEW

## 3.0 Chapter Summary

This chapter summarises the related work and the various techniques currently being applied/used for hate speech detection. A brief discussion of the different machine learning techniques has also been presented.

## 3.1 Related Work

Ruwandika and Weerasinghe [17] provide data obtained from the Colombo Telegraph consisting of other comments written by users on articles related to Sri Lankan. The selected papers were published between April and May of 2017, having more than 25 comments in them. The dataset prepared compromised of 1500 comments. From the comments collected, 1000 were manually labelled either as hate or no hate. After performing the literature review for hate speech, the datasets were manually annotated based on the definition got from the study of hate speech, in which 421 were listed as comments with hate and 579 were comments without hate.

Ruwandika and Weerasinghe [17] made a comparison between supervised machine learning techniques (NBC, D.T., L.R., and SVM) with K-means Clustering [17] in hate speech detection. The researchers performed an extraction of the features by use of sci-kit learn. BoW features extraction was done by use of a count vectorizer. [17]

They considered four different features, including Bag-of-words, Tf-IDF & Bag-of-Features. The five different classifiers were trained using the "four feature types" and evaluated using the testing dataset. The researchers also aimed at determining the effect caused by the size of a dataset on the accuracy of a model. From the original set of 1000 comments, only 500 of them were used for both the testing and training of the models [17].

The evaluation metrics chosen were recall, accuracy, F-score, and precision. The classifier is a binary classifier because, from the data set, a comment either contains hate or not. All values which were checked against the evaluation metric all relied on the idea of negatives and positives. A negative is a comment without hate, while a positive is a comment with hate. Since one of the models was unsupervised, a separate evaluation was done for it.

Rohmawati, Sari Widya, and Cahyani [18] used a dataset comprising of different comments taken from various media pages like Twitter, YouTube, Instagram, and Facebook. The words were taken from other famous accounts of people living in Indonesia, hater account or fans page, which later was split into two categories labelled as either safe or not safe. The safe class of 700 was a class that didn't contain either offensive language or hate speech. The Not safe type also of 700 included either of the two.

Rohmawati, Sari Widya, and Cahyani [18] also compared different supervised machine learning techniques NBC, D.T., L.R., and SVM with KNN [18]. They performed classification by testing different classifiers and determining which one was good at hate speech detection. This classification was done using the Sckit-learn supervised learning module. In measuring the performance of classifiers, an accuracy score (ratio of correct predictions to total predictions) was used to determine how effective a classifier can be. During classification, the errors are evaluated using confusion matrices since they clearly show the misclassifications that are made [18].

Rodríguez, Argueta, and Yi-Ling [19] dataset consisting of 2400+ tweets were retrieved from Twitter. "The tweets depicted various racist, violent, slurs, offensive language and abusive remarks towards an individual or a group of individuals."

According to Yash Saini, Bachchas, Yogesh, and Sanjay [19], different unsupervised techniques were also used to examine Abusive text and/or hate speech using SOM and K-means clustering after the application of Latent Dirichlet Allocation [21]. They achieved this by creating ten topics from the corpus of the dataset while assigning a value to each dominating topic. [19]

The experiment described in [19] showed a clear indication that using the ten topics yields good results.

The researchers in [19] first used the LDA then after applied the SOM & K-means as follows;

- They decided to train SOM using values 1 and 0.5 for sigma & learning rate, respectively, for over 100 iterations.
- K-means clustering is performed with the value of K as ten equal to the number of topics.

Rodríguez, Argueta, and Yi-Ling [19] detect hate speech on social media, specifically Facebook pages that could potentially promote hate speech which is identified using graph analysis, and after sentiment and emotion analysis is applied to them in which the most negative posts together with the comments were obtained [19]. The most discussed topics were determined, and K-means clustering was applied to them. Researchers used different Sentiment and emotion parameters were used by researchers in [19]. They also used clustering parameters to achieve better clusters, as mentioned in [19].

Amrutha and Bindu [16] use and compare the performance of different deep neural networks in hate speech detection, i.e. Gated Recurrent Unit (GRU) good at "capturing sequence orders", Convolution Neural Network (CNN) ideal for "feature extraction" and Universal Language Model Fine-tuning model (ULMFiT) that's based on "transfer leaning technique" [16]. Once the data collection is done from Twitter, the next step is preprocessing, followed by a layer where word embedding is performed on the tweets to obtain a "numerical vector" [16]. This is followed by applying the GRU or the three layers of CNN (convolution, pooling, and fully connected layer) model shown in [16]. The ULMFiT model generates a "pre-trained model" obtained using an extensive data set to train it. The aim is to fine-tune the dataset when it is later fed into an RNN model, i.e. the three-layered LSTM after which its soft-max is calculated [16].

Saksesi, Nasrun, and Setianingsih [1] performed hate speech detection using recurrent neural networks. The data is collected from several Twitter accounts that potentially spread hates speech. [1] A total of 1235 tweets verified by Balai Bahasa & collected using an official Twitter API were labelled as 1 for hate speech tweet & 0 for not hate speech where 652 had hate speech & 583 didn't have.

