Revista ELECTRO, Vol. 47, 2025, pp. 329 - 334
https://itchihuahua.mx/revista_electro
ISSN Electrónico: 3061-774X

DESARROLLO DE UN SISTEMA PARA IDENTIFICACIÓN DE ACOSO EN REDES SOCIALES

A.A. Villalba- García[1], C.G. Nava-Dino[1], R.A. Acosta-Chávez[1],
M.C. Maldonado-Orozco[1], N.L. Mendez-Mariscal[1], J.M.S Aceves[2], J.P Flores -De Los Ríos[3]

[1]Universidad Autónoma de Chihuahua, Facultad de Ingeniería.
Circuito Número I s/n, Nuevo Campus Universitario, Nte. 2, 31125 Chihuahua
614-4429500

[2]Universidad Autónoma de Ciudad Juárez. Manuel Díaz H. No. 518-B Zona Pronaf Condominio, 32315 Juárez, Chih. México

[3]Tecnológico Nacional de México- Instituto Tecnológico de Chihuahua, Av. Tecnológico 2909, Chihuahua 31130, Chih. México.

ndino@uach.mx

RESUMEN

En el presente trabajo de investigación se desarrolló un modelo de aprendizaje automático capaz de detectar comportamientos de acoso en redes sociales como X, con el objetivo de facilitar la identificación automática y precisa de comportamientos negativos. Para ello, se utilizó un conjunto de datos de J. Wang (*SOSNet, A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection)* que cuenta con un conjunto de datos de entrenamiento. Se entrenó un modelo BERT para que este pudiera categorizar el texto en alguna de las categorías definidas anteriormente. En los resultados se observa que el sistema implementado, es capaz de identificar entre una variedad de tipos de acoso con un nivel de precisión aceptable.

Palabras Clave: Acoso en Redes, Aprendizaje Automático, Python, Redes Sociales, Comunicaciones.

ABSTRACT

In this research work, a machine learning model capable of detecting harassment behaviors in social networks such as X was developed with the aim of facilitating the automatic and accurate identification of negative behaviors; a dataset from J. Wang (*SOSNet, A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection)* A BERT model was trained to categorize the text into one of the categories defined above. The results obtained by the system, permits identify several types of bullying with acceptable precision.

Keywords: Cyberbullyng, Machine Learning, Python, Social Networks, Comunications.

1. INTRODUCCIÓN

El procesamiento del lenguaje natural es una de las ramas que se encuentra en una intersección entre la lingüística aplicada y las ciencias de la computación. El objetivo de éste es el estudiar los métodos necesarios para que una computadora pueda procesar de alguna manera los lenguajes naturales, como el español o el inglés.

La historia del procesamiento del lenguaje natural comenzó aproximadamente en la década de 1950, con una publicación de Alan Turing titulado «Inteligencia» en el que planteó el "Test de Turing", la cual implicó que una máquina tenía la capacidad de comunicarse en un lenguaje natural con un ser humano [1].

El procesamiento del lenguaje natural es un gran campo de estudio que tiene varias técnicas para extraer la estructura gramatical y su significado con el objetivo de ser procesado y que dé como resultado una tarea útil.

Actualmente, este campo de investigación es utilizado en muchos de los productos que se tienen en una casa, en muchas estaciones de trabajo o en aplicaciones dentro de cualquier dispositivo inteligente, como los asistentes virtuales, como Siri, Alexa o Google Assistant, los cuales también cuentan con funciones de análisis de voz y son capaces de buscar información; traducción automática, con empresas como Google y Microsoft [2].

Aunque existen más campos de investigación y desarrollo para el procesamiento del lenguaje natural, una de las más importantes para esta investigación será la capacidad para el análisis de sentimientos y emociones en textos de redes sociales. Esta es una

de las características que las empresas usan para medir la satisfacción del cliente y la retroalimentación de los empleados.

Después de la creación del "Test de Turing", la Universidad de Georgetown, en colaboración con la empresa IBM, en 1954, hicieron una demostración del experimento Georgetown-IBM, que consistía en una serie de más de sesenta frases que fueron traducidas del ruso al inglés.

Años después, en la década de 1970, surgieron los generadores de analizadores léxicos, conocidos técnicamente como *lexer*, y los generadores de analizadores sintácticos. Estos se encargaban de transformar un texto en *tokens*; después, estos se envían en un *parser*, el cual valida la secuencia de los *tokens* [3].

