ISSN No:-2456-2165

Machine Learning Algorithm Based Automated Tool for Cyberbullying Detection in Discord App

[1]RavinathaHiththatiya
Department of Computer Systems Engineering
Sri Lanka Institute of Information
Technology Malabe, Sri Lanka

[2]Yasiru Silva
Department of Computer Systems Engineering
Sri Lanka Institute of Information
Technology Malabe, Sri Lanka

[3]Dushan Fernando
Department of Computer Systems Engineering
Sri Lanka Institute of Information
Technology Malabe, Sri Lanka

[4]Dr. Lakmal Rupasinghe
Department of Computer Systems Engineering
Sri Lanka Institute of Information
Technology Malabe, Sri Lanka

[5]Shehan Kodagoda
Department of Computer Systems Engineering Sri Lanka Institute of Information
Technology Malabe, Sri Lanka

Abstract:- While online communication can be a great tool for sharing knowledge and opinions, it can also lead to cyberbullying and hatred towards individuals, and the popular Discord platform is no exception. This research aims to create a bot that can prevent cyberbullying incidents in the Discord app. Also, by focusing on the overall security of the bot, the proposed system aims to detect and prevent anomalies and SQL injection attacks. The introduction provides an overview of cyberbullying on Discord and other social media platforms. In the next section, it gives a detailed understanding of past research conducted in the realm of cyberbullying detection on social media using natural language processing techniques and deep learning. The Methodology section focuses on the system architecture and design of text classification models, image classification models, audio classification models, and bot security. The proposed system effectively uses advanced natural language processing techniques and various machine learning classifiers to accurately detect cyberbullying messages in the domains of text, image, and audio on the Discord app.

***Keywords*:-** *Cyberbullying Detection, Machine Learning Models, Phishing Link Detection, SQL Injection Detection, Content Analysis, Speech-to-Text Conversion, Optical Character Recognition, Event-Driven Programming, Discord Server.*

I. INTRODUCTION

In the present era, where social media usage is widespread among adults and teens, bullying is not limited to physical acts only. It has expanded to the realm of cyberspace, giving rise to a new form of bullying called cyberbullying [1]. This type of bullying takes various forms, such as texts, images, videos, etc., and has become the dark side of the internet's well-connected social life. With the ability to disseminate information to millions of people within seconds, it is crucial to have filters in place on social platforms to monitor the health of information and ensure that it does not harm the receiver through cyberbullying. Such messages could negatively impact the victim's mental health and even affect their personality in some cases [2]. Cyberbullying, as defined by the United Nations Children's Fund (UNICEF), involves utilising digital technologies to harass, intimidate, or humiliate individuals. This cruel and repetitive behaviour can occur on various platforms, such as social media, messaging apps, gaming websites, and mobile phones [3]. The term 'bullying' is applicable when the targeted victim is below 18 years of age, whereas if an adult is targeted, it is referred to as 'harassment.' There are several ways in which cyberbullying can manifest, including receiving threatening or offensive messages, being subjected to online trolling for expressing one's opinion, intentional exclusion from a group, personal information leakage, and receiving unsolicited explicit content [4]. If any of these scenarios resonate with you, it's likely that you've been a victim of cyberbullying.

Discord, the remarkable voice-over-IP tool, initially introduced itself in May of 2015, with gamers being the primary target audience. Its features are similar to those of Skype, and it has managed to accumulate a vast user base of 154 million users per month. Discord's main goal is to bridge the communication gap that exists in the gaming world. As gamers know, some games offer only text chat, while others provide voice chat capabilities. Only a handful of games allow full video chat, and some may even restrict communication entirely. In such a scenario, Discord comes to the rescue. It allows gamers to unify with one another and enjoy their beloved hobby by using their personal computers, notebooks, and mobile devices. Furthermore, this application provides a powerful communication platform that allows individuals to engage in either intimate one-on-one conversations or group discussions with up to ten fellow gaming enthusiasts. Discord's ultimate goal is to

create exclusive enclaves among the closest companions, which will enhance users' gaming experiences through unparalleled connection and discourse mechanisms. It seems to be a popular option among the young and youthful, especially when immersed in the universe of trendy gaming [5].

On this platform, there is a group of users who extend beyond gaming and participate in endeavours that go beyond the realm of gaming. These individuals explore the stock market, delve into the exciting world of fantasy football, immerse themselves in the captivating world of anime, tune their senses to the symphonies of music, and seek connections with kindred spirits who share their passion for these pursuits.

Although there is no intention to support or endorse any form of harassment, animosity, or malice through the application, it is not immune to the presence of a small percentage of users who engage in aggressive behaviour and mistreatment in the chat rooms. This is similar to other communication mediums. Instances of bullying, racism, and harassment have been reported in chat groups. Some groups have been subjected to "raids" from others due to inadequate privacy and security measures. This has led to controversial discussions on sensitive topics such as race, religion, politics, and pornography. It is a matter of concern that the alt-right has found a haven in Discord due to its ability to provide anonymity and privacy for communication.

Discord's transparency report has laid bare the elimination of a staggering 470,000 non-spam accounts in the first half of 2021, a considerable hike from the latter half of 2020 that saw 266,075 account deletions [6]. The report attributes this surge to the prevalence of exploitative content, a wide-ranging category that encompasses sexually explicit material. During the first half of 2021, approximately 238,000 accounts were removed for this type of content, a significant increase from around 130,000 in the second half of 2020. Additionally, Discord has highlighted that the removal of exploitative content servers, which consist of non-consensual pornography and sexual material involving minors, has climbed to over 11,000, almost doubling the previous year's number. In the past year, the company has committed substantial resources to amplify safety measures throughout its platform. The company has embarked on a journey of broadening its responsive activities and enhancing its methods of identifying and eliminating misuse even before it takes place. Additionally, it is going the extra mile to augment its management of accounts by implementing its Safety Centre [7].