They perform preprocessing of the data in different stages, i.e., Case folding, Tokenizing, cleaning, and then stemming. After preprocessing, the sentences are then converted to vector values using the word2vec module of NLP. [1] They also considered three training parameters; Epoch, Batch size, and Learning Rate. [1]

After determining the sentences' vector values, the next stage is a calculation in RNN using LSTM. The softmax value is then calculated based on the LSTM value that was calculated. In the last layer, the dropout, the class is classified as either 1 or 0 for hate speech or not hates speech. [1]

The softmax is calculated using the formula below.

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_j}} \qquad (1)$$

# CHAPTER 4: METHODOLOGY

## 4.0 Chapter Summary

This chapter discusses the approach and techniques used to undertake the project work. In our proposed method, LSTM enhancement is done using knowledge distillation. From [4] knowledge distillation system has different components such as the type or kind of knowledge to be used, the Training/distillation scheme used. We propose to use Response-based knowledge with an offline distillation architecture/scheme.

## 4.1 Response-based knowledge

Knowledge is an essential part of the distillation system since it provides or specifies what the student model has to learn. Response-based knowledge is the response of the final output from the teacher model. The sole aim is to try and imitate the final prediction made by the teacher model.

The commonly used response-based knowledge for classification is the soft targets. These soft targets are the probabilities that the inputs belong to a particular class. [4]

After determining the LSTM value in [1], these values are then used to calculate softmax. In knowledge distillation, the softmax function, as mentioned in [3], slightly changes from the one mentioned above.

A new parameter called the temperature denoted as T is introduced to produce a softer probability distribution. The new softmax function for distillation uses the Temperature T to make the logits value smaller by having a high T value, as shown below.

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_i/T)} \qquad (2)$$

Where T is the temperature value that is adjusted until a softer distribution is achieved. The value of T is usually set to 1. [3]

## 4.2 Offline Distillation

We proposed using offline distillation in the proposed method, where we first create the teacher network and train it using the training dataset. The pre-trained network is then used to guide the student model or network. The teacher model extracts the knowledge from the immediate feature or in the form of logits, and this knowledge is used to guide the distillation process. [4]

Offline distillation often involves the one-way transfer of knowledge, i.e., from the teacher to the student. There is a knowledge gap because the student model is highly dependent on the teacher model.

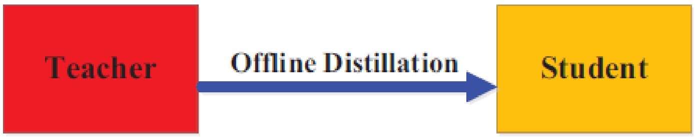


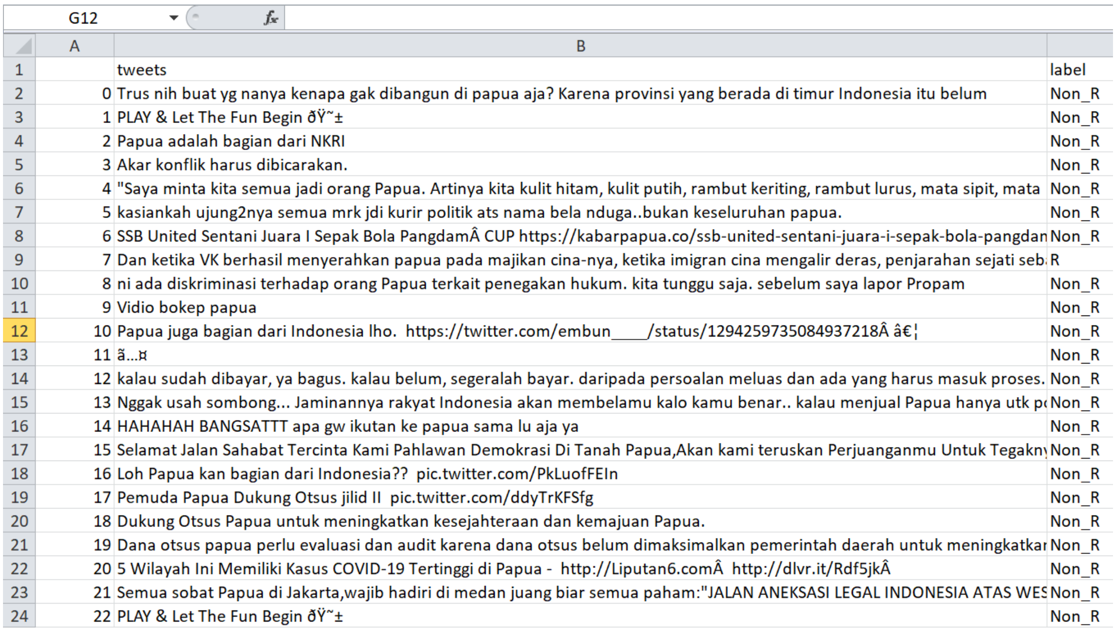
*Fig. 4: Offline Teacher-Student distillation. [4]*

## 4.3 Dataset

### 4.3.1 Twitter Data

The dataset used is a publically available dataset used to detect hate speech in Indonesia. The data was collected from different Indonesian Twitter accounts possibly promoting or demonstrating hate speech. [22]

A total of 686 tweets were collected, of which were manually labelled. One hundred seventy-five of them were marked to have hate speech for racism denoted as "R," while the remaining balance of 511 was labelled as not having hate speech marked as "Non_R."