Estos simplifican la implementación de lenguajes de programación al tomar expresiones regulares como entrada y autómatas. Aunque actualmente existen varios lenguajes de programación con los que se puede trabajar con el procesamiento del lenguaje natural, en 1972, fue inventado Prolog, un lenguaje de programación lógico que tenía como objetivo facilitar las aplicaciones de éste gracias a su sintaxis, que fue diseñada específicamente para escribir gramáticas en un estilo más práctico.

El siguiente gran paso para el procesamiento del lenguaje natural fue al final de la década de 1980, en ésta hubo un gran progreso con la introducción de algoritmos de aprendizaje automático. Esto fue gracias al aumento constante de la capacidad computacional, descrito en por la ley de Moore [4]. Esto alentó a investigadores a centrarse en el desarrollo de modelos estadísticos; dichos modelos son utilizados en sistemas de reconocimiento de voz.

A partir del año 2000, se comenzó a ver una nueva tendencia hacia técnicas de procesamiento del lenguaje natural utilizando redes neuronales. El primer modelo de lenguaje basado en una red neuronal utilizaba una red *FeedForward* [5].

Para la década de 2010, se presentaron resultados que mostraron que las técnicas basadas en aprendizaje automático con el uso de redes neuronales fueron capaces de lograr resultados de vanguardia en muchas tareas del lenguaje natural.

La palabra *"bully"* proviene aproximadamente del año 1530, aunque su origen etimológico es más reciente, del año 1973, acuñado por el psicólogo Dan Olweus para referirse a la violencia en el 9 ámbito escolar [6]. Es su significado más básico, involucra a dos personas, un *bully* o más y una víctima. El *bully* agrede a la víctima de manera física, verbal o de cualquier manera que le haga tener un sentimiento de superioridad y poder. Dichas acciones pueden ser directas, como golpes o insultos cara a cara; o indirectas, como esparcir rumores sobre la víctima.

El *bullying* junto con el crecimiento exponencial que mostraron las redes sociales en los últimos años ha permitido que el *cyberbullying* sea aún más frecuente, convirtiéndose en un problema social que afecta a jóvenes que usan internet o redes sociales actualmente. Este puede causar baja autoestima, ansiedad, depresión, desesperanza, e incluso puede llevar a la falta de motivación para vivir, lo que puede llevar a la muerte de la víctima; además de presentar problemas físicos como cansancio o las víctimas experimentan síntomas como dolores de cabeza o de estómago; también se pueden presentar bajo rendimiento académico, fracaso o rechazo escolares.

Como una manera de poder ayudar a regular este tipo de actividades, se planteo desarrollar un identificador de *cyberbullying* que permita identificar casi en tiempo real los *tweets* que puedan contener *cyberbullying* de cualquier tipo. Así, es posible tomar medidas y precauciones para solucionar el problema y aminorar el daño psicológico que se pueda causar en la víctima.

2. PROCESAMIENTO DE LENGUAJE NATURAL

2.1. Conjunto de Datos

Para abordar el tema del *cyberbullying* en redes sociales es necesario conocer el funcionamiento de la plataforma que estamos analizando, en este caso X (antes Twitter). Esta es una red social que se enfoca en *posts* de una longitud de 260 caracteres, en los cuales se pueden encontrar oraciones completas, *emojis*, *hashtags* o menciones a otros usuarios de la red.

Actualmente se cuentan con diferentes métodos con los que, gracias al procesamiento del lenguaje natural, es posible realizar un análisis de sentimientos sobre algún texto; pudiendo ser desde un método de clasificación binaria (como el expresar un sentimiento positivo o negativo), hasta un método de clasificación compleja que nos permita identificar algún sentimiento en específico entre una gama variada de sentimientos.

Prácticamente siendo una regla, en cualquier modelo propuesto anteriormente se necesita un *dataset* (conjunto de datos), el cual debe de ser preprocesado dependiendo de las necesidades que presente el proyecto y el modelo propuesto, ese mismo preprocesamiento será aplicado al texto que se quiera clasificar en un futuro.

El preprocesamiento, independientemente del idioma que se esté analizando, normalmente cuenta con una *tokenización* del texto, el cual es el proceso de convertir secuencias de caracteres en unidades más pequeñas que pueden ser procesadas de manera semántica. El resultado de la *tokenización* es una lista de palabras [7].