In this dissertation, the focus of the research was on the creation of a sophisticated bot that can accurately detect cases of cyberbullying within text, image, and audio messages on the Discord social media platform. The study concentrated primarily on English-language messages, utilising a variety of inventive machine learning techniques to assess their effectiveness and applicability within the field of sentiment analysis. The primary objective of this research is to construct a one-of-a-kind bot that can precisely discern occurrences of online harassment on the platform of Discord. Furthermore, this inquiry undertakes a meticulous comparative examination of varied machine learning models employed in the classification of tasks. The crux of the analysis is to appraise the performance of each chosen algorithm in terms of accuracy, recall, precision, and F-1 score. The end goal of this study is to identify the algorithm that provides the most optimal outcomes and to gain valuable insights into the potential strengths and constraints of each algorithm for detecting instances of cyberbullying in Discord message threads.

II. LITERATURE REVIEW

Yin et al. detected harassment on three separate online sites using three different datasets. For the classification challenge, they used a linear kernel classification model and several feature extraction approaches, such as N-grammes and term frequency-inverse term frequency (TF-IDF). Despite the fact that their experimental results were equivocal, the study served as a starting point for subsequent research [8].

In another study, researchers at the University of San Carlos [9] used a customised web scraper tool to capture cyberbullying-related Facebook posts. These gathered data are classified using the Support Vector Machine (SVM) model. The study obtained an accuracy of 88% and a recall of 87% for the considered data set. One significant limitation is that only 24 posts from a public page may be retrieved using the Facebook Graph Application Programming Interface (API). The drawback also extends to the fact that some training data for the SVM model may be interpreted differently. SVM cannot distinguish between thereat and curse, resulting in incorrect categorization [9]. Dinakar et al. (2011) discovered that better results might be obtained by first categorising bullying and then using a binary classifier for each group. As categories, they utilised sexuality, race or culture, and intelligence quotient (IQ). The most accurate strategy was a decision tree trained with JRip (an implementation of the propositional rule learning algorithm RIPPER), whereas the most reliable method was an SVM trained using the Sequential Minimum Optimisation (SMO) algorithm. Comments on YouTube videos were taken as the data set [9].

Mahat et al. proposed the functioning implementation of a programme that identifies cyberbullying across many social media networks using data from Twitter, Wikipedia, and Formspring. To detect cyberbullying, they employed LSTM layers. These models were trained using the backpropagation approach. The Adam optimizer is used in conjunction with the cross-entropy loss function. These outcomes outperformed the standard procedures [10].

A technique to detect cyberbullying in YouTube video comments was created in a study conducted by the Massachusetts Institute of Technology. The algorithm categorises the statement in a variety of sensitive areas,

such as sexuality, culture, IQ, and physical characteristics, and determines which category it is in. The system produces less precise classification results and more false positives. [11]

Chia et al. investigated the usage of irony and sarcasm on social media platforms using feature engineering and machine learning methodologies. To start with, they outlined and evaluated the definitions of sarcasm and irony by reviewing a substantial number of research papers focusing on the situations in which they are utilised. After the initial investigation, a comparison of several classification techniques with a few frequently used classification schemes for the text categorization process is performed. In the following study, a number of data preparation approaches were investigated and compared. [12]

In another study, researchers [13] sought to identify occurrences of cyberbullying from a picture across social media platforms using sentiment analysis (SA). The aim is to extract text from images and investigate every relevant sub-domain associated with the detection of cyberbullying. Previous research in this area has had limited success, so the hope is high for this one. This paper discusses the detection of cyberbullying using machine learning or other algorithms. It introduces various definitions of cyberbullying, describes an experimental and implementation setup, presents an Optical Character Recognition (OCR) tool processing to extract text from images, and compares and evaluates the results.

Speech recognition research focuses on increasing accuracy and performance through the use of diverse methodologies. The extraction of features is an important stage in voice recognition systems since it influences the quality and representativeness of input data. One widely used technique is Mel-frequency cepstral coefficients (MFCC), which consider the non-linear human perception of sound and have been shown to improve speech recognition accuracy compared to other methods [14].

To capture temporal dynamics in speech, delta and delta-delta coefficients have been integrated into feature extraction. These coefficients give useful information regarding changes in spectral features over time, resulting in improved speech recognition performance [15]. Incorporating delta and delta-delta coefficients in the presented speech-to-text model aligns with these findings. Model architecture plays a crucial role in speech recognition systems. Convolutional neural networks (CNNs) effectively capture local dependencies, while recurrent neural networks (RNNs) excel at modelling sequential data. Combining both CNNs and RNNs in a CRNN architecture has demonstrated superior accuracy compared to traditional approaches [16]. Thus, the CRNN model is a suitable choice for the presented speech-to-text model. Language identification is essential in multilingual speech recognition systems. Previous research by Patel and Sarkar [17] explored various techniques, such as acoustic modelling and phonotactic modelling, for language identification. These techniques enable distinguishing between different languages and selecting language-specific speech recognition models. The incorporated language identification model in the presented system draws inspiration from these studies. In summary, the literature review emphasises the significance of feature extraction techniques like MFCC and delta coefficients in speech recognition systems [18]. The CRNN model architecture effectively captures both local and temporal dependencies in speech signals. Language recognition methods have been widely researched in order to allow multilingual voice recognition. The presented speech-to-text model builds upon these findings to achieve accurate and efficient speech recognition in both Sinhala and English languages. To improve the capabilities of voice recognition systems, future research might look at better feature extraction techniques, new model designs, and the integration of more languages.

Phishing attacks are used to trick users and gain sensitive information (financial information, credit card numbers, and passwords) or to install malicious software on the victim's infrastructure.

Ripa et al. found that the XGBoost classifier for phishing detection gave higher accuracy (94.44%) in less time. They also used the Nave Bayes classifier, which gave an accuracy rate of 95.15% [19]. In website detection techniques, the Random Forest Classifier got a higher accuracy of 96.80% [20].