| | A | B | |
|---|---|---|---|
| 1 | | tweets | label |
| 2 | 0 | Trus nih buat yg nanya kenapa gak dibangun di papua aja? Karena provinsi yang berada di timur Indonesia itu belum | Non_R |
| 3 | 1 | PLAY & Let The Fun Begin ðŸ˜± | Non_R |
| 4 | 2 | Papua adalah bagian dari NKRI | Non_R |
| 5 | 3 | Akar konflik harus dibicarakan. | Non_R |
| 6 | 4 | "Saya minta kita semua jadi orang Papua. Artinya kita kulit hitam, kulit putih, rambut keriting, rambut lurus, mata sipit, mata | Non_R |
| 7 | 5 | kasiankah ujung2nya semua mrk jdi kurir politik ats nama bela nduga..bukan keseluruhan papua. | Non_R |
| 8 | 6 | SSB United Sentani Juara I Sepak Bola PangdamÂ CUP https://kabarpapua.co/ssb-united-sentani-juara-i-sepak-bola-pangdan | Non_R |
| 9 | 7 | Dan ketika VK berhasil menyerahkan papua pada majikan cina-nya, ketika imigran cina mengalir deras, penjarahan sejati seb | R |
| 10 | 8 | ni ada diskriminasi terhadap orang Papua terkait penegakan hukum. kita tunggu saja. sebelum saya lapor Propam | Non_R |
| 11 | 9 | Vidio bokep papua | Non_R |
| 12 | 10 | Papua juga bagian dari Indonesia lho.  https://twitter.com/embun____/status/1294259735084937218Â â€¦ | Non_R |
| 13 | 11 | ã…¤ | Non_R |
| 14 | 12 | kalau sudah dibayar, ya bagus. kalau belum, segeralah bayar. daripada persoalan meluas dan ada yang harus masuk proses. | Non_R |
| 15 | 13 | Nggak usah sombong... Jaminannya rakyat Indonesia akan membelamu kalo kamu benar.. kalau menjual Papua hanya utk p | Non_R |
| 16 | 14 | HAHAHAH BANGSATTT apa gw ikutan ke papua sama lu aja ya | Non_R |
| 17 | 15 | Selamat Jalan Sahabat Tercinta Kami Pahlawan Demokrasi Di Tanah Papua,Akan kami teruskan Perjuanganmu Untuk Tegakn | Non_R |
| 18 | 16 | Loh Papua kan bagian dari Indonesia??  pic.twitter.com/PkLuofFEIn | Non_R |
| 19 | 17 | Pemuda Papua Dukung Otsus jilid II  pic.twitter.com/ddyTrKFSfg | Non_R |
| 20 | 18 | Dukung Otsus Papua untuk meningkatkan kesejahteraan dan kemajuan Papua. | Non_R |
| 21 | 19 | Dana otsus papua perlu evaluasi dan audit karena dana otsus belum dimaksimalkan pemerintah daerah untuk meningkatka | Non_R |
| 22 | 20 | 5 Wilayah Ini Memiliki Kasus COVID-19 Tertinggi di Papua -  http://Liputan6.comÂ  http://dlvr.it/Rdf5jkÂ | Non_R |
| 23 | 21 | Semua sobat Papua di Jakarta,wajib hadiri di medan juang biar semua paham:"JALAN ANEKSASI LEGAL INDONESIA ATAS WES | Non_R |
| 24 | 22 | PLAY & Let The Fun Begin ðŸ˜± | Non_R |

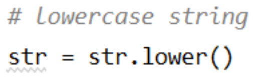
*Fig. 5: Sample of tweets collected from different Indonesia Twitter Accounts [22]*

### 4.3.2 Preprocessing

Preprocessing is the next step once the data has been retrieved from the different Twitter accounts and saved a CSV file, as shown in figure 5 above. Preprocessing involves several sub-processes or stages performed until the resulting data is ready for the classification stage. These processes include the following;

a) Changing the Case

The first sub-step in preprocessing the data is to convert the data or sentence or tweet to a lower case. Meaning all the capital letters in the tweet will be changed to lowercase to make all the text uniforms.



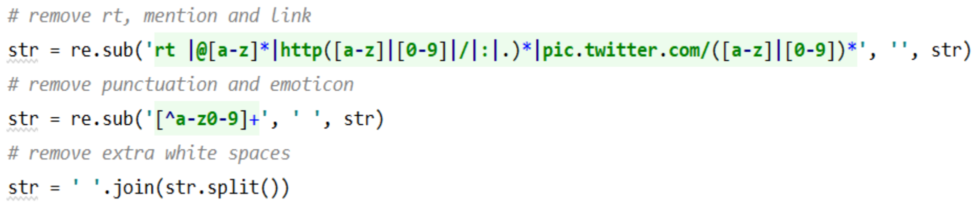
```
# lowercase string
str = str.lower()
```

*Fig. 6: Converting the text to lowercase*

b) Cleaning

The next sub-step of preprocessing is cleaning the data. After changing the data to lowercase, we have to clean it by removing unnecessary text with the sentence. Cleaning the data could be any of the following;

- Removing a link or mention from the tweet.
- Removing punctuation or emoticon from the tweet.
- Remove all the extra white spaces that might be there.



```
# remove rt, mention and link
str = re.sub('rt |@[a-z]*|http([a-z]|[0-9]|/|:|.)*|pic.twitter.com/([a-z]|[0-9])*', '', str)
# remove punctuation and emoticon
str = re.sub('[^a-z0-9]+', ' ', str)
# remove extra white spaces
str = ' '.join(str.split())
```

