



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

**GRADO EN INGENIERÍA EN TECNOLOGÍAS Y SERVICIOS
DE TELECOMUNICACIÓN**

ÁREA DE INGENIERÍA TELEMÁTICA

**ANÁLISIS DE MENSAJES DE APLICACIONES DE MENSAJERÍA
INSTANTÁNEA PARA LA DETECCIÓN DE ACOSO ESCOLAR**

Dña. MORÁN LEVAS, Olaya
TUTOR: D. CORCOBA MAGAÑA, Víctor
COTUTOR: D. MATA ALFONSO, José

FECHA: Julio 2020

Índice general

1. Hipótesis de partida y alcance	1
1.1. ¿Qué es el ciberacoso?	1
1.2. Efectos del acoso tradicional y del ciberacoso	2
1.3. Contramedidas de las redes sociales frente al ciberacoso	4
2. Estado del arte	5
2.1. Detección automática del ciberacoso	5
2.1.1. Detección basada en contenido	6
2.1.2. Detección basada en el usuario	6
2.1.3. Detección basada en la red	7
2.2. Inconvenientes de la detección basada en características	8
3. Objetivos	9
4. Planificación temporal del desarrollo efectuado	10
5. Metodología de trabajo	12
5.1. Conjunto de datos	12
5.1.1. Pasos de preprocesamiento efectuados	13
5.1.2. Generación de datos sintéticos	15

5.2.	Algoritmos para la Detección de Ciberacoso	16
5.2.1.	Word Embeddings	16
5.2.2.	SNN - Red Neuronal Simple	17
5.2.3.	RNN - Red Neuronal Recurrente	18
5.3.	Métricas de rendimiento para problemas de clasificación en Machine Learning	19
6.	Trabajo realizado y resultados obtenidos	21
6.1.	Evaluación de los modelos de clasificación	21
6.2.	Características del PC utilizado en las pruebas	23
6.3.	Evaluación de diferentes configuraciones de redes neuronales	24
6.3.1.	Evaluación de la Red Neuronal Simple (SNN)	24
6.3.2.	Evaluación de la Red Neuronal Recurrente (RNN)	27
6.4.	Discusión de los resultados	28
7.	Conclusiones y trabajos futuros	29
8.	Presupuesto	30
8.1.	Recursos Humanos	30
8.2.	Equipamiento Hardware	30
8.3.	Equipamiento Software	31
8.4.	Presupuesto Final	31
	Bibliografía	33

Índice de figuras

4.1. Diagrama de Gantt del proyecto	11
5.1. Diagrama de flujo pasos de preprocesamiento	14
5.2. Palabras semejantes a 'smart'	16
5.3. Visualización de la Red Neuronal Simple	17
6.1. Técnica validación cruzada (Fuente: Towards Data Science [28])	21
6.2. Underfitting y overfitting (Fuente: Aprende Machine Learning [29])	22

Índice de tablas

2.1. Estudios de detección de ciberacoso basados en diferentes características	7
5.1. Instancia de datos tras la eliminación de campos innecesarios.	12
5.2. Estructura matriz de confusión	19
6.1. Especificaciones equipo de pruebas	23
6.2. Evaluación SNN de 3 capas <i>Dense</i> con separación de muestras	24
6.3. Evaluación SNN de 3 capas <i>Dense</i> con validación cruzada	24
6.4. Evaluación SNN de 4 capas <i>Dense</i> con separación de muestras	25
6.5. Evaluación SNN de 4 capas <i>Dense</i> con validación cruzada	25
6.6. Evaluación SNN de 5 capas <i>Dense</i> con separación de muestras	26
6.7. Evaluación SNN de 5 capas <i>Dense</i> con validación cruzada	26
6.8. Evaluación RNN con separación de muestras	27
6.9. Evaluación RNN con con validación cruzada	27
6.10. Comparativa de los experimentos más favorables	28
8.1. Costes de recursos humanos	30
8.2. Costes de equipamiento hardware	30
8.3. Costes de equipamiento software	31
8.4. Costes totales	31