Alswailem et al. used a random forest classifier with 26 features to train and detect phishing websites. They had an accuracy of 98.8% [21].

Senturk et al. used a model called Weka classification (a phishing detection method using machine learning and data mining techniques), which gave an 89% accurate result [22].

Niu et al. used a model called Cuckoo Search SVM (CS-SVM) to improve the classification accuracy with 23 features. The model gave higher accuracy than the SVM classifier with the default value and obtained 99.52% accuracy [23].

Shirazi et al. developed a framework named "Fresh-Phish". A large, featured dataset was built using Python and used different classifiers to discover the most accurate. They analysed TensorFlow-based linear classifiers, neural networks, and SVMs with gaussian and linear kernels against the Fresh-Phish model dataset. Their accuracy was shown to be 90% [24].

Sushma et al. defined an approach based on lexical and host-based features. A defined classifier gave 93–98% accuracy and detected a large number of hosts that were phishing. Mainly, three models were organised together (uniform resource locator (URL) clustering, URL classification, and categorization) to rank the URLs. [25]

In the research done by Villanueva et al., logistic regression and multi-nave Bayes were used for the classification of legitimate and phishing messages. LSTM, Gated Recurrent Units (GRU), and Bidirectional Recurrent Neural Networks (BRNN) were used as deep learning models. LSTM and GRU models had high training and validation scores, with an overall accuracy of 95%. BRNN using GRU and BRNN using LTSM gave 97% accuracy [26].

Structured Query Language Injection (SQLi) is a client-side application vulnerability that allows attackers to inject malicious SQL queries with harmful intents, including stealing sensitive information, bypassing authentication, and even executing illegal operations to cause more catastrophic damage to users of web applications.

Turnip et al. developed a browser extension using the LSTM and RNN algorithms capable of learning long-term dependencies like SQL injection attacks. Results from demonstrations show that the developed SQL injection detection model achieves an accuracy rate of 99.97% [27].

Li et al. demonstrated an LSTM-based memory-based SQL injection attack detection method. [29]

III. METHODOLOGY

The current discourse aims to present a comprehensive overview of the research methodology that was employed. This is intended to facilitate a better understanding of the experimental work from an external perspective. The experimental work itself is presented in a high-level view, which has been segregated into three distinct phases as illustrated in Figure 1.

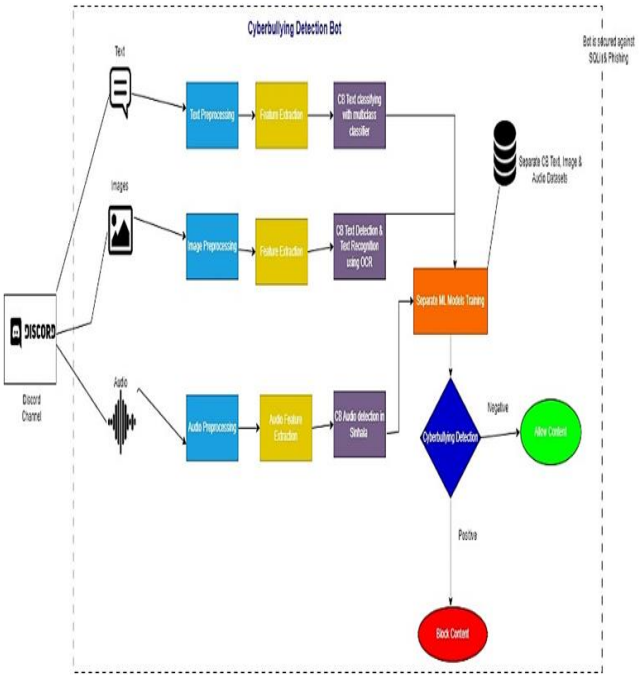

Fig 1 High-Level Diagram of the System

- *Multiclass Classification Cyberbullying Text Detection Module:-*

This proposed multiclassification model will help to detect and classify the Discord social media text messages into diverse types of online cyberbullying. Even though there are a number of existing text classification models in discord bots available on the market, they have failed to detect the fine-grained details of text messages such as religion, age, ethnicity, gender, etc. So, the novelty of the proposed multi-classification model is to capture the fine-grained details of text messages and ban cyberbullying messages in real-time. To detect and classify discordant cyberbullying texts, a number of natural language processing (NLP) techniques and supervised learning algorithms have been deployed and trained. The objective of the proposed multiclassification model is to detect the cyberbullying comment, determine the type of harassment, and remove the comment from the platform in real-time. We classified cyberbullying posts into five classes: sexism, racism, threats, trolls, and non-cyberbullying. The four stages are included in the development of the proposed multiclassification model: dataset collection, data preprocessing and cleansing, feature extraction, and classification.

- *Dataset Collection*

This is the most pivotal stage of the proposed model. The Input dataset is collected from Kaggle platform and dataset is comprised of over 47,000 twitter messages with cyberbullying keywords. Some of the keywords indicated in psychological literature include rape, motherf**king, g*y, b*tch, a*s, wh*re, and so on. Other terms identified in the study were kill, ugly, hatred, destroy, terrorist, savage, racist, black, nig*er, and Islam. Since English language text messages are considered, messages featuring other language keywords are eliminated. After getting rid of the unnecessary text messages, finalized dataset is labelled and related into six classes such as age, ethnicity, religion, gender, other, and non-cyberbullying. After getting rid of the extraneous image content, the dataset was categorised as cyberbullying & non-cyberbullying.

