ICCoSITE

*The 2023 International Conference on **Computer Science, Information Technology, and Engineering***

Ref. : ICCoSITE/LoA/I/2023/69
Date : January 5, 2023

LETTER OF ACCEPTANCE

2023 International Conference on Computer Science, Information Technology, & Engineering

Virtual Conference (Online) 16-17 February 2023

Paper	:	1570874935 **Discord Bot Design for Hate Speech Sensor Using Convolutional Neural Networks (CNN) Method**
Authors	:	Nicholas Hadi (Tarumanagara University, Indonesia); Viny Christanti Mawardi (Taman Semanan Indah A2' / 12A, Indonesia); Janson Hendryli (Universitas Tarumanagara, Indonesia)

Dear Author(s),

We are pleased to inform you that your paper has been **accepted to be presented** in the 2023 International Conference on Computer Science, Information Technology, & Engineering (ICCoSITE) which organized by **Bina Insani University** with our partner **APTIKOM, Universitas Indraprasta PGRI**, **Universitas Nasional, Institut Teknologi Garut,** and **STMIK ROSMA**.

The ICCoSITE will be conducted as a virtual conference on 16-17 February 2023. The accepted and presented paper has been strictly undergone the peer-reviewed process and will be submitted for uploading to the IEEEXplore Digital Library and it will be normally indexed in SCOPUS database.

This the detail of catalog number IEEE Xplore Library

1) XPLORE COMPLIANT CFP23DI9-ART ISBN: 979-8-3503-2095-4
2) USB CFP23DI9-USB ISBN: 979-8-3503-2094-7

Please understand perfectly that the publication of papers in the IEEEXplore Digital Library will take AT LEAST 1 – 4 months from the conference date. By registering to this conference, authors and co-authors will be deemed to agree, understood, and accept all IEEE Official Conference Terms and Conditions at https://www.ieee.org/web/conferences/organizers/pubs/conference_publications.html

Congratulations on the acceptance of your paper and thank you for your interest. We look forward to seeing you at the conference soon.

General Chair,

BINA INSANI
INTERNATIONAL CONFERENCE

Ahmad Chusyairi, S.Kom., M.Kom.

ICCoSITE
The 2023 International Conference on **Computer Science, Information Technology, and Engineering**

REGISTRATION PROCEDURE

All papers that have been received from the review process, please proceed with the payment registration process.

INTERNATIONAL PARTICIPANTS
Please make payments using a credit card at the EDAS system https://edas.info/r30091
Registration payments for international participants are made via EDAS using a credit card at an exchange rate of US dollars. Confirmation is automatic because it uses the EDAS system. If your payment is successful, the status will change to PAID.

DOMESTIC PARTICIPANTS
For Indonesian domestic participants, please make payment via bank transfer. To make it easier to check, please add the last 3 digits of your id paper to the registration fee. Confirmation is manual, meaning you must inform and send proof of transfer at the link provided.
For details, please contact our representative via Whatsapp +62 889-9080-8120

Participant Category	Overseas Participant	Domestic Participant
Student (IEEE Member)	US$ 175	IDR 1,750,000
Student (Regular/Non IEEE Member)	US$ 200	IDR 2,000,000
Profesional (IEEE Member)	US$ 250	IDR 2,500,000
Profesional (Regular/Non IEEE Member)	US$ 275	IDR 2,750,000

Registration Payment for Domestic Participants can be made by bank transfer to:

Name of Bank : Bank Syariah Indonesia (BSI)
PT Beneficiary Name : UNIVERSITAS BINA INSANI
Account number : 882-299-777-6

To make it easier to check, please add the last 3 digits of your id paper to the registration fee. Confirmation is manual, meaning you must inform and send proof of transfer at the link provided.
Click Here To Confirm Registration Payment

After completing the payment, participants can continue the next process, namely:

1. Fill in the e-Copyright form on the EDAS system
2. Uploading the final manuscript file. Convert by PDF Express.
   Before uploading the final version (camera ready) of your paper we kindly ask you to verify if your PDF is compatible with IEEE Xplore. IEEE offers a service for checking the PDF compatibility:
   1. Please go to https://ieee-pdf-express.org/
   2. Enter the following conference ID: **57641X**
   3. Log into the PDF Express Website. If you do not have an account please create one.
   4. Follow the steps to complete the PDF verification process.
3. Upload a presentation slide file

For any questions, feel free to contact our representatives.Best Regards.