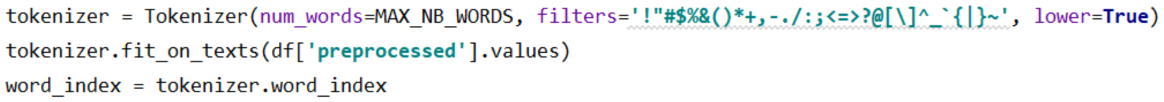
*Fig. 7: Cleaning the text, removing spaces and unnecessary characters*

c) Stemming

Stemming is a process of grouping similar words together that might be in order forms or tenses and become one word such as cold, coldness would all equal one word as cold. This process is also vital in reducing the number of indexes of a document.

d) Tokenization

Tokenizing is a process in which text is split or broken down into words or phrases, usually referred to as tokens. This process aims at exploring the words in a sentence, and it's also essential for other functions like text mining or parsing.



```
tokenizer = Tokenizer(num_words=MAX_NB_WORDS, filters='!"#$%&()*+,-./:;<=>?@[\]^_`{|}~', lower=True)
tokenizer.fit_on_texts(df['preprocessed'].values)
word_index = tokenizer.word_index
```

*Fig. 8: Tokenization of the text*

*Table I: Illustration of preprocessing*

| Process Data | **Data** |
|---|---|
| **Real data** | Apa Kabar Papua? |
| **Changing the case** | apa kabar papua? |
| **Cleaning** | apa kabar papua |
| **Stemming** | apa kabar papua |

### 4.3.3 Training Parameters

The algorithm utilizes several parameters to control specific properties during the model's training process in machine learning. In most cases, it is clear that the value of this parameter can affect the results of the evaluation at a later process. Also, variations of the value can produce different results. In the previous studies, such as in [1], the researchers used their parameters during the training process, i.e. epoch, batch size, and learning rate. The parameters each vary the results at different values. In this research, we maintain a constant value for each parameter and then introduce the fourth parameter, temperature, whose values are varied.

a) Batch Size
Since the neural network cannot pass through the entire datasets at once, the batch size represents the total number of training samples within a batch, the batch size 64.

b) Epoch
Epoch is referred to when the all dataset is passed from forward to backward through the network once. In this case, all three models or networks each have their epoch number. The teacher model is trained for five full epochs; the distilled student model trained from the teacher model is trained for three full epochs. The cloned student model that is trained from scratch is also trained for three full epochs.

c) Learning Rate
The learning rate is an essential parameter during the training phase helpful in determining the value of weight correction. This rate is known to affect the accuracy of the network.

d) Temperature
The temperature is introduced as a new parameter to help in the distillation. The temperature is used in the softmax function to smoothen the probability distributions. Like the standard distillation process, a high temperature is essential to evaporate the residues that are not needed, and the same is for the neural network. A higher temperature produces a higher accuracy of the model.

### 4.4 Methodology

The process used in this paper comprises six steps described below. The process is based on and uses the concept of the Keras knowledge distillation class. [9]

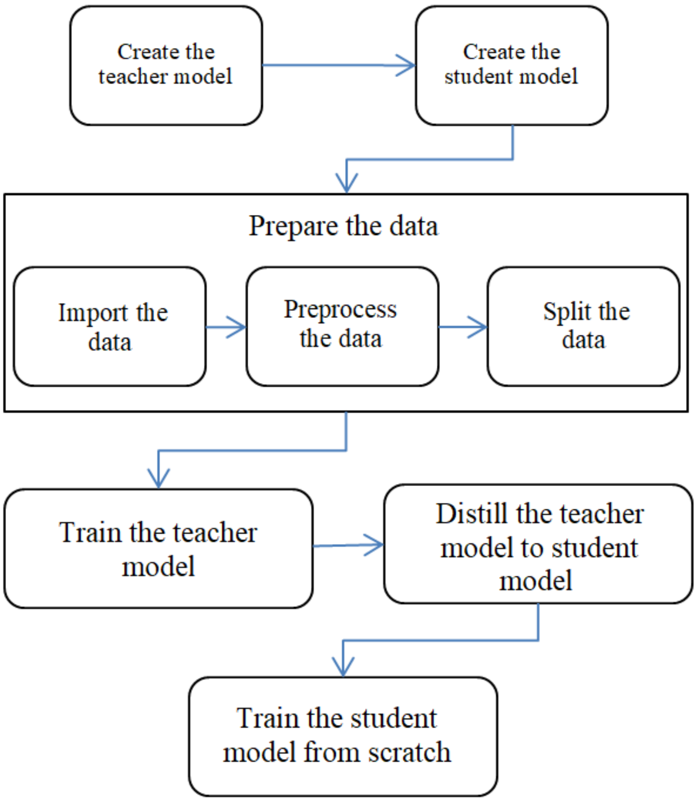


*Fig. 9: Flow chart of the proposed methodology*

### 4.4.1 Create teacher model

The initial step is to create a teacher model using the sequential function from Keras. [9] The model is an LSTM neural network.

### 4.4.2 Create a student model

We create a small student neural network using the sequential function [9], Similar to the teacher but with different values. The model is also an LSTM model as the teacher.