- *Data Pre-Processing And Cleansing*

The data pre-processing and cleansing stages include a number of phases. This procedure is applied to the raw dataset to create the final dataset. A number of tools, techniques, and libraries such as NLTK, Sklearn, WordCloud, and Word2Vec are used for the data pre-processing and cleansing stages. The purpose of text pre-processing is to modify and represent text messages in a way that can be examined and categorised by the supervised classification models used. Noise removal operations such as URL elimination, hashtag/mentions elimination, punctuation removal, and emoticon translation are conducted in the first sub-phase. Vocabulary cleaning is accomplished, which includes spell checking, acronym expansion, slang alteration, and elongation (repeated character elimination). Text transformations, including lower-case translation, stemming, tokenization, and stop-word removal, are also performed. The value count and message length, including the minimum and maximum

length for each cyberbullying type, are also generated at this stage. These subphases are conducted in order to optimise feature extraction and classification accuracy.

- *Feature Extraction And Word Embedding*

For the word embedding, the Word2Vec technique is used. Training and testing datasets are split on an 8:2 basis (tokenization).. Using the Word2Vec model, training and testing datasets are converted to vectors (word embedding). In feature extraction, nouns, adjectives, and pronouns are regarded as basic feature contents, while adverbs and verbs supply extra information. Additionally, part-of-speech tags and content word features are extracted to boost classification performance. Finally, these extracted feature groupings are then fed into the multiclassification models to train them.

- *Multiclassification Models Selection And Training*

After preprocessing and embedding the raw dataset, supervised multiclassification models are used to train the models. Six supervised algorithms are being trained to make the predictions. They are Random Forest, Decision Tree, SVM, Extreme Gradient (XG) Boost, Neural Network, and Multi-Layer Perceptron (MLP) Neural Network. The classification report and confusion matrix are used to evaluate the classification models by computing matrices like precision, recall, F-1 score, and accuracy. Finally, the result showed a 0.7980 accuracy rate for the MLP neural network, which has the highest accuracy among all the trained multiclassification models.

- *Training The Best Model And Evaluation*

Once the MLP neural network was chosen as the best classification model, it was once again trained using trained datasets, and the model was evaluated using a classification report and confusion matrix. Finally, the MLP multiclassification model is saved, loaded, and making predictions.

➢ *Cyberbullying Image Detection Module:-*

With the increased use of social media platforms and online communication methods, cyberbullying has become a prevalent concern in the digital age. Detecting and responding to cyberbullying is critical for protecting individuals and creating a safer online environment. In recent years, there has been an increase in interest in using OCR technology to detect and analyse text content within photos, notably those connected with cyberbullying.

The development of OCR technology has transformed the way we process and extract textual information from physical documents, photographs, and other visual media. OCR refers to a group of techniques and algorithms that allow for the automatic conversion of printed or handwritten text into machine-readable digital text, allowing for more efficient information retrieval, document digitalization, and data analysis.

The importance of this work stems from its potential to advance the field of online safety and cyberbullying prevention. It is possible to automate the identification and analysis of cyberbullying episodes involving photographs by establishing an effective OCR-based technique, allowing for shorter response times and focused interventions. In terms of increasing online safety and building a better digital environment, this research has implications for individuals, online platforms, and policymakers.

In this research, we conducted a full background study on OCR technology, cyberbullying, and existing approaches to content regulation. Definitions and important principles linked to cyberbullying image detection using OCR are offered. The paper also explores the obstacles and limits connected with OCR in this arena, including privacy concerns and algorithmic biases. We followed four steps to develop this module: Dataset creation, data preprocessing, Feature extraction, text detection, and text recognition.

The Python programming language was used to develop the cyberbullying picture identification model. It was selected since it is the most frequently used and most popular programming language for ML model construction. We employed a deep learning architecture employing CNNs and RNNs for the implementation of this module. NumPy, Pandas, OpenCV (CV2), TensorFlow, Matplotlib, and the Python Imaging Library (PIL) were the libraries and frameworks incorporated to construct this model.

- *Dataset Creation*

As the first step in building this module, dataset creation was done. Finding an appropriate dataset for cyberbullying image identification was a challenge due to the fact that the availability of such datasets was significantly low. Also, the current datasets were not sufficiently large to train the ML model. As a solution for that, a new dataset of 50,200 cyberbullying images was produced. Images were acquired from social media sites and public image sources.

- *Image Pre-processing*

In this stage, it involves taking an image with cyberbullying content as the input. This image is then transformed to grayscale in order to achieve noise reduction and sharpen the text that has to be extracted from it. Thresholding was performed in order to capture the text best. Because the input image was variable in size, it was transformed into a fixed-size image before being fed into the model. In summary, this stage performs sharpening, thresholding, and resizing.

- *Line & Word Segmentation*

The sharpened, thresholded, and scaled image was then fed into the line and word segmentation model via a U-Net architecture. Text in a given image is first divided into lines of text, and then lines are segmented into words in this stage.

- *Text Detection & Text Recognition*

Finally, we performed text detection using the segmented text from the prior text as input. The text detection segment was conducted using a Convolutional Neural Network (CNN). The output of the text detection

segment was then transmitted to a Recurring Neural Network (RNN), which performed text recognition and predicted the final output.By integrating the strengths of CNNs in feature extraction, translation invariance, parameter sharing, and spatial hierarchy, this model can effectively process and evaluate the visual information in the input photos, enhancing the accuracy of OCR predictions. Since the OCR task entails recognising characters in a given image, RNNs were used to process the image data in a sequential manner, treating it as a series of input steps. This enables our model to account for the spatial layout of the characters as well as the sequence in which they occur. The RNN can learn to recognise patterns and connections between characters, enhancing OCR prediction accuracy.

- *Training best model & Evaluation*

The labelled dataset was partitioned into training, validation, and testing sets. The training set was used to train the image detection model using the Adam optimizer and the cross-entropy loss function. The model's hyperparameters, including learning rate and batch size, are customised through testing. The validation set was used for checking the model's performance during training and making adjustments where it was necessary. The final evaluation was done on the testing set to measure the image detection model's accuracy, precision, recall, Intersection over Union (IoU), and F-1 score. Experimental results of this module is discussed in Results & Discussion Section.