Bina Insani University
Jl. Raya Siliwangi, Sepanjang Jaya, Rawalumbu, Bekasi, Jawa Barat, Indonesia
https://biic.binainsani.ac.id/ICCoSITE.html

Discord Bot Design for Hate Speech Sensor Using Convolutional Neural Networks (CNN) Method

Nicholas Hadi
Faculty of Information Technology
Tarumanagara University
Jakarta, Indonesia
nicholas.535190048@stu.untar.ac.id

Viny Christanti Mawardi
Faculty of Information Technology
Tarumanagara University
Jakarta, Indonesia
viny@untar.ac.id

Janson Hendryli
Faculty of Information Technology
Tarumanagara University
Jakarta, Indonesia
jansonh@fti.untar.ac.id

Abstract*—Discord is growing in popularity, makes hard for an admin of Discord server maintaining their member in their everyday chat activity in their server. This no longer an issue if there is Discord bot that can detect hate speech feature in text message that member send and automatically censor them. The classifier for this experiment is using Convolutional Neural Network (CNN) method. The dataset for training and validation model are containing total 6 category of hates speech, abusive language, religion, race, gender, physical, and non-hate speech. The Discord bot program only can classify a message in Indonesian language. The dataset that been used for the training and validation model is obtained from Kaggle and for additional data is been scraped from Discord server messages. In this experiment we found that the CNN method can be implement to Discord bot that can be used to censor hate speech messages in real time.

Keywords—convolutional neural network, deep learning, discord, hate speech

I. INTRODUCTION

In today's technological era, many Instant Messaging (IM) and Voice over IP (VoIP) applications are advanced and allow humans to socialize more efficient.

Discord is an application that allow users can communicate via voice, video, and text. Besides that, this application can offer complete features with an easy-to-access user interface. The main feature of Discord is that user can make a server for free where users can invite their friend to join and make communities for people to gather and communicate to each other. In 2020, Discord is recorded to have increased number of users increased by 20% in total 250 million users [1]. This lead by of topic where Discord's server was used for a place where some communities can gather. Because of that, many of Discord's server members can send some message that can be abusive to another individual, and making Discord a dangerous platform for people to communicate.

Many Discord's server admins provide a solution to this problem by hiring the services of an individual to become the server moderator. The main task of a server moderator is to maintain the behavior of each member of the server in order to comply with the rules in order to obtain a safe and comfortable community environment for all server members. But if it's just a moderator himself, there are opportunities where negative content can be missed and it won't be effective because a moderator can't monitor server activity 24 hours a day. Therefore, this problem can be solved by using a Discord Bot that can censor hate speech message in real time using Natural Language Processing (NLP) technology.

For this Discord bot to be able to censor hate speech, this application requires a classification model to detect whether the utterance contains the meaning of hate speech or not. The classification model was designed using a deep learning algorithm method called Convolutional Neural Network (CNN). The CNN method is a deep learning method that is usually used as a Computer Vision process. CNN has been proven to be able to train short text sentence classification models and get results that have great accuracy and can be compared with the sequential LSTM training method [2]. This classifier model will assign each message to six hate speech i.e., Abusive Language, Religion, Race, Gender, Physical, and Non-Hate Speech. This classifier also can only classify Indonesian language messages.

II. EXISTING SYSTEM

In 2019, Zakhayu Rian, Viny Christanti, Janson Hendryli using CNN method for image retrieval based on its content. Using total 579184 training images, 13 masterclass, 5089 subclasses, the model training using VGG16 model with 0,0001 learning rate, 7000 epochs, has succeeded in classifying the images in the validation dataset, with accuracy (F1-Score) of 73% and an average of precision in retrieval is 89.6% [3].