1. Hipótesis de partida y alcance

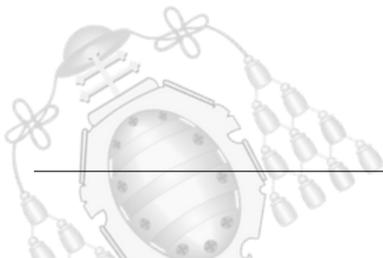
Debido a la proliferación de amistades y relaciones online, las comunicaciones han alcanzado un nivel completamente nuevo. Como resultado de este escenario, existe una creciente evidencia de que las aplicaciones de mensajería instantánea se utilizan con frecuencia para el acoso escolar.

1.1.- ¿Qué es el ciberacoso?

El ciberacoso es el acoso que tiene lugar a través de dispositivos digitales. Puede ocurrir por medio de servicios de mensajes cortos (SMS), mensajes de texto y aplicaciones o en redes sociales, foros o juegos donde las personas pueden ver, participar o compartir contenido.

El acoso cibernético incluye el envío, publicación o intercambio de contenido negativo, dañino o falso sobre un individuo. Puede incluir compartir información personal o privada sobre otra persona, causando vergüenza o humillación. Existen casos de ciberacoso que se han convertido en un comportamiento ilegal o criminal. Por ello, es uno de los problemas más ampliamente reconocidos por individuos y comunidades. Algunas de las definiciones más citadas de ciberacoso son:

- “Un acto agresivo e intencional, llevado a cabo por un grupo o un individuo, utilizando formas electrónicas de contacto, repetidamente o a lo largo del tiempo, contra una víctima que no puede defenderse fácilmente” [1].
- “El uso de tecnologías de información y comunicación por parte de un individuo o grupo para cometer actos deliberados, repetidos y hostiles para lastimar intencionalmente a otros” [2].





Los lugares más comunes donde se produce el ciberacoso son las redes sociales como Facebook®, Instagram®, Snapchat® y Twitter®; SMS, mensajería instantánea, servicios de proveedores de correo electrónico, aplicaciones y funciones de mensajería de redes sociales.

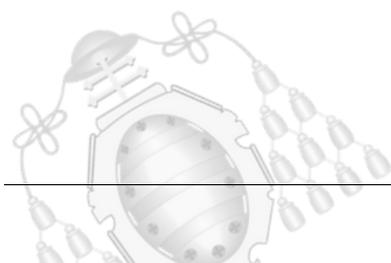
1.2.- Efectos del acoso tradicional y del ciberacoso

El acoso afecta a todos los involucrados: los que son intimidados, los que intimidan y los que son testigos de la intimidación. Un estudio [3] donde se examinó a un total de 7000 estudiantes, concluyó que el acoso escolar contribuye a niveles más altos de soledad y niveles más bajos de bienestar social. Por ello, psicólogos reconocen el *bullying* como un fenómeno estrechamente relacionado con el bienestar de los individuos.

El acoso está vinculado a muchos resultados negativos, incluidos los impactos en la salud mental [4], el uso de sustancias [5] y el suicidio [6]. El *bullying* es más común entre menores, los que son acosados pueden experimentar problemas negativos de salud física, escolar y mental. Estos son más propensos a experimentar:

- Depresión y ansiedad, mayores sentimientos de tristeza y soledad, cambios en los patrones de sueño y alimentación y pérdida de interés en las actividades que solían disfrutar. Estos problemas pueden persistir hasta la edad adulta [7].
- Disminución del rendimiento académico y participación escolar. Es más probable que falten o abandonen la escuela [8].
- Todos estos problemas podrían incluso conducir a tendencias suicidas [6].