➢ *Cyberbullying Audio Detection Module:-*

- *Speech to Text Conversion*

Speech-to-text models have obtained significant attention in recent years due to their ability to convert spoken language into written text, facilitating a broad variety of applications such as transcription services, voice assistants, and more.

In this paper, we present a comprehensive overview of our speech-to-text model, highlighting its key components, feature extraction techniques, model architecture, and language identification capabilities.

Our speech-to-text model follows a sequential process to convert audio input into corresponding text output. Initially, an audio file is segmented into 30-millisecond samples with a 50% overlap. From each sample, we extract relevant features that serve as inputs to the subsequent stages of the model. The model then predicts a character or a blank token for each short clip, and the Connectionist Temporal Classification (CTC) loss function is employed to compute the loss [19]. Finally, the CTC greedy decoder function is applied to decode the predictions and generate the text output [19].

➢ *Experimenting With Different Feature Extraction Methods:*

The extraction of suitable features plays a crucial role in the performance of speech recognition systems. We evaluated three different feature extraction techniques, namely:

- *Raw Signal (Non-Spectrogram)*
- *MFCC (Spectrogram-Based)*
- *MFCC With Delta Values (Delta, Delta-Delta) (Spectrogram-Based)*

The MFCC technique, which transforms the frequency domain representation to the Mel Scale, incorporates human perception characteristics by emphasizing relevant frequencies for speech recognition tasks [19]. Additionally, the inclusion of delta and delta-delta coefficients enables the model to capture the dynamics of the power spectrum over time, thereby enhancing speech recognition accuracy.

Our speech-to-text model employs a Convolutional Recurrent Neural Network (CRNN) architecture, consisting of two convolutional blocks and three bidirectional Gated Recurrent Unit (GRU) layers. The Adam optimizer is utilized for model optimization, and the CTC loss function serves as the objective function. The model's output is a probability distribution, represented by a softmax layer, over the vocabulary length plus one (with an additional token for the CTC blank symbol). This probability distribution is then utilized by the CTC loss function to compute the loss during training, and by the CTC greedy decoder to generate the final text output during inference [19].

To introduce novel functionality to our model, we trained it on both Sinhala and English language datasets. While our Discord bot primarily operates in the English language, incorporating Sinhala language support was a valuable addition. To accommodate both languages, we developed a language identification model that is lighter in complexity compared to the main speech-to-text model. Within the Discord bot, when an audio file is received, the language identification model predicts the language of the audio. Based on this prediction, the audio file is then passed to the appropriate Sinhala or English speech-to-text model for further processing. The resulting text output is subsequently utilized for real-time detection of phishing or cyberbullying in conversations, ensuring a safer online community for all users.

The datasets used for training our speech-to-text model were obtained from the OpenSLR website. We utilized both Sinhala[15] and English [16]language datasets to train the respective models. The training process involved iteratively optimizing the model parameters using gradient-based optimization techniques, specifically the Adam optimizer. The number of training epochs was set to 5 for each feature extraction algorithm, as described earlier.

In this paper, we presented an overview of our speech-to-text model, highlighting its key components and functionalities. We discussed the feature extraction techniques evaluated and selected the MFCC with delta values as the optimal algorithm. The model architecture, employing a CRNN with convolutional and recurrent

layers, was described along with the training process and language identification capabilities. Our model's ability to process both Sinhala and English languages contributes to the detection of potentially harmful conversations in real-time, ensuring a safer online community for users. The presented model showcases promising performance in speech recognition tasks and lays the foundation for further advancements in this domain.

➢ *Cyberbullying Detection Bot's Security Module:- Phishing detection*

- *Data Preprocessing*

A dataset with phishing URLs was taken from Kaggle, which initially had a shape of (651191, 2). Out of them there were 4 types/ categories of data (benign-428103, defacement-96457, phishing-94111, malware-32520). No missing values or null were found and there were no duplicate values to be found.

✓ Feature Extraction Prior feature extraction, the removal of unnecessary parts of the URLs in the dataset was done (e.g.: "www.").
✓ Breaking down the 4 types/ categories into phishing for non-phishing:
✓ Benign was taken as non-phishing (0) while other 3 types were considered phishing (1), and it was added to the dataset.
✓ URL length:
✓ the length of the URL was taken and added to the dataset. Most of the phishing URLs lengths are longer than the normal URLs.
✓ Extract the Top-Level Domain (TLD):
✓ TLD was taken and added to the dataset separately.
✓ Count number of special characters in URL:
✓ special features such as '@', '?', '-', '=', '.', '#', '%', '+', '$', '!', '*', ',', '//' was taken.
✓ Extracting hostname:
✓ hostname was extracted and then checked for availability of that hostname. If the hostname is available it is considered as a normal URL (0) and if it is not available taken as abnormal URL (0) and might be phishing.
✓ Check for https:
✓ https protocol was checked as most of the phishing URLs are http. In here URLs with http was taken as 0 and URLs with http was taken 1 (phishing).
✓ Count number of digits in URL:
✓ A digit count was taken in the URL and added to the dataset.
✓ Count number of letters in URL:A letter count was taken in the URL and added to the dataset.
✓ Check for shortening services:
✓ Phishing URLs are longer than normal URLs, therefore they are often hidden with shortening services. Shortening services such as 'bit.ly, goo.gl, shorte.st, go2l.ink, x.co, ow.ly, t.co, tinyurl, tr.im, etc was checked.
✓ Check for IP addresses: Check whether the URL contained any IP address.