Research from Kevin Antarkisa, Y. Sigit Purnomo WP, and Dra. Ernawati that tries analysis of Naïve Bayes, SVM, and Logistic Regression methods. The classified data is a collection of Twitter application tweet data collected as many as 20921 data which are not hate speech and 384 data which are hate speech in Indonesian. The method with the greatest accuracy is the Logistic Regression algorithm model with the N-gram Char Level method which obtains an accuracy of 98% with an F-score of 97%, while the lowest accuracy result is the Bernoulli Naïve Bayes method with the N-gram Char Level feature extraction method [4].

Research from Awet Fesseha, Shengwu Xiong, Eshete Derb Emiru, Moussa Diallo, and Abdelghani Dahou was made to try to classify magazines in Tigrinya language which is a Low Resource language using the CNN method. This study also compared the results of CNN classification accuracy with other classification methods such as Naïve Bayes, Random Forest, SVM, Decision Tree. This study also builds a CNN model with a word embedding process using the Word2Vec and Fast Text methods. With a dataset of 30,000 articles, the best classification model results were obtained using the CNN Model Word2Vec continuous bag-of-words (CBOW) method with an accuracy of 93.41% [5].

Mark Hughes, Irene LI, Spyros Kotoulas, and Toyotaro Suzumura. This research uses the CNN classification model

to classify 4000 medical sentence data with 26 medical categories by comparing other methods, namely LogR with Doc2Vec, ZeroMean with Word2Vec, ElimMean with Word2Vec, BOW with LogR, and CNN with Word2Vec. The results of this study the method that produces the best model is the CNN + Word2vec method where this method produces an accuracy value of 68% [6].

Discord bot by Rizki Septiansyah, Sabriansyah Rizqika Akbar, and Rizal Maulana. In this research, a bot application was designed that runs on Raspberry Pi and can be used to read sensors on Raspberry Pi. The bot that will be designed is based on the Discord application. In its implementation, this bot has carried out functional tests and every case tested can work properly and is declared successful. In the accuracy test, the smallest average percentage error was obtained using the SRF05 sensor of 3.387%. and processing time testing using the LM35 sensor has the fastest average with an execution time of 545.54 ms learn commands, 8.007 ms cmd commands and 1.07 ms forget commands [7].

III. DISCORD

The design of the hate speech sensor system will be based on an application called Discord. Discord is a new application that provides communication features with a design that is simple, practical, easy to use, attractive and accessible from various gadgets. Discord uses Voice over Internet Protocol (VoIP) technology. VoIP is a technology that is usually used for sending voice, video, and data file-type communication media via the internet by converting voice and video data into a digital code [8]. One example of an application that used VoIP technology before discord was the Skype application which was founded in 2003. This Skype application is able to provide voice and video call services for free and has better sound and video quality than a telephone connection. Discord is similar to the Skype application, but in general the Discord application is made for use when playing games. The Discord application can be accessed for free through the official Discord website for Windows and Linux users. For mobile gadget users, Discord can be accessed from the Google Play Store for Android and the App Store for MacOS.

Users with Discord account can log in to Discord application and immediately be taken to the main page of the Discord application which can be seen in Fig. 1 where users can use the features available in the Discord application, such as creating a server for a group of friends or community, customizing profiles, viewing friends' lists and activities. who are active on discord, and send private messages to friends. The main feature of Discord is that users can create a chat group called a server. To create a server, users only need to press the plus-shaped icon on the left side of the main page. Discord also provides various templates that users can use to customize the server theme they want to create.

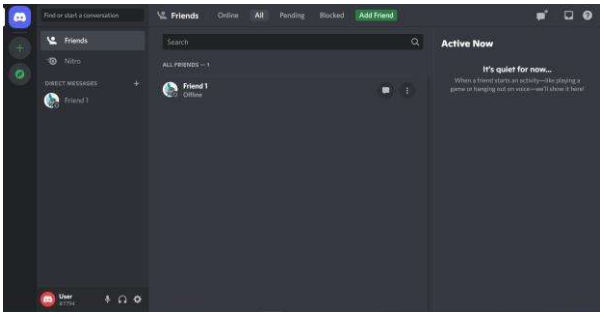

Fig. 1. Discord Main Page

On the Discord server, the user who creates the server will get a role, namely the server admin who has the power to add and remove members from the server, delete server member messages, add channels on the server for communication facilities, and so on. As can be seen in the display of the Discord server page in Fig. 2, there are 2 types of channels, namely text channels and voice channels.