2.2. Algoritmos de Procesamiento

Existen distintos algoritmos para procesar la información, entre ellos se consideraron a:
Naive Bayes que es un algoritmo de clasificación de machine learning basado en el teorema de Bayes. Este algoritmo comparte características con otros algoritmos de aprendizaje supervisado, principalmente el hecho de que utiliza las características de los datos conocidos para predecir la clase de los datos que se desean clasificar [8].

Support Vector Machines que son un tipo de algoritmos de aprendizaje supervisado usados comúnmente en tareas de regresión y clasificación. Es ampliamente utilizado en campos de reconocimiento de patrones, análisis de imágenes y procesamiento del lenguaje natural [9].

Así como a *BERT* que su nombre proviene de (*Bidirectional Encoder Representations from Transformers*). Es un modelo desarrollado por Google que ha tenido un alto impacto en el procesamiento del lenguaje natural. Es un modelo de transformadores preentrenado con un corpus grande de manera autosupervisada [10].

2.3. *Data Set* de Entrenamiento

El *dataset* de entrenamiento utilizado en este proyecto cuenta con categorías que representan diferentes tipos de acoso, como acoso racial, basado en la xenofobia, de salud o seguridad, entre otras categorías. Se cuenta con un *dataset* de pruebas, del cual la información útil para las pruebas del modelo, son simplemente los *tweets* que contiene.

Los *tweets* del *dataset* de entrenamiento tienen que pasar por varias técnicas de preprocesado para que tengan las características adecuadas para ser utilizados en la creación de un modelo que pueda detectar el acoso en algún texto. Este modelo va a ser evaluado, comparándolo con resultados previamente etiquetados en un *dataset* de prueba, el cual es similar al *dataset* de entrenamiento, pero no cuenta con 3 columnas.

Para el preprocesamiento de los datos se siguió el siguiente proceso:

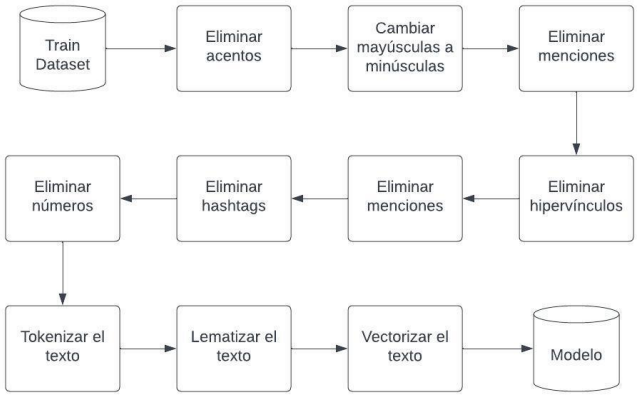

Figura 1. Proceso para el preprocesamiento de datos

Se obtienen los *tweets* del *dataset*, pero este texto no se encuentra listo en la base de datos para ser utilizado para entrenar un modelo.

Se eliminan los acentos, ya que éstos, aunque estos puedan brindar un significado a la palabra al momento de hablar, estos pueden no brindar significado para el procesamiento; muchas veces es necesario hacer este proceso, aunque de manera escrita sí cambie el significado de la palabra cuando ésta se encuentra acentuada.

Se cambian las mayúsculas a minúsculas, ya que, así como con la acentuación de las palabras, no se obtiene una diferencia. Además de que, para evitar la duplicación de datos en el modelo, se usan las palabras escritas de una sola manera; ya que para el modelo "entrenar" sería diferente de *"Entrenar"* o de *"ENtrenar",* aunque las tres palabras signifiquen lo mismo.

Se eliminan las menciones de otros usuarios; aunque esto no es una regla, para este ejercicio no nos brinda ningún significado o emoción. Para el entrenamiento del modelo, se necesita *tokenizar*, l*ematizar* y *vectorizar* el texto.