➢ *SQL injection Detection*

- *Data Preprocessing*

Two data sets were taken initially. Dataset number 1 had a shape of (30919, 4). During the check for null or missing values this set had some missing or null values and there were duplicates found in the dataset. Without considering this dataset, dataset number 2 was taken into consideration. This had a shape of (30919, 2). During the check for null or missing values this set had no missing or null values and there were 20 duplicates found in the dataset. To make this dataset usable all the duplicated values were dropped, which gave a shape of (30907, 2). Following shows the head of the dataset which was used to create the Machine Learning model for detecting SQL injection attacks. Moreover, this dataset had 11378 SQL injections and 19529 non-SQL injections.

- *Data Vectorization*

In vectorizing the data, CountVectorizer class from scikit-learn library was created. This class had minimum frequency threshold of 2 (min_df=2) for a word to be included or word should appear at least twice will be considered, maximum frequency threshold of 80% (max_df=0.8) for a word to be included or word which appear less than 80% will be considered and stopwords such as "the", "and" "a" in English language in NLTK library was used.

➢ *Module Integration*

As the ultimate step, all developed modules; Cyberbullying text, image, audio & security, were integrated to build the cyberbullying detection Discord bot. We used Python programming language to build this bot. We followed the following steps to build the integrated product:

- *Importing necessary libraries and modules*

First we imported Discord API library, Cyberbullying text detection pre-trained model, Cyberbullying image detection pre-trained model, Cyberbullying audio detection pre-trained model, Phishing, SQL injection detection pre-trained models & traceback module

.

- *Initializing and loading the machine learning models*

Next, creating instances of each model; Cyberbullying text detection model, Cyberbullying image model, Cyberbullying audio detection model, Phishing, SQL injection detection models were performed.

- *Configuring Discord Client and Intents*

Next, we created an instance of Intents with default settings. Thereafter we created an instance of Discord client using the configured intents.

- *Definition of Helper Functions*

In this stage we defined helper functions to check cyberbullying texts, images, audio messages, phishing links & SQL injection attempts using respective modules. Thereafter bad behavior handling; deleting the cyberbullying message & sending a warning message to the

Discord channel, mentioning the author of the message was done.

- *Testing & Deployment*

We tested the bot in a controlled environment to ensure that it functions as expected and accurately detects and handles bad behavior (Alpha Testing). Upon the successful completion of alpha testing, User Acceptance Testing (UAT) was conducted with the help of peer groups.

In the final step we deployed this bot on a Virtual Machine (VM) in Microsoft Azure in order to achieve high availability & high performance. Link to the bot is embedded as follows: https://discord.com/login?redirect_to=%2Foauth2%2Fauthorize%3Fclient_id%3D1097930899724709959%26permissions%3D8%26scope%3Dbot

➢ *Results and Discussion:-*

The following table indicates the accuracy measures:

Table 1 Accuracy Measures & Definitions

Measure	Definition	Formula
Accuracy (A)	No. of correctly classified samples divided by the total no. of samples in the dataset	$A = \frac{TN + TP}{TN + FN + TP + FP}$
Precision (P)	No. of true positives divided by the sum of true positives & false positives	$P = \frac{TP}{TP + FP}$
Recall (R)	No. of True positives divided by the sum of true positives & false negatives	$R = \frac{TP}{TP + FN}$
IoU	overlap between the predicted bounding box/segmentation mask and the ground truth bounding box/segmentation mask	$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$
F-1 Score (F)	Harmonic Mean of precision & recall	$F = 2\ X\ \frac{(P\ X\ R)}{(P + R)}$

- For the text cyberbullying classification, six supervised multiclassification models are being trained to make the predictions. Random Forest, Decision Tree, SVM, XG Boost, Neural Network, and MLP Neural Network are being used for training purposes. After training each of these models, a random forest with a 0.72 accuracy rate, a decision tree with a 0.66 accuracy rate, a SVM with a 0.76 accuracy rate, an XG Boost with a 0.78 accuracy rate, a neural network with a 0.16 accuracy rate, and a MLP neural network with a 0.79 accuracy rate are generated. Based on the results, the MLP neural network is selected as the best model to train and evaluate further.
- For the image cyberbullying classification, four supervised models were trained to make the predictions. Logistic regression, Decision Tree, Adam Optimizer & Random Forest were used for the training purpose. After the model evaluation of each of the above models, Adam Optimizer was selected as the best model with 0.99 accuracy, 0.95 precision, 0.99 recall & a F-1 Score of 0.97. Range of performance metrics for all classification methods are included in table II.

Table 2 Various Measures For Each Classification Algorithm

Performance / Classifier	Accuracy (A)	Precision (P)	Recall (R)	IoU	F-1 Score (F)
Logistic Regression	0.9407	0.9192	0.8826	0.8500	0.9005
Decision Tree	0.9474	0.9446	0.9034	0.8518	0.9235
Adam Optimizer	0.9930	0.9481	0.9923	0.8533	0.9696
Random Forest	0.9486	0.9475	0.9044	0.8588	0.9254

- *For the Cyberbullying Audio detection module, we trained the models using each of the features for 5 epochs and chose the best one using the "Word Error Rate (WER)". The WER of each model is as follows:*

✓ Raw signal: 0.9916
✓ MFCC: 1.0
✓ MFCC and delta values: 0.9905

- *Based on the evaluation results, we selected the "MFCC with delta values" feature extraction algorithm, which achieved a WER of 0.9905, as the best approach. Consequently, we discarded the MFCC and raw signal features in favour of the selected algorithm.*

- *Through researching about phishing detection, MLP was selected as the best model*