### 4.4.3 Preparing the dataset

The dataset used to train the teacher, and the student is as mentioned above. Both models are also evaluated based on this dataset. In this step, the data in a CSV format is imported, and preprocessing of the data also happens. The preprocessing consists of converting words to lowercase, removing the mentions and links, removing the punctuation, removing the extra spaces in the sentence, and finally, tokenization. Data analysis is done by differentiating the tweets and counting them to ensure they were loaded.

The data is then split into training data and the testing data using the split function.

### 4.4.4 Train the teacher model

In this step, we train the teacher model using the data set loaded and processed in step 3 above. We train it using the compile function and the fit function [9] to assume that the teacher model is always fixed.

Then we also go ahead and evaluate the teacher model based on the data set.

### 4.4.5 Distil teacher model to student model

After training the teacher model from step 4 above, the student model is trained by the teacher by creating an instance of the distiller in Keras. [9] It will also compile the loss during distillation, loss of the student model, optimizers, and hyperparameters. The temperature value can be adjusted accordingly to soften the logits, improve the probability distribution, and expose the relationship in classes.

Let $T_i$ be soft_teacher_predictions,

($S_i$) be soft_student_predictions.

$D_{loss}$ is the distillation loss.

$$T_i = \frac{\exp(teacher\_predictions/T)}{\sum_j \exp(teacher\_predictions/T)} \qquad (3)$$

$$S_i = \frac{\exp(student\_predictions/T)}{\sum_j \exp(student\_predictions/T)} \qquad (4)$$

$$D_{loss} = T_i - S_i \qquad (5)$$

### 4.4.6 Train another student model from scratch for comparison with the distilled model

In this step, a model similar to the student model is created and trained from scratch using the dataset without the teacher model's help. This will help in evaluating the performance that has been gain after knowledge distillation.

## 4.5 Step by Step Algorithm

**Step 1:** Let T.R. be the variable containing a dense LSTM Teacher network/model (T.R. = Teacher) initialized with 200 layers, dropout of 0.2, the dense value of 2, and activation function as 'softmax'.

**Step 2:** Let S.T. be the variable containing a small LSTM student network (S.T. = Student) initialized with 100 layers, dropout of 0.1, the dense value of 2, and activation function as 'softmax'.

**Step 3:** Import the dataset to a variable D.F.

Clean the imported data D.F. by removing spaces, punctuations, mentions using the string function. Using the train_test_split function, split data set to training & testing data (x_train, x_test, y_train, and y_test).

**Step 4:** Using the dataset D.F., train the teacher model T.R. with x_train & y_train values of the dataset for five epochs.

Evaluate the T.R. model using x_test & y_test from the D.F. dataset.

Let loss_TR be the loss during training of the teacher model, and accuracy_TR is the accuracy obtained by the teacher model.

**Step 5:** Initialize a distillation function D with inputs T.R. as a teacher, ST as a student, T (initial value as 3) as distillation temperature.

Perform distillation from T.R. to S.T. using the training values of the data set x_train & y_train for three epochs.

Evaluate the distillation using testing values x_test and y_test of the data set.

Let accuracy_ST be the accuracy of the obtained by student model after distillation, loss_ST be the student loss after distillation, and D_loss be the distillation loss.

**Step 6:** Let ST_S be the student network/model created as a clone of the student model ST and the same values.

ST_S is trained from scratch without the teacher model using x_train & y_train training values for three epochs.

Evaluate the ST_S model using the test data x_test & y_test.

Let accuracy_ST_S be the accuracy of the student model trained from scratch.

**Step 7:** Repeat step 1 - step 6 using different temperature values T in step 5. Compare the accuracy of distilled student model accuracy_ST with the accuracy of the student model trained from scratch accuracy_ST_S.

## 4.6 Calculation of Model Accuracy

At the last stage of the layer dropout, as mentioned in [1], the classification is done per the vector value where 1= 0.5 is hate speech and 0 for other values is classified as not hate speech. Using these values, we can determine the model's accuracy from the values predicted by the model versus the actual prediction values, as shown in the formula below.

$$Accuracy = \frac{Correct\ predictions}{Total\ predictions\ made} \tag{6}$$

# CHAPTER 5: RESULTS AND ANALYSIS

## 5.0 Chapter Summary

The results of the experiment have been discussed in this chapter. Some introductory theory on the various steps taken to obtain the results has also been elaborated. The chapter is concluded with a discussion of the results obtained and their implication.

## 5.1 Modelling

In this section, a summary of the various steps to train and test the models has been presented.

The libraries that were used in the experiment included pandas, tensor flow, sklearn, and Keras. These were imported and implemented using Python 3 after the relevant dependencies were installed.

The hate speech dataset that is .csv format is uploaded and read using the pandas library. The preprocessing of the uploaded dataset is done using the string functions available in Python, and after the tokenizer from the tensor flow is applied to the data.

The teacher model (extensive neural network) of LSTM is then created using the Keras library, and it's trained using the dataset that was uploaded.

The student model (smaller neural network) of LSTM is created and trained using the teacher model that was already trained. The distillation process happens at this stage.

A clone or identical model of the student network is created and trained from scratch using the dataset uploaded.

The above processes are done with different values of the T. The temperature and the accuracies of each are obtained and recorded in a tabular format.