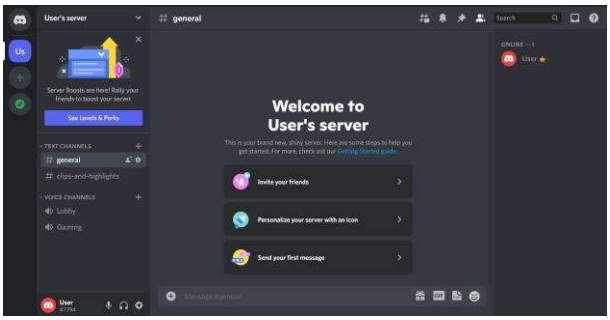

Fig. 2 Discord Server Page

The text channel is a place where all server members can communicate by sending messages in the form of messages, while the voice channel is a place where all members can enter and communicate audio and video with other members in the voice channel.

IV. DISCORD API AND BOT ARCHITECTURE

Designing a discord application-based bot for hate speech sensors requires an Application Programming Interface (API) that has been provided by Discord called the Discord API. Basically, Discord API uses 2 main layers, namely REST API which works between HTTP requests and WebSocket connections which are used for real time actions. In order for the Discord API to authenticate a bot in the Discord application, the developer needs to obtain a bot token or use the OAuth2 bearer token which can be obtained from the OAuth2 API.

Developers can access the Discord API using the Python programming language with library named discord.py and Javascript with discord.js. The process of designing and deploying a discord bot application follows sequential stages starting from choosing a tool to write code to choosing a platform between self-hosting or using other alternative hosting options [9].

The next step, the bot application needs to be created and registered along with the name through the Discord Developer Portal site. After the bot has been successfully registered, a bot token will be given and it needs to be kept secret only for the bot developer. The token is used to pass

to the bot application backend for the API to be able to authorize and identify the bot to connect to.

The way the bot application works starts from the bot listening to specific user messages according to the configuration. If the message is input to a server where the bot application is located and given permission to read messages on the server, the bot application can be triggered and can identify the response that was issued based on the designed backend. If the discord bot application is to be used publicly, then this bot application needs to be hosted in the cloud. For the bot application to run non-stop, a tool can be used to monitor ping connections. The tool can also help to provide information to the developer if the system has problems running the code. The workflow process for the discord bot from user input to the application backend system can be seen in Fig. 3.

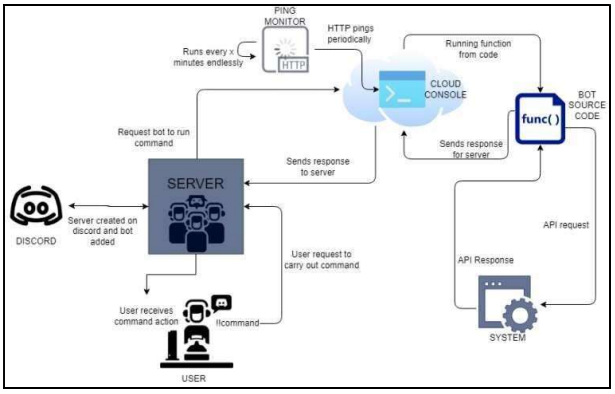

Fig. 3. Discord Bot Architecture [9]

V. CONVOLUTIONAL NEURAL NETWORK FOR TEXT CLASSIFICATION

In this Discord bot design, the process for creating a hate speech classification model will use a Deep Learning algorithm called the Convolutional Neural Network. Although CNN is generally used in Computer Vision, it is starting to be widely used for NLP processes. The CNN method is stated to be more accurate than the LTSM method, especially during the feature extraction process. 1D CNN can process text data as one-dimensional images and 1D CNN is used to capture hidden relationships between word relationships, in contrast to the LTSM method, which processes each word in a sequential pattern. The CNN architecture used in the text classification process can be seen in Fig 4. which consists of several layers, namely the convolution layer, pooling layer, and fully connected layer [10].