Según se relata en “Educación y Futuro” [9], en el caso del *bullying* tradicional, las actuaciones del agresor tienen por finalidad dañar a la víctima y se observa una situación de desigualdad, de desequilibrio de fuerzas; el alumno expuesto a las acciones negativas tiene dificultad para defenderse y se encuentra indefenso ante los alumnos y alumnas que lo acosan.



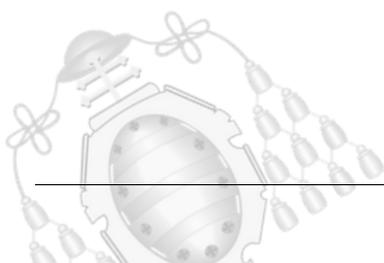


El *cyberbullying*, aunque con ligeras diferencias, sigue compartiendo las características de un acoso escolar tradicional: la agresividad de la conducta, el desequilibrio de poder entre el agresor y la víctima y la continuidad en el tiempo de la agresión. En cambio, las consecuencias del *cyberbullying* son especialmente graves ya que es un fenómeno que afecta, de forma transversal, al resto de espacios de socialización del joven: familia, amigos, compañeros de clase, etc., no existiendo un lugar seguro en el que protegerse de este acoso psicológico continuado.

Por todo ello, se considera el *cyberbullying* como una nueva forma de *bullying*, pero con algunas características específicas, se destacan:

- Anonimato y sensación de impunidad: el acoso no es necesario que sea presencial, por lo que el agresor puede sentirse menos culpable o tener menos empatía por la víctima. El empleo de falsas identidades y la invisibilidad que aporta en muchas ocasiones la red favorece la propagación del fenómeno y la angustia de la víctima.
- En cualquier lugar y momento: Internet está siempre disponible y la movilidad y conectividad de las nuevas tecnologías provoca que se traspasen los límites temporales y físicos que antes se producían en la escuela.
- Rapidez y comodidad: las nuevas tecnologías tienen una capacidad exponencial para propagar cualquier mensaje, reenviar un mensaje a una lista de contactos, grupos, etc.
- El acosador no marginal: en el acoso escolar, los acosadores suelen tener malas relaciones con los profesores y mala imagen social por personalidad conflictiva. En cambio, en el *cyberbullying* los agresores pueden tener una impecable imagen social.

Por lo tanto, el acoso cibernético puede provocar daños sociales y debe frenarse si no se elimina por completo.





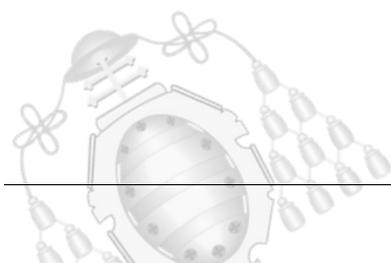
1.3.- Contramedidas de las redes sociales frente al ciberacoso

Las redes sociales brindan cierto grado de apoyo para una experiencia web segura. Las herramientas actuales del mercado, que actúan como salvaguarda, funcionan de la siguiente manera:

- Eliminan el contenido de intimidación y desactivan la cuenta de cualquier persona que intimida o ataca a otra persona, pero solo después de que el destinatario recibe el contenido y lo informa a las autoridades.
- Permite la configuración de privacidad para permitir que personas específicas vean las publicaciones.

Cuando se observa de cerca, todos estos métodos utilizan el filtrado después de que la publicación o mensaje sea leído por el usuario o haya sido publicado en su muro. Existe un retraso entre el momento en que se publica el mensaje y el instante en que las autoridades lo pueden retirar.

En este punto, muchas personas pueden leer la publicación o el mensaje, causando mayor daño al receptor. Esto puede tener un efecto duradero en el usuario, por lo tanto, se necesita un sistema que pueda detectar el ciberacoso antes de que alcance su objetivo y pueda causar cualquier tipo de deterioro, físico o mental.