## 5.2 Results & Analysis

Different tests were performed with various values of the temperature. The aim was to determine the highest accuracy achieved with the other values obtained during distillation and by the individual model trained from scratch, after which the values are compared. The knowledge transfer is done from the more extensive network (teacher) to the smaller student network. The teacher has trained for five full epochs to output accuracy of 76.8%, the student network trained from the teacher network for three entire epochs attained an accuracy of 82.6%. Another student model cloned and trained from scratch for three full epochs instead of the teacher network achieves an accuracy of 75.4%.

*Table II: Accuracies of different models against various temperature values*

| **Temperature** | **Accuracy** | | |
|---|---|---|---|
| | **Teacher** | **Distilled Student** | **Student From Scratch** |
| 3 | 76.8 | 78.3 | 76.8 |
| 5 | 76.8 | 82.6 | 75.4 |
| 7 | 79.7 | 79.7 | 72.5 |
| 9 | 85.5 | 79.7 | 73.9 |

During the teacher's distillation to the student, the distillation function's temperature value is varied to achieve higher desirable accuracy. In the same concept of standard distillation, heating or high temperatures cause the impurities to evaporate. Therefore, with the same idea in mind, the unnecessary feature can be removed with high-temperature values, as shown in the following.

*Table III: Precision of different models against various temperature values*

| **Temperature** | **Precision** | | |
|---|---|---|---|
| | **Teacher** | **Distilled Student** | **Student From Scratch** |
| 3 | 0.8551 | 0.7681 | 0.7391 |
| 5 | 0.8551 | 0.8261 | 0.7826 |

| | | | |
|---|---|---|---|
| 7 | 0.7826 | 0.7536 | 0.6812 |
| 9 | 0.7101 | 0.8406 | 0.7391 |

The precision from the table above is obtained by training the teacher model for seven full epochs, distilling the student model trained for five full epochs, and the student model trained from scratch is trained for three full epochs. The highest precision is obtained at a temperature of 9 by the student model trained by the teacher of 84.1%

# CHAPTER 6: CONCLUSION AND FUTURE WORK

## 6.0 Chapter Summary

In this chapter, a summary of the work done has been given in the conclusion section, and lastly, the possible areas for future work have been highlighted in the last section.

## 6.1 Conclusion

Knowledge distillation is a growing part of neural networks since it helps reduce the performance constraints experienced by them.

Knowledge distillation is "a model compression process where a smaller model is trained to match a larger pre-trained model." From the tests carried out in this project and the analysis, we can make the following conclusions.

- Applying a higher temperature value during distillation can effectively improve the logits' probability distribution and enhance performance.
- Distilling the student from the teacher model performs much well than when the student is trained independently and, in some cases, better than the teacher model.
- In the model performance, the student distilled model's accuracy, teacher model, and student model trained from scratch are 82.6%, 76.8%, and 75.4%.

## 6.2 Future Work

The high-performance measure attained by the distilled student model reaffirms a promising direction of improving neural network performance using knowledge distillation. The next direction of this work involves using another dataset that is not highly imbalanced and applying the same concept to other deep learning algorithms for hate speech detection and/or comparing the results of the different algorithms.

# REFERENCES

[1] A. S. Saksesi, M. Nasrun, and C. Setianingsih, "Analysis Text of Hate Speech Detection Using Recurrent Neural Network," 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Indonesia, 2018, pp. 242-248, DOI: 10.1109/ICCEREC.2018.8712104.

[2] T. Lynn, P. T. Endo, P. Rosati, I. Silva, G. L. Santos, and D. Ging, "A Comparison of Machine Learning Approaches for Detecting Misogynistic Speech in Urban Dictionary," 2019 International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber S.A.), Oxford, United Kingdom, 2019, pp. 1-8, doi: 10.1109/CyberSA.2019.8899669.

[3] Geoffrey Hinton, Oriol Vinyals and Jeff Dean, "Distilling the Knowledge of a Neural Network," 2014.

[4] Jianpinug Go, Baosheng Yu, Stephen J. Maybank Dacheng Tao, "Knowledge Distillation, A survey,".

[5] LeCun, Yan. Yosua Bengio. Geoffrey Hinton. 2015. Deep Learning. Jepang.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, p. 436, 2015.

[7] M. Sundermeyer, R. Schluter, and H. Ney, "Lstm neural networks for language modeling," in Thirteenth annual conference of the international speech communication association, 2012.

[8] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, w.-K. Wong, and w.c. Woo, "Convolutional Istm network: A machine learning approach for precipitation nowcasting," in Advances in neural information processing systems, 2015, pp. 802-810.

[9] Kenneth Borup, "Introduction to knowledge distillation", Keras Python Module https://keras.io/examples/vision/knowledge_distillation/, 2020

[10] Isaac Akileng, Raju Kumar and Aruna Bhat, Hate Speech Detection using Machine Learning Techniques, International Conference on Sustainable Advanced Computing (ICSAC 2021), Bangalore, India, 5th – 6th March 2021.

[11] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Eighth international AAAI conference on weblogs and social media, pp.216–225, 2014.

[12] A. H. Aliwy and E. H. A. Ameer, "Comparative Study of Five Text Classification Algorithms with their Improvements," Int. J. Appl. Eng.Res., vol.12, no.14, pp.4309–4319,2017.

[13] L. Argerich, M. J. Cano, and J. T. Zaffaroni, "Hash2Vec: Feature Hashing for Word Embeddings," arXiv, 2016.

[14] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," arXiv Prepr. arXiv1703.04009, 2017.