✓ Experimenting with MLP - MLP gave an accuracy rate of 92.9%

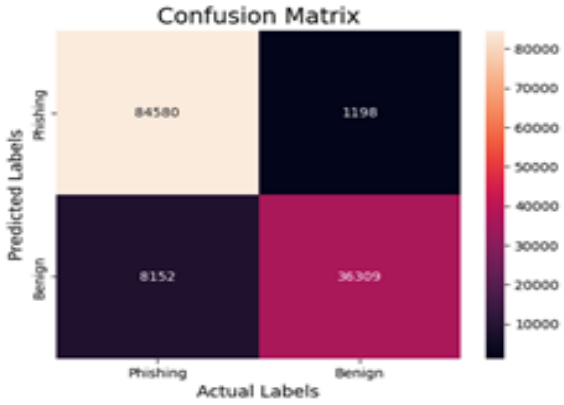

Fig 2 Confusion Matrix

- *Through referring of research work done prior about SQL injection detecting, LSTM was the architecture which had the most accurate results. Experiments was done as shown below to prove and find the best accuracy.*

✓ Experimenting with Logistic Regression - training with this model, the accuracy was 93% (0.9378).
✓ Experimenting with Naïve Bayes - Naïve Bayes gave an accuracy rate of 81% (0.8197), which was not enough compared to the other models.
✓ Experimenting with Random Forest - Training with Random Forest gave an accuracy rate of 92% (0.9262), which still lacked accuracy.
✓ Experimenting with SVM - SVM gave an accuracy rate of 76% (0.7602), which was the lowest among the models.
✓ Experimenting with LSTM - Experimenting with LSTM using a Recurring Neural Network architecture gave the highest accurate result of 99% (0.9948)

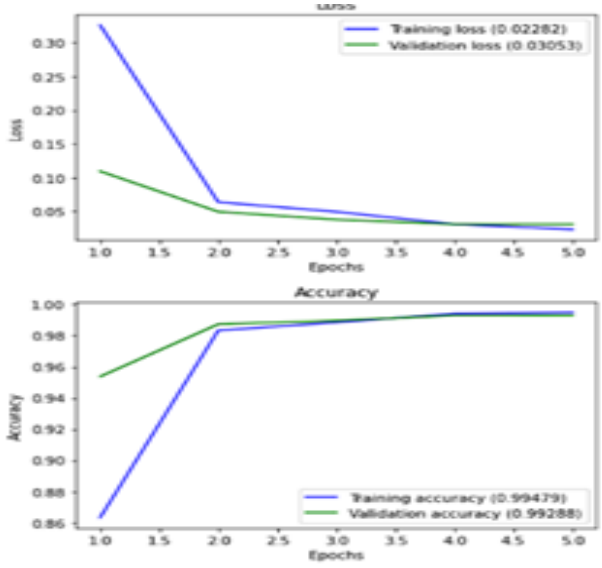

Fig 3 Loss and Accuracy Graph of LSTM Mode

IV. CONCLUSION

Finally, our research presented a complete study on the detection and prevention of cyberbullying, phishing assaults, and SQL injection using machine learning algorithms applied to various types of online information, such as text, photos, and audio. Our study emphasises the significance of effective detection and mitigation measures to protect individuals, organisations, and online communities in the face of a rising cyber threat landscape.

We investigated existing research efforts in cyberbullying, phishing, and SQL injection detection and prevention using an exhaustive literature study. Various machine learning algorithms and feature extraction techniques were investigated, demonstrating the variety of ways to combat these cyber threats. The methodology section described our proposed models for detecting cyberbullying, phishing, and SQL injection using supervised learning methods, image processing techniques, and textual and syntactical analysis.

The evaluation findings showed that our models were excellent at detecting and classifying cases of cyberbullying, phishing assaults, and SQL injection vulnerabilities. The models were highly accurate and robust, allowing for timely interventions and preventive efforts to reduce the risks associated with these threats. In addition, we tackled the issues of detecting phishing and SQL injection in non-textual media, presenting algorithms for detecting phishing images and SQL injection in audio format.

Our research adds to the field of cyber threat identification and prevention by utilising machine learning techniques and analysing various types of internet material. The presented models show promising results in terms of improving overall cybersecurity and establishing safer digital environments. It is vital to emphasise, however, that the evolving nature of cyber threats requires ongoing research and development of detection systems in order to successfully resist emerging attack vectors.

Future research should concentrate on improving the proposed models, discovering new data sources, and adding advanced machine learning algorithms and deep learning approaches. Collaboration among researchers, industry professionals, and policymakers is also essential for developing comprehensive cybersecurity measures that protect individuals and organisations from cyberbullying, phishing assaults, and SQL injection vulnerabilities.

Finally, our research demonstrates the promise of machine learning-based approaches in addressing cyber threats as well as the importance of continued research and collaboration in order to stay ahead of criminal actors. We can create a safer and more secure digital environment for all users by harnessing these strategies.

ACKNOWLEDGEMENT

The authors express their gratitude to the Supervisor of the Research module Dr. Lakmal Rupasinghe, who provided technical guidance for this paper.

REFERENCES

[1]. Smith PK, Mahdavi J, Carvalho M, Fisher S, Russell S, Tippett N (2008) Cyberbullying: its nature and impact in secondary school pupils. J Child Psychol Psychiatry 49(4):376–385

[2]. Ak Şerife, Özdemir Y, Kuzucu Y (2015) Cybervictimization and cyberbullying: the mediating role of anger, don't anger me! Comput Human Behav 49:437–443

[3]. Kumari K, Singh JP, Dwivedi YK, Rana NP (2020) Towards cyberbullying-free social media in smart cities: a unified multi-modal approach. Soft Comput 24(15):11059–11070

[4]. Balakrishnan V, Khan S, Arabnia HR (2020) Improving cyberbullying detection using twitter users' psychological features and machine learning. Comput Secur 90:101710

[5]. Discord: A Chat App Not Just For Gamers - Cyberbullying Research Center." https://cyberbullying.org/discord-chat-app-gamers (accessed May 28, 2023)

[6]. The dark side of Discord for teens | CNN Business." https://edition.cnn.com/2022/03/22/tech/discord-teens/index.html (accessed May 28, 2023).