2. Estado del arte

A pesar de que el acoso escolar siempre ha sido un grave asunto y por ende, ha recibido mucha atención; Internet junto a las redes sociales han hecho que el problema sea aún más crítico y extendido. Esto se debe a que abren las puertas a los acosadores brindando acceso, las veinticuatro horas del día, los siete días de la semana, a víctimas de todas las edades y orígenes, manteniendo sus identidades anónimas [10].

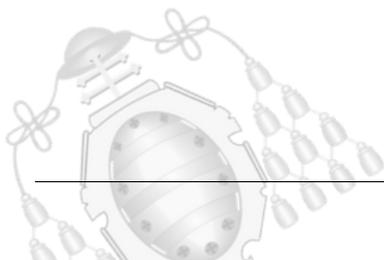
Debido al peligro impuesto por el ciberacoso a las víctimas, este campo de estudio está madurando, con una gran cantidad de investigaciones y hallazgos que evolucionan todos los días. La amplia gama de estudios de ciberacoso existentes abarca campos como la psicología, la lingüística y la informática.

Expertos en el campo del acoso cibernético fueron entrevistados sobre las acciones apropiadas que deben tomarse en respuesta al creciente número de incidentes de ciberacoso; y la mayoría estuvieron a favor de la supervisión automática del mismo [11].

2.1.- Detección automática del ciberacoso

El monitoreo automático del acoso cibernético ha ganado considerable interés en el campo de la informática. La mayoría de investigaciones lo consideraron una tarea de clasificación binaria, donde el texto se clasifica como *bullying* o no *bullying* [12]. Esto se logra mediante la extracción de características del texto y su alimentación a un algoritmo de clasificación.

Múltiples estudios han abordado la detección del acoso cibernético desde perspectivas diferentes, sin embargo, todos se dividen en tres categorías de características: características basadas en contenido, basadas en el usuario y en las redes sociales.





2.1.1.- Detección basada en contenido

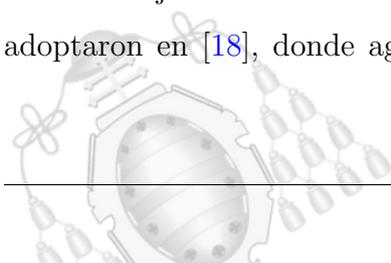
Entre los primeros en abordar el acoso escolar en las redes sociales está [13], donde se creó un framework incorporando la API de Twitter para recopilar tweets y posteriormente clasificarlos de acuerdo a su contenido. El estudio combinó la esencia del análisis de sentimientos y la detección de *bullying*. Como primera fase, los tweets se clasifican como positivos o negativos y luego se clasifican como positivos con contenido de *bullying*, positivo sin contenido de *bullying*, negativo con contenido de *bullying* y negativo sin contenido de *bullying*. Se implementó Naïve Bayes resultando una precisión relativamente alta (70 %).

Un estudio posterior [14] incorporó medidas estadísticas además de extraer características de contenido como: malas palabras y pronombres. Otros investigadores [15], continuaron buscando la detección del acoso cibernético desde una perspectiva basada en el contenido introduciendo nuevas características como: emoticonos y diccionario de jeroglíficos. Su enfoque se probó utilizando varios algoritmos de aprendizaje: Naïve Bayes, SVM y J48. El mejor resultado se registró con SVM logrando una precisión del 81 %.

2.1.2.- Detección basada en el usuario

Muchos investigadores opinan que la información del usuario, como la edad y el número de tweets, pueden indicar la probabilidad de infringir daño a otros. En el estudio [16], se incorporó información del usuario como: el número de tweets o el número de seguidores. Sus características totales, basadas en el usuario y otras, dieron como resultado buenas predicciones con una precisión del 85 %.