[15] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for

Sentiment Analysis of Social Media Text," Eighth international AAAI conference on weblogs and social media, pp.216–225, 2014.

[16] B. R. Amrutha, K. R. Bindu, "Detecting Hate Speech In Tweets Using Different Deep Neural Network Architectures," Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS), 2019.

[17] N.D.T. Ruwandika and A.R. Weerasinghe, "Identification of Hate Speech in Social Media," IEEE International Conference on Advances in ICT for Emerging Regions, 2018.

[18] Umu Amanah Nur Rohmawati, Sari Widya Sihwi, and Denis Eka Cahyani, "An Interface for Indonesian Hate Speech Detection Using Machine Learning," IEEE International Conference on Research of Information Technology and Intelligent Systems, 2018.

[19] Axel Rodríguez, Carlos Argueta and Yi-Ling Chen, "Automatic Detection of Hate Speech on Facebook Using Sentiment and Emotion Analysis" IEEE, 2019.

[20] Yash Saini, Vishal Bachchas, Yogesh Kumar, and Sanjay Kumar, "Abusive Text Examination Using Latent Dirichlet Allocation, Self Organizing Maps and K Means Clustering," IEEE International Conference on Intelligent Computing and Control Systems, 2020.

[21] D. M. Blei, M. I. Jordan, and A. Y. Ng, Latent Dirichlet Allocation, Journal of machine learning research, 2003.

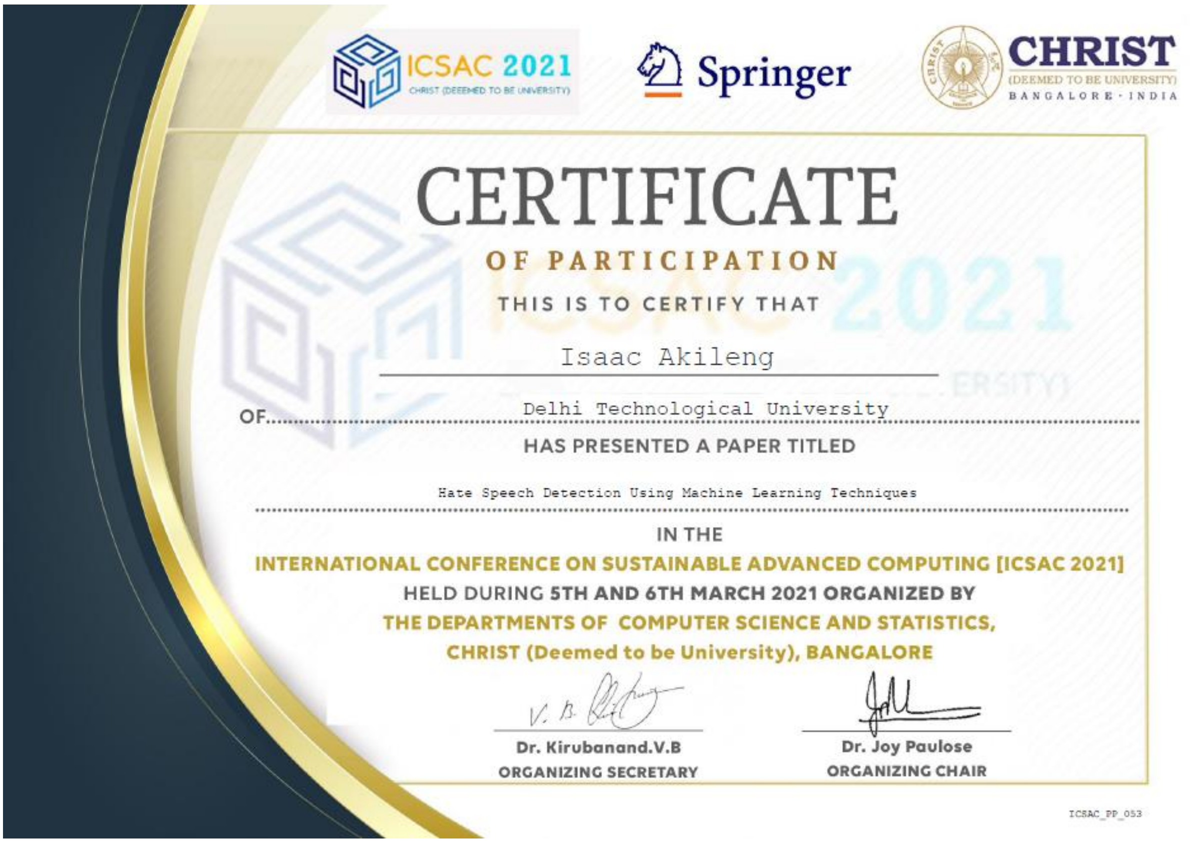
[22] A. Marpaung, "Racism Detection System", https://github.com/angelamarpaung99/racism-detection, 2020

# LIST OF PUBLICATIONS

1. Isaac Akileng, Raju Kumar and Aruna Bhat, *Hate Speech Detection using Machine Learning Techniques*, International Conference on Sustainable Advanced Computing (ICSAC 2021), Bangalore, India, 5th – 6th March 2021

2. Isaac Akileng and Aruna Bhat, *A Conceptual Enhancement of LSTM Using Knowledge Distillation for Hate Speech*, Fifth International Conference on Smart Trends for Computing & Communication (SmartCom 2021), Las Vegas, Nevada, United States (Virtual Mode), 15th- 16th April 2021

# CERTIFICATES OF PARTICIPATION



ICSAC 2021
CHRIST (DEEMED TO BE UNIVERSITY)

Springer

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

CERTIFICATE

OF PARTICIPATION

THIS IS TO CERTIFY THAT

Isaac Akileng

OF............ Delhi Technological University ............