El modelo se desarrolló en el lenguaje de programación Python, en su versión 3.11. Para asegurar la compatibilidad entre las librerías que se utilizaron se usaron variables de entorno, en las cuales se incluyen las siguientes librerías, aunque solo se enlistan las más relevantes: *Natural Language Toolkit*, *Transformers* (Una plataforma de código abierto, desarrollada por la empresa Hugging Face). Esta librería proporciona herramientas para el desarrollo de modelos de machine learning, clasificación, sintetización y generación de textos, reconocimiento de entidades o el desarrollo de modelos capaces de contestar preguntas. Una característica importante de esta librería es la 23 que da acceso al modelo preentrenado BERT.
Keras que es una librería de alto nivel de *TensorFlow* para construir y entrenar modelos de aprendizaje profundo [11].

Revista ELECTRO, Vol. 47, 2025, pp. 329 - 334
https://itchihuahua.mx/revista_electro
ISSN Electrónico: 3061-774X

Con la intención de evaluar la estabilidad del modelo y cerciorarse de que tan uniforme es el *dataset* [12] con el que se cuenta, se realizó el entrenamiento tres veces, esto ya que, normalmente, el entrenamiento de estos modelos se realiza con inicializaciones aleatorias que podrían afectar los resultados. En la figura 2, se puede observar que el modelo tiene el comportamiento esperado en cualquiera de los entrenamientos realizados:

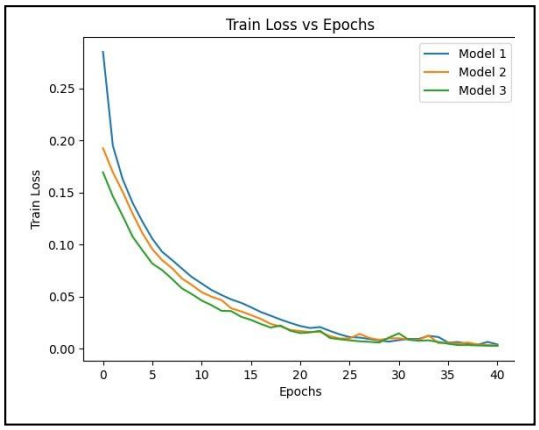

Figura 2. Entrenamiento del mismo proceso 3 veces.

Además, en la figura 3 también se puede observar que los modelos detuvieron su entrenamiento alrededor de la época 40, esto se debe a que el entrenamiento está programado para detenerse si no se encuentra mejora significativa en su *train loss* durante las últimas épocas.

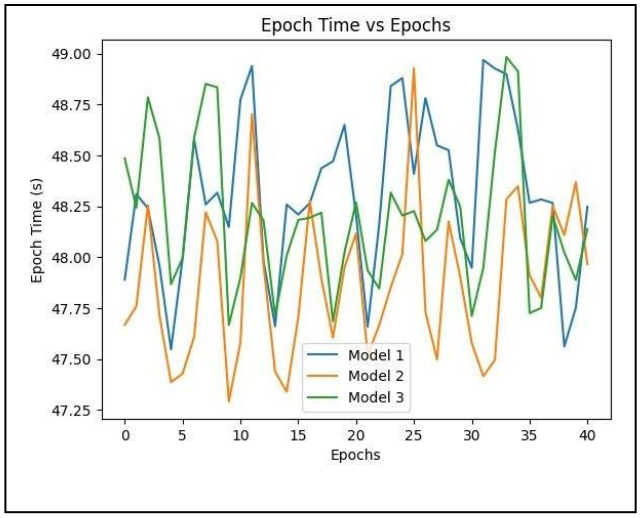

Figura 3. Tiempo de entrenamiento por época.

Siguiendo los mismos modelos de la figura 2, la figura 3 muestra la duración de las épocas (en segundos) frente al número de épocas para los tres modelos. Esta variación de tiempos también puede demostrar que los modelos fueron entrenados con el mismo *dataset*, pero fueron inicializados con datos diferentes. En la figura 4, se muestra el diagrama de flujo del modelo.

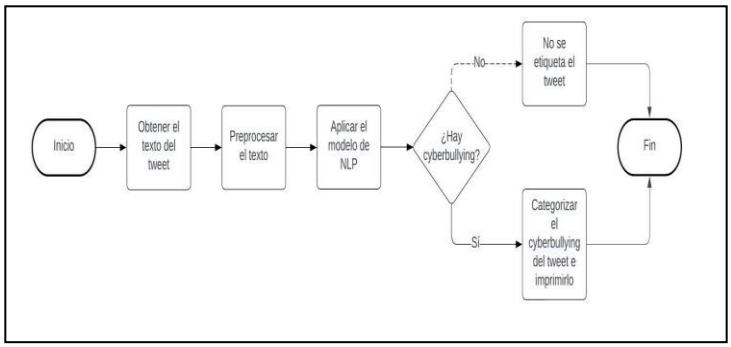

Figura 4. Diagrama de flujo del modelo.