Del mismo modo, en [17] agregaron la información de perfil del usuario como característica. Además, asumen que si un usuario intimida en el pasado, es más probable que vuelva a cometer actos de *bullying*. Concluyeron que el efecto de agregar datos del usuario mejora el recall un 5 %. Las características basadas en el usuario también se adoptaron en [18], donde agregaron el género, la edad y la ubicación del usuario al





conjunto de características. La suposición fue que diferentes géneros usan diferentes idiomas y las personas de distintas edades tienen distintos estilos de escritura.

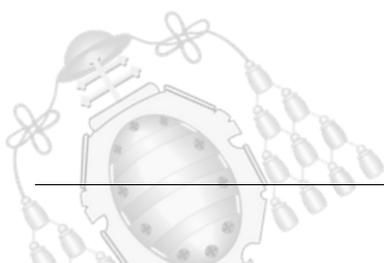
2.1.3.- Detección basada en la red

Una perspectiva interesante para la detección del acoso cibernético estudia la estructura social de los usuarios. Se trata de dibujar la estructura de la red y derivar las características del gráfico. En [19], se centraron en derivar características del gráfico de redes sociales. Características incluidas: número de nodos que indican qué tan grande es la comunidad y número de bordes que indican qué tan bien conectada está la comunidad.

Otra investigación que abordó las características basadas en la red [14], utilizó *Gephi*, una interfaz gráfica para visualizar la conectividad de un usuario basada en las publicaciones de acoso escolar. Posteriormente, investigaron el papel de los participantes en la intimidación, ya sean víctimas o acosadores.

Detección basada en	Estudio	Características
<i>Contenido</i>	[13]	Análisis de sentimientos
	[14]	Análisis de malas palabras y pronombres
	[15]	Análisis de emoticonos y diccionario de jeroglíficos
<i>Usuario</i>	[16]	Análisis de número de tweets y número de seguidores
	[17]	Análisis de la información de perfil de usuario.
	[18]	Análisis del género, la edad y la ubicación del usuario
<i>Red</i>	[19]	Análisis del número de nodos y número de bordes
	[14]	Análisis de la conectividad de un usuario

Tabla 2.1.- Estudios de detección de ciberacoso basados en diferentes características





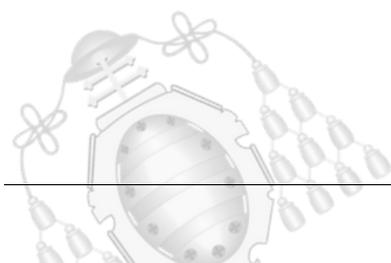
2.2.- Inconvenientes de la detección basada en características

Aunque el estado del arte en la detección del acoso cibernético está evolucionando rápidamente, han surgido diversos inconvenientes. Un problema fundamental aún presente es que la mayoría de las investigaciones intentan mejorar el proceso de detección al sugerir nuevas características. Sin embargo, este enfoque podría generar una gran cantidad de características que requieren fases cuidadosas de extracción y selección que conducen a una sobrecarga computacional.

Otro inconveniente es que fracasan a la hora de adaptarse a la naturaleza cambiante del lenguaje. Las palabras ofensivas, que se consideran características en la mayoría de los enfoques de detección, no son estáticas y cambian con el tiempo.

En conclusión, los enfoques de detección no deben basarse en características estáticas sino en mecanismos más automatizados.

Además, a pesar del éxito de los estudios actuales, se ha abordado poco un problema central: la semántica de las palabras, su significado y sus relaciones.



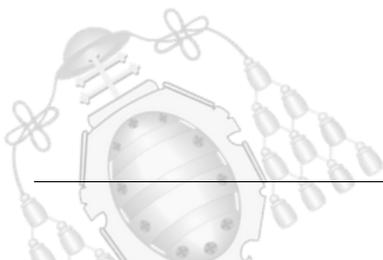


3. Objetivos

En el trabajo realizado, se proponen varios algoritmos de detección de ciberacoso que pretenden solucionar los problemas actuales no resueltos. Estudios anteriores comienzan la detección del ciberacoso con la extracción de características seguida de la selección de las mismas. En el estudio actual se aspira a reducir el flujo de trabajo de detección clásico, realizando detecciones sin características. Para ello, se transforma el texto en *word embeddings* con las que se alimenta el modelo.

El objetivo principal del estudio es desarrollar un enfoque de detección eficiente capaz de abordar la semántica y el significado de las palabras a la vez que se producen resultados precisos, manteniendo el tiempo y el coste computacional al mínimo posible. Además se adaptará el algoritmo propuesto para el contenido en español.

Una vez logrado dicho algoritmo de detección de ciberacoso se implementará en la plataforma de mensajería instantánea *Telegram*, para comprobar la eficacia del mismo con casos reales.





4. Planificación temporal del desarrollo efectuado

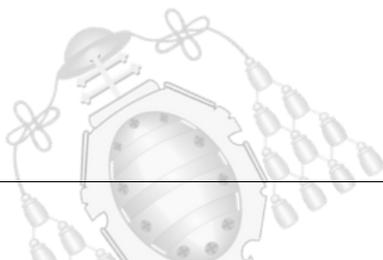
La primera fase del proyecto consiste en la investigación previa, que se divide en dos tareas:

- Investigación teórica: tiene como principal objetivo la obtención de conocimientos en el campo de la supervisión automática del acoso cibernético.
- Investigación aplicada: se centra en encontrar mecanismos o estrategias que permiten lograr los objetivos mencionados en el capítulo 3.

La segunda fase consiste en el desarrollo del estudio. Las tareas realizadas en esta fase son:

- Preparación *dataset*. Los detalles de la tarea se encuentran en la sección 5.1.
- Diseño de las Redes Neuronales. Los detalles de la tarea se encuentran en la sección 5.2.
- Pruebas y mejoras de las Redes Neuronales. Los detalles de la tarea se encuentran en la sección 6.3.
- Pruebas con el algoritmo propuesto en la plataforma de mensajería instantánea *Telegram*.

Además, durante el progreso de la investigación se genera la documentación de la misma, recogiendo las fases previamente descritas.





A continuación, en la figura 4.1 se muestran las fases que componen el estudio desarrollado y la duración temporal de las mismas.

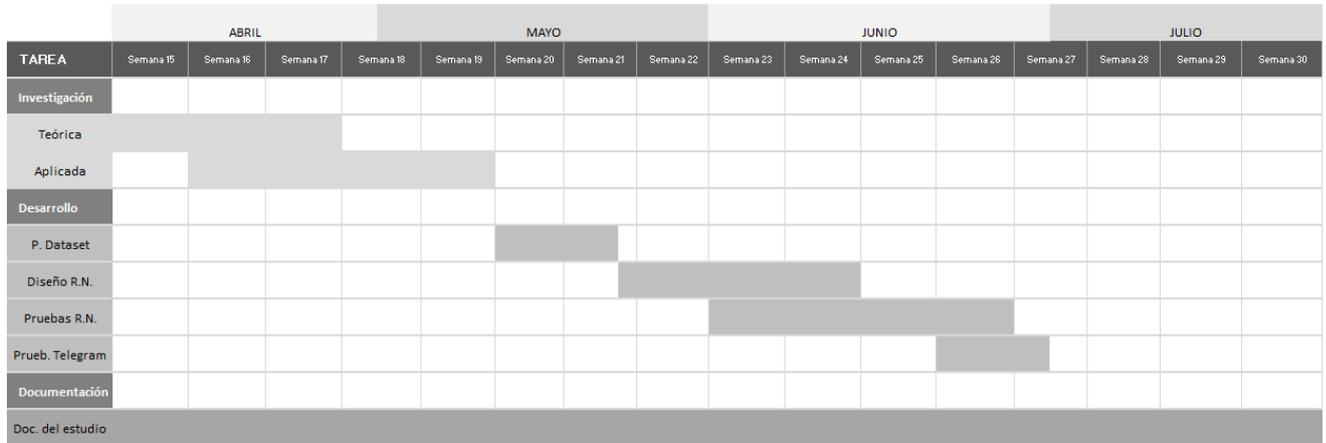
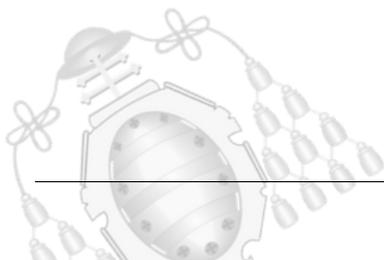


Figura 4.1.- Diagrama de Gantt del proyecto





5. Metodología de trabajo

5.1.- Conjunto de datos

Se utiliza el conjunto de datos de Formspring [20] para entrenar el modelo. Los parámetros originales del *dataset* incluyen los siguientes campos:

User Id	Post	Question	Answer	Asker	Answer 1-3	Severity 1-3	Bullying 1-3
---------	------	----------	--------	-------	------------	--------------	--------------

A continuación se muestra un ejemplo de una instancia de datos real proporcionada en el *dataset*:

“aguitarplayer94 Q: whats your favorite song? :D A: I like too many songs to have a favorite whats your favorite song? :D I like too many songs to have a favorite None No 0 n/a No 0 n/a No 0 n/a”

Este formato no es adecuado para entrenar el modelo ya que existen valores “n/a” en la instancia de datos y las etiquetas están en medio del texto. Por ello, se limpian las mismas y se convierten a un formato tabular con los siguientes campos:

- SpanishData: conversación de texto real.
- BullyingLabel: 0 o 1 para clasificar los datos como no acoso o acoso respectivamente.

SpanishData	BullyingLabel
favorito canción me gusta muchos canciones favorito	0

Tabla 5.1.- Instancia de datos tras la eliminación de campos innecesarios.

La tabla 5.1 muestra el ejemplo mencionado anteriormente, tras la eliminación de los campos innecesarios y la realización de los pasos de preprocesamiento en los datos.



5.1.1.- Pasos de preprocesamiento efectuados

1) Conversión de mayúsculas y minúsculas.

Conversión de todos los mensajes a minúsculas para que “Cómo” y “cómo” se perciban como la misma palabra. Este paso facilita que no se encuentren funciones duplicadas, ya que sería redundante.

2) Eliminación de palabras de detención.

Se utilizan, y se eliminan del *dataset*, las palabras de detención predefinidas proporcionadas por el paquete *nlTK* [21], que cuenta con un total de 183 palabras.

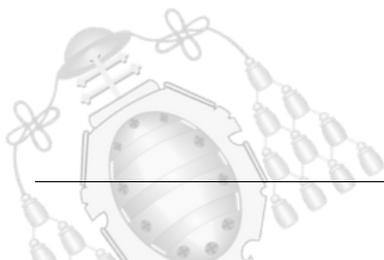
3) Eliminación de signos de puntuación.

Los signos de puntuación no son significativos para el modelo, ya que el acoso cibernético no depende de un signo de interrogación o de exclamación. Además, es una práctica extendida la utilización de múltiples puntuaciones en un mensaje de chat como “.....” o “?????”. Por lo que, al ser irrelevante para el modelo, tales repeticiones se han suprimido.

Otro aspecto importante de la comunicación mediante mensajes de texto es el uso de emoticonos como “:)” o “:(”. Al eliminar los signos de puntuación, también se suprimen estos últimos.

4) Traducción del conjunto de datos

Se utiliza la biblioteca API Translation [22] de Google Cloud Platform para la traducción del *dataset*, con esta solución se traduce el campo “*Data*”, el cual contiene los mensajes del conjunto de datos, del idioma inglés (en) al español (es). Se escoge esta opción debido a la limitación de solicitudes encontrada en las demás APIs de traducción.