HAS PRESENTED A PAPER TITLED

Hate Speech Detection Using Machine Learning Techniques

IN THE

INTERNATIONAL CONFERENCE ON SUSTAINABLE ADVANCED COMPUTING [ICSAC 2021]

HELD DURING 5TH AND 6TH MARCH 2021 ORGANIZED BY

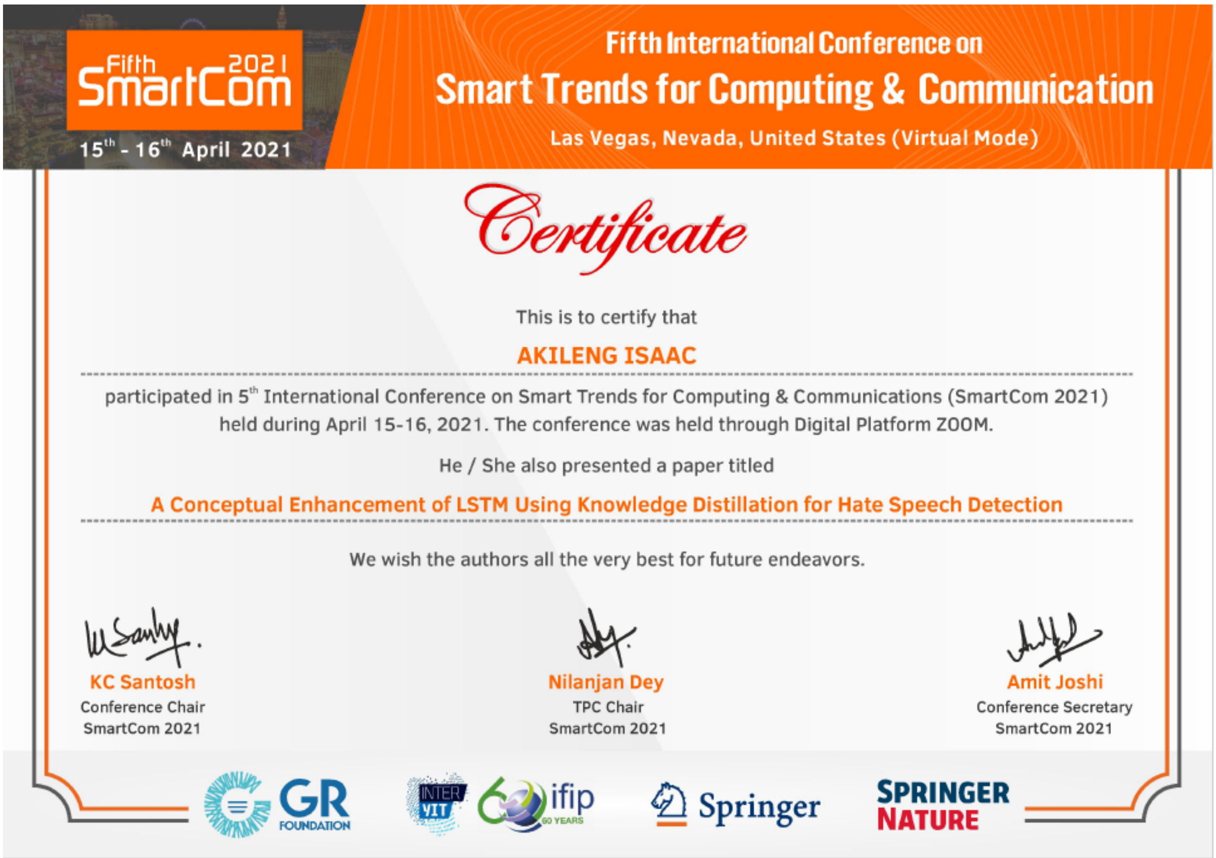
THE DEPARTMENTS OF COMPUTER SCIENCE AND STATISTICS,

CHRIST (Deemed to be University), BANGALORE

Dr. Kirubanand.V.B
ORGANIZING SECRETARY

Dr. Joy Paulose
ORGANIZING CHAIR

ICSAC_PP_053



Fifth SmartCom 2021

15th - 16th April 2021

Fifth International Conference on

Smart Trends for Computing & Communication

Las Vegas, Nevada, United States (Virtual Mode)

# Certificate

This is to certify that

**AKILENG ISAAC**

participated in 5th International Conference on Smart Trends for Computing & Communications (SmartCom 2021) held during April 15-16, 2021. The conference was held through Digital Platform ZOOM.

He / She also presented a paper titled

**A Conceptual Enhancement of LSTM Using Knowledge Distillation for Hate Speech Detection**

We wish the authors all the very best for future endeavors.

KC Santosh
Conference Chair
SmartCom 2021

Nilanjan Dey
TPC Chair
SmartCom 2021

Amit Joshi
Conference Secretary
SmartCom 2021

GR FOUNDATION

INTER VIT

ifip 60 YEARS

Springer

SPRINGER NATURE