2.4. Resultados

A lo largo del desarrollo de este modelo se encontraron diversos problemas. Uno de los principales, fue la escasez de datos en los *datasets* disponibles, y la complejidad de crear un *dataset* más grande.

El *dataset* contiene un aproximado de 3053 registros, los cuales son suficientes en promedio en comparación de un *dataset* más desarrollado. Después de experimentar e intentar crear un modelo entrenado únicamente con los datos disponibles, se encontraron problemas en los que los modelos desarrollados no predecían de manera satisfactoria los textos que contenían algún tipo de *cyberbullying*.

Durante la experimentación, existieron varios problemas de escasez de recursos de cómputo, principalmente, insuficiencia de memoria de video, lo que obligaba varias veces a detener el entrenamiento del modelo sin llegar a un resultado satisfactorio. Este problema pudo ser resuelto después de varios periodos de optimización de código para que consumiera menos memoria de video, como la implementación de *data loaders*, que permiten entrenar el modelo con cantidades más pequeñas de datos, además de que permite que la selección de los datos sea aleatoria si es que así se desea, lo que también podría ser beneficiosa para el modelo.

Además de que se desarrollaron varios modelos con diferentes bases de BERT, los cuales no tuvieron resultados satisfactorios, posiblemente las causas principales fueron que eran modelos que sí diferenciaban entre las letras mayúsculas y las minúsculas, por

lo que la cantidad de *tokens* diferentes disponibles para el entrenamiento podía variar.

Para lograr el modelo considerado como satisfactorio, se requirieron de 25 épocas, logrando una pérdida en el entrenamiento de 0.016057654376488536 con una precisión de 0.99%. La clasificación de texto en la detección de 14 tipos diferentes de cyberbullying en redes sociales. La arquitectura del modelo se basa en una versión multilingüe de BERT que no distingue entre mayúsculas y minúsculas (bert-base-multilingual-uncased), la cual tuvo que ser ajustada para la correcta clasificación de los siguientes tipos de cyberbullying, que se muestran en la tabla 1.

1. *racial_target:* Ataques dirigidos hacia un individuo o grupo por su raza o etnia.

2. *other_target:* Agresiones enfocadas en características personales específicas no contempladas en otras categorías.

3. *implicit:* Comentarios o insinuaciones que, sin ser directos, promueven la discriminación. Por ejemplo, el doble sentido.

14. *others:* Cualquier forma de *cyberbullying* que no encaje en las categorías anteriores.

Cabe recalcar la diferencia entre los tipos *other_target* y *others: other_target* se refiere a ataques dirigidos hacia una característica específica de la persona que no encaja en las categorías tradicionales; por ejemplo, burlas sobre su forma de vestir, preferencias musicales o estilo de vida.

Tabla 1. Resultados del rendimiento del modelo.

	Precisión	Recall	F1-Score	Support
racial_target	1.00	1.00	1.00	216
other_target	0.00	0.00	0.00	1
implicit	1.00	0.97	0.99	119
stereotype	1.00	0.99	1.00	163
xenophobia	0.00	0.00	0.00	2
suffering	0.93	1.00	0.97	14
economic	1.00	0.86	0.92	7
migration	1.00	1.00	1.00	70
culture	1.00	0.94	0.97	33
benefits	1.00	1.00	1.00	36
health	1.00	0.50	0.67	2
security	1.00	1.00	1.00	48
dehumanization	1.00	0.25	0.40	12
others	0.93	1.00	0.97	14

Tabla 2. Métricas promediadas.

Métrica	Precisión	Recall	F1-Score	Support
micro avg	1.00	0.97	0.98	737
macro avg	0.85	0.75	0.78	737
weighted avg	0.99	0.97	0.98	737
samples avg	0.30	0.29	0.29	737

Estas métricas son usualmente utilizadas en clasificación para evaluar el rendimiento del modelo en tareas *multi-label*. A continuación, se explican lo que representan cada una:

• *Micro average (micro avg):* Calcula la métrica (precisión, recall, f1-score) considerando todas las instancias del conjunto, es decir, sumando los valores de las clases y calculando la métrica de manera global.