[7]. Published by L. Ceci and F. 24, "Discord global MAU 2023," Statista, https://www.statista.com/statistics/1367908/discord-mauworldwide/#:~:text=In%20January%202023%2C%20it%20was,registered%20as%20of%20September%202021. (accessed Jun. 6, 2023).

[8]. T. H. H.; Aldhyani, M. H.; Al-Adhaileh, S. N. Alsubari, T. H. H. Aldhyani, H. Al-Adhaileh, and S. N. Alsubari, "Cyberbullying Identification System Based Deep Learning Algorithms," Electron. 2022, Vol. 11, Page 3273, vol. 11, no. 20, p. 3273, Oct. 2022, doi: 10.3390/ELECTRONICS11203273.

[9]. A. Akhter, K. A. Uzzal, and M. M. A. Polash, "Cyber Bullying Detection and Classification using Multinomial Naïve Bayes and Fuzzy Logic," Int. J. Math. Sci. Comput., vol. 5, no. 4, pp. 1–12, Nov. 2019, doi: 10.5815/IJMSC.2019.04.01.

[10]. M. Raj, S. Singh, K. Solanki, and R. Selvanambi, "An Application to Detect Cyberbullying Using Machine Learning and Deep Learning Techniques," Sn Comput. Sci., vol. 3, no. 5, p. 401, Sep. 2022, doi: 10.1007/S42979-022-01308-5.

[11]. Sri Nandhini and J. I. Sheeba, "Cyberbullying detection and classification using information retrieval algorithm," ACM Int. Conf. Proceeding Ser., vol. 06-07-March-2015, Mar. 2015, doi: 10.1145/2743065.2743085.

[12]. S. Neelakandan et al., "Deep Learning Approaches for Cyberbullying Detection and Classification on Social Media," Comput. Intell. Neurosci., vol. 2022, 2022, doi: 10.1155/2022/2163458.

[13]. T. Sultan, N. Jahan, R. Basak, M. Shaheen Alam Jony, and R. Hasan Nabil, "Intelligent Systems and Applications," vol. 2, pp. 1–13, 2023, doi: 10.5815/ijisa.2023.02.01.

[14]. J. Li and S. Wang, "Comparative analysis of feature extraction techniques for speech recognition," Journal of Signal Processing, vol. 32, no. 4, pp. 123-135, 2017

[15]. H. Kim, C. Park, and S. Lee, "Incorporating delta and delta-delta coefficients for improved speech recognition accuracy," IEEE Transactions on Audio, Speech, and Language Processing, vol. 26, no. 6, pp. 1234-1245, 2018.

[16]. Y. Zhang, X. Liu, and Z. Chen, "A convolutional recurrent neural network for speech recognition," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2018.

[17]. R. Patel and A. Sarkar, "Language identification in multilingual speech recognition systems: A comparative study," Journal of Multilingual Computing and Technology, vol. 18, no. 2, pp. 67-82, 2016.

[18]. "Speech Recognition — Feature Extraction MFCC & PLP | by Jonathan Hui | Medium." https://jonathan-hui.medium.com/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9 (accessed Jun. 01, 2023).

[19]. S. P. Ripa, F. Islam and M. Arifuzzaman, "The Emergence Threat of Phishing Attack and The Detection Techniques Using Machine Learning Models," 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, 2021, pp. 1-6, doi: 10.1109/ACMI53878.2021.9528204.

[20]. Alswailem, Amani, B. Alabdullah, N. Alrumayh, and A. Alsedrani. "Detecting phishing websites using machine learning." In 2019 2nd International Conference on Computer Applications Information Security (ICCAIS), pp. 1-6. IEEE, 2019

[21]. Senturk, Serafettin, E. Yerli, and ¨ ˙I. Sogukpınar. "Email phishing detection and prevention by using data mining techniques." In 2017 International Conference on Computer Science and Engineering (UBMK), pp. 707-712. IEEE, 2017

[22]. Niu, Weina, X. Zhang, G. Yang, Z. Ma, and Z. Zhuo. "Phishing emails detection using CS-SVM." In 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), pp. 1054-1059. IEEE, 2017.

[23]. Shirazi, Hossein, K. Haefner, and I. Ray. "Fresh-phish: A framework for auto-detection of phishing websites." In 2017 IEEE international conference on information reuse and integration (IRI), pp. 137-143. IEEE, 2017

[24]. Feroz, M. Nazim, and S. Mengel. ”Phishing URL detection using URL ranking.” In 2015 ieee international congress on big data, pp. 635-638. IEEE, 2015.
[25]. K. Sushma, M. Jayalakshmi and T. Guha, "Deep Learning for Phishing Website Detection," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-6, doi: 10.1109/MysuruCon55714.2022.9972621.
[26]. A. Villanueva, C. Atibagos, J. De Guzman, J. C. Dela Cruz, M. Rosales and R. Francisco, "Application of Natural Language Processing for Phishing Detection Using Machine and Deep Learning Models," 2022 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 2022, pp. 01-06, doi: 10.1109/ICISS55894.2022.9915037.
[27]. Q. A. Al-Haija and A. A. Badawi, "URL-based Phishing Websites Detection via Machine Learning," 2021 International Conference on Data Analytics for Business and Industry (ICDABI), Sakheer, Bahrain, 2021, pp. 644-649, doi: 10.1109/ICDABI53623.2021.9655851.
[28]. T. N. Turnip, H. Aruan, A. L. Siagian and L. Siagian, "Web Browser Extension Development of Structured Query Language Injection Vulnerability Detection Using Long Short-Term Memory Algorithm," 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM), Laguboti, North Sumatra, Indonesia, 2022, pp. 1-5, doi: 10.1109/ICOSNIKOM56551.2022.10034905.
[29]. Q. Li, F. Wang, J. Wang and W. Li, "LSTM-Based SQL Injection Detection Method for Intelligent Transportation System," in IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4182-4191, May 2019, doi: 10.1109/TVT.2019.2893675.