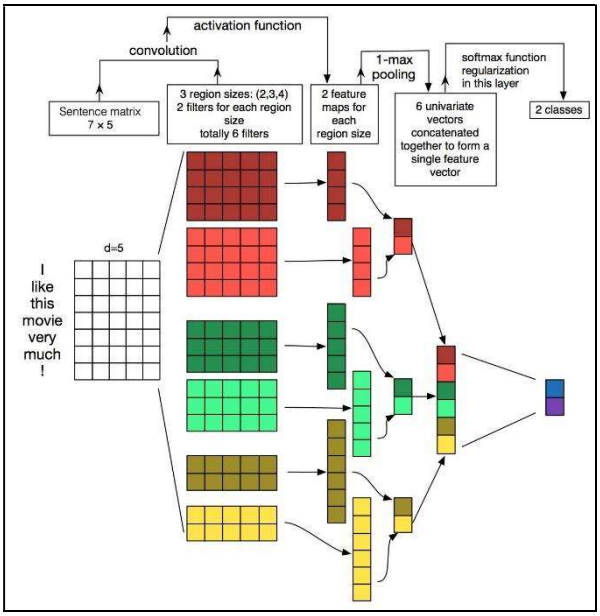

Fig. 4. CNN Architecture for Text Classification [10]

The first layer is the convolutional layer, this layer contains a series of filters with a fixed size to be used to convolution the data. The output issued at this layer is a matrix called feature maps. For the calculation of the output of this layer can be done in equation (1) [10] .

$$FM_{a,b} = bias + \sum_c^c \sum_D^D Z_{c,d} * X_{a+c-1,b+d-1} \quad (1)$$

The process of the output for the convolutional layer process in Fig. 5 can be seen that the input layer matrix will be calculated using a 3x3 filter matrix which will form a feature map whose size is determined from the size of the filter entered. Before entering the pooling layer, feature maps will be processed by the ReLU activation function. the ReLU activation function works to change the contents of neurons on feature maps that have a negative value to a value of 0 and can be seen in equation (2)[3].

$$ReLu(x) = \max(0, x) \quad (2)$$

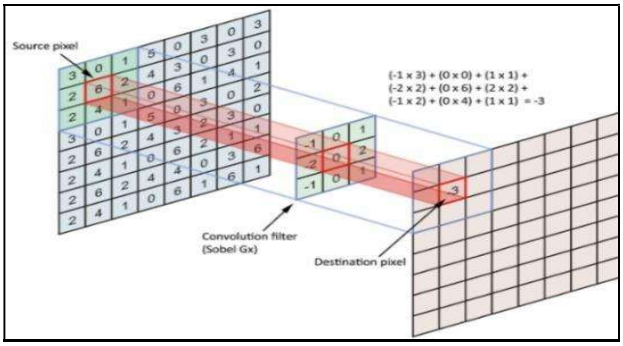

Fig. 5. Convolutional layer process [10]

Pooling Layer is works to reduce the dimensions of the feature maps that will be used in the next layer, an example of the method used is max pooling where on feature maps the largest value of the matrix will be searched and then it will be taken as an output with smaller matrix dimensions for the result. For an illustration of the pooling layer illustration, see Fig.6.

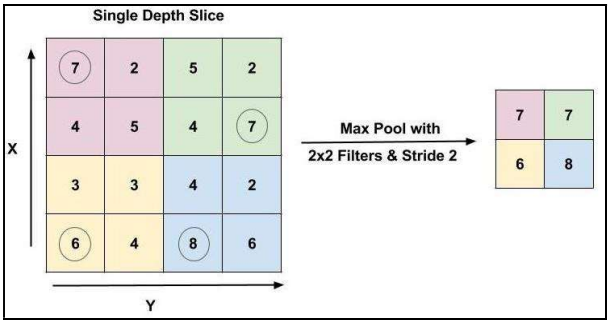

Fig. 6 Max Pooling Layer [10]

The last layer is the fully-connected layer which works to transform the matrix so that it can be classified based on the many existing classes. In the fully connected layer, there is a sigmoid activation function which aims to calculate the probability value of each label so that the class that has the highest probability value can be found as input. After the process is complete, the output results are calculated by the loss value using the binary cross-entropy method. The equation of the loss function can be seen in equation (3) [10].

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \times \log(p(y_i) + (1-y_i) \times \log(1-p(y_i)) \quad (3)$$

VI. TRAINING CNN MODEL

For training the CNN classifier model, dataset hate speech in Indonesian language was needed. Training and validation data was collected from a Kaggle site called "Indonesian Abusive and Hate Speech Twitter Text". The dataset contains a collection of text data scraped from social media Twitter. The amount of data to be retrieved is 13,170 data[11].

For additional data, a large number of chat data will be used. So, for data collection, we will use the method of scraping chat data from the Discord application. Before starting the scraping process, it is necessary to find an active and large-member Indonesian Discord server community. Found a discord server named HYPEABIS. This HYPEABIS Discord server is an Indonesian community server with 5000-6000 active members. Because this server has many and active members. This server also has moderators and strict rules in the chat, therefore the chat on the text channel server is suitable to be used as a server for scraping non hate speech label data and the total data that will be used is 18,986 data. Proportion dataset for each category label can be seen in Table I.

TABLE I. PROPORTION TRAINING AND VALIDATION DATA

Data	Number of Data
Abusive	3937
Religion	793
Race	566
Physical	322
Gender	306
Non HS	13182

The proportion of dataset text are split into 2 datasets. 80% of dataset will be for training process, and 20% of the dataset will be for validation process.

Before model training process, the collected data needs to be pre-processed. Data pre-processing performs several processes, namely filtering, case folding, stemming, and tokenizing. After that, the data that has been tokenized will be processed by a process called text to sequence, where this process changes all the word tokens in the data into integer sequences. Then the data will be pad sequenced where this process will turn the integer into an array list of the same size according to the maximum parameter length of the word sequence set in this model of 100 words.

For the word embedding process, the model pre-trained word2vec model with a size of 50 dimensions from the Indonesian Wikipedia data. With this the system can obtain the weights to be used when embedding the CNN layer. The next process is to train the data input into a 1D CNN or Conv1D parameter model.

The 1D CNN training model has an embedding layer of 50 windows with a size of 100 steps per window. After that, the model performs 3 convolution layers and 3 max pooling and 2 fully connected layers with size 6 based on the category output. The activation function that used in this classify model is the sigmoid activation function.

By splitting training data by 80% and test data by 20%, with 50 epoch and also using a learning rate parameter of 0.0001, Adam optimizer, and a batch size of 128. Total amount of time of training is 2111ms. The accuracy, loss can be seen in Table II.

TABLE II. ACCURACY AND LOSS RESULT TRAINING MODEL

Epoch	Loss	Accuracy	Val Loss	Val Acc
10	0,112	86,77%	0,139	90,02%
20	0,0694	91,31%	0,215	85,59%
30	0,0475	93,45%	0,1862	88,97
40	0,0431	93,68%	0,2398	87,51%
50	0,0397	94,12%	0,205	89,07%

From the table, we can conclude that the accuracy using training data keep increasing but the validation accuracy using validation data is inconsistent around 90-85%. Same for loss result, the training loss is keep decreasing after training epoch while the loss epoch is inconsistent around 0,139 – 0,24. For the classification report result the training model can be seen in Table III.

TABLE III. CLASSIFICATION REPORT MODEL

Label	Precision	Recall	F1-Score
Abusive	0,85	0,88	0,86
Religion	0,89	0,96	0,93
Race	0,97	0,89	0,92
Physical	0,77	0,82	0,80
Gender	0,96	0,85	0,87
Non HS	0,96	0,95	0,95

VII. IMPLEMENTING AND TESTING DISCORD BOT

After the model is trained, the model and its weights will be stored using tensorflow library in the form of H5 and JSON files for the weights. The data training tokenizer will also be stored in a JSON file. The saved files will be loaded for use on the system backend of the discord bot application.

For the discord bot creation process, the bot first needs to be registered and created through the Discord Developer Portal site to get a secret token that the backend code uses to call the discord bot, and an OAuth2 URL that is used to enter the bot into the server, as well as grant admin access. for the bots. For the backend code of this application, the Discord API is used which can be called via a python library called Discord py. With this API, the backend code can use the syntax to connect the backend code to the discord bot via the previously obtained token, and also the backend code can perform user actions that have admin access. With python code, saved models and tokens can be loaded and classification models can be used as bot process output. The backend process for the created bot can be seen in Figure 7.

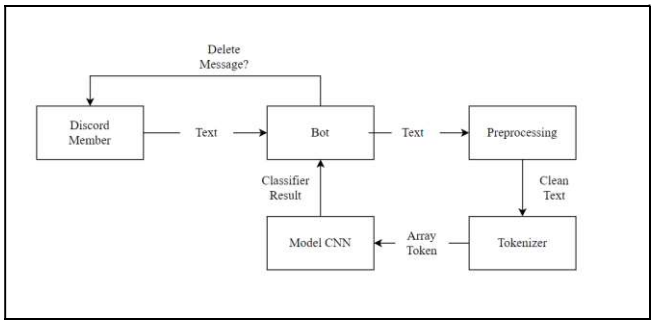

Fig. 7. Backend Process Discord Bot

A database is also created with the mySQL software to store bot classification results and also bot setting data on a bot server. With this database, bots can create additional features where users can see their hate speech info and also the bot setting feature. The database is connected to the python backend with the SQL connector library.

In Fig.8. there is the Discord bot interface when censoring hate speech message.

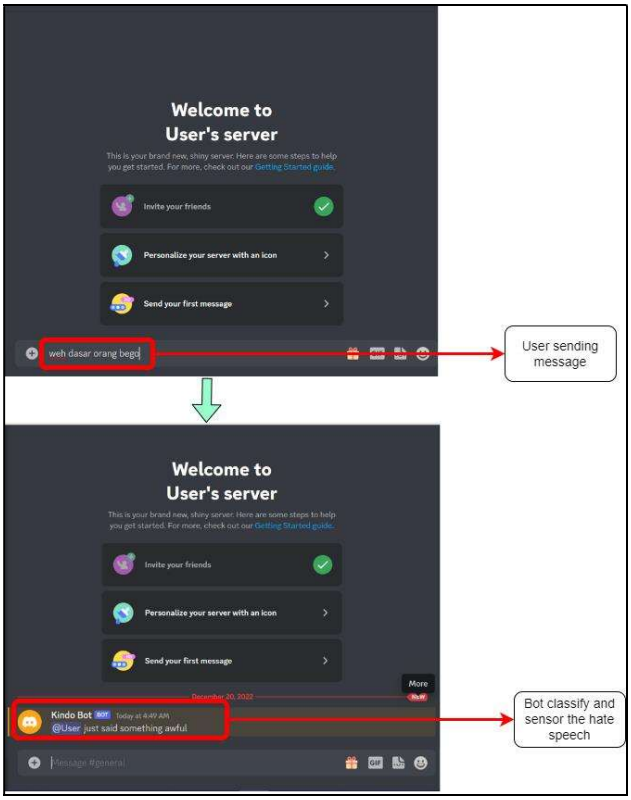

Fig. 8 Discord Bot Censor Hate Speech Interface

Discord bot testing process will be carried out by users of this program, namely members and server admins. In this testing process, this discord bot will be entered into 2 servers, namely "Vianney University" and "Graduation 2023". Members of these two servers will try to test this application by sending various messages.

Total message that been sent while the bot in the server already been deployed into the server is 279 messages, 41 abusive, 10 Religion, 28 Race, 12 Physical, 4 Gender, 184 Non-Hate Speech. The evaluation result will be carried out with confusion matrix that can be seen in Table IV.

TABLE IV. EVALUATION TESTING WITH CONFUSION MATRIX

System Output	Abusive	24	0	6	2	1	32
	Religion	0	6	0	0	0	4
	Race	0	0	14	2	0	0
	Physical	0	0	0	4	0	0
	Gender	0	1	0	1	1	0
	Non HS	17	3	8	3	2	148
		Abusive	Religion	Race	Physical	Gender	Non HS
		Target Output					

From the Discord bot test results, we will sample 10 message data from the 279 total testing messages for analysis. The sample is presented in the Table V.