• *Macro average (macro avg):* Calcula la métrica para cada clase individualmente y luego obtiene el promedio.

• *Weighted average (weighted avg):* Similar al macro *average*, pero pondera cada clase según el número de instancias.

• *Samples average (simples avg):* Es específico para problemas de clasificación multietiqueta. Calcula la métrica por instancia en lugar de por clase, y después, promedia sobre todas las instancias.

2.5. Conclusiones

El desarrollo de este proyecto permitió la implementación de un modelo de aprendizaje automático que es capaz de identificar entre una variedad de tipos de *cyberbullying*, con un nivel de precisión aceptable. Sin embargo, quedaron algunas posibilidades de desarrollo que podrían explorarse para futuros desarrollos. La principal área de mejora es la ampliación del *dataset*. Incluir más datos de diferentes fuentes puede mejorar la capacidad de clasificar más datos y ajustarse a casos que pueden presentarse con más frecuencia en la vida real. También se podría evaluar la implementación de técnicas de data *augmentation* para aumentar la diversidad de datos sin la necesidad de recopilar datos nuevos. En la figura 5 se observa la grafica de resultados obtenidos para discriminación a migrantes, que actualmente tiene un alto contenido de importancia social.

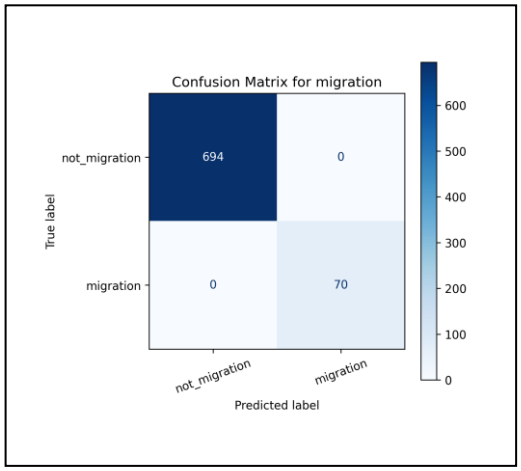

Figura 5. Grafica de resultados de discriminación a migrantes.

2.6. Referencias

[1]Christian Hugo Hoffmann, Is AI intelligent? An assessment on artificial intelligence, 70 years after Turing, Technology in Society, Volume 68, February 2022, 101893.

[2] Devadas Menon , K. Shilp, "Hey, Alexa" "Hey, Siri", "OK Google" …." exploring teenagers' interaction with artificial intelligence (AI)-enabled voice assistants during the COVID-19 pandemic, International Journal of Child-Computer Interaction, Volume 38, December 2023, 100622.

[3]Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. Natural Language Processing: An Introduction. Journal of the American Medical Informatics Association, 18(5), 2011, 544-551.

[4]Mike Kuniavsky, Chapter 1 - Introduction: The middle of Moore's law, Smart Things, 2010, Pages 3-11.

[5]Bengio, A Neural Probabilistic Language Model, Journal of Machine Learning Research 3, 2003, 1137–1155.

[6]Online Etymology Dictionary. (25 de Octubre de 2022). bully (n.). Obtenido de Online Etymology Dictionary: https://www.etymonline.com/word/bully#etymonline_v_45973

[7]Mark Steedman, Chapter 8 - Natural Language Processing, Artificial Intelligence, Handbook of Perception and Cognition, 1996, Pages 229-266.

[8]Yihui Ma, Construction and Data Analysis of a New Media Content Popularity Prediction Model Based on Naive Bayes Algorithm, Procedia Computer Science, Volume 261, 2025, Pages 294-302.

[9] Luca Bergamin, Fabio Aiolli, An investigation into creating counterfactual examples for non-linear Support Vector Machine, Neurocomputing, Volume 651, 28 October 2025, 130809.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, Association for Computational Linguistics, 4171–4186.

[11]E. Dumić, Learning neural network design with tensorflow and keras, ICERI2024 Proceedings, 2024, 10689-10696.

[12] J. Wang, K.Fu, C.T. Lu "SOSNet: A Graph Convolutional Network Approach to Fine Grained Cyberbullyng Detection¨, Proceedings of the 2020 IEEE International Conference on Big Data (IEEE BigData 2020), December 10-13, 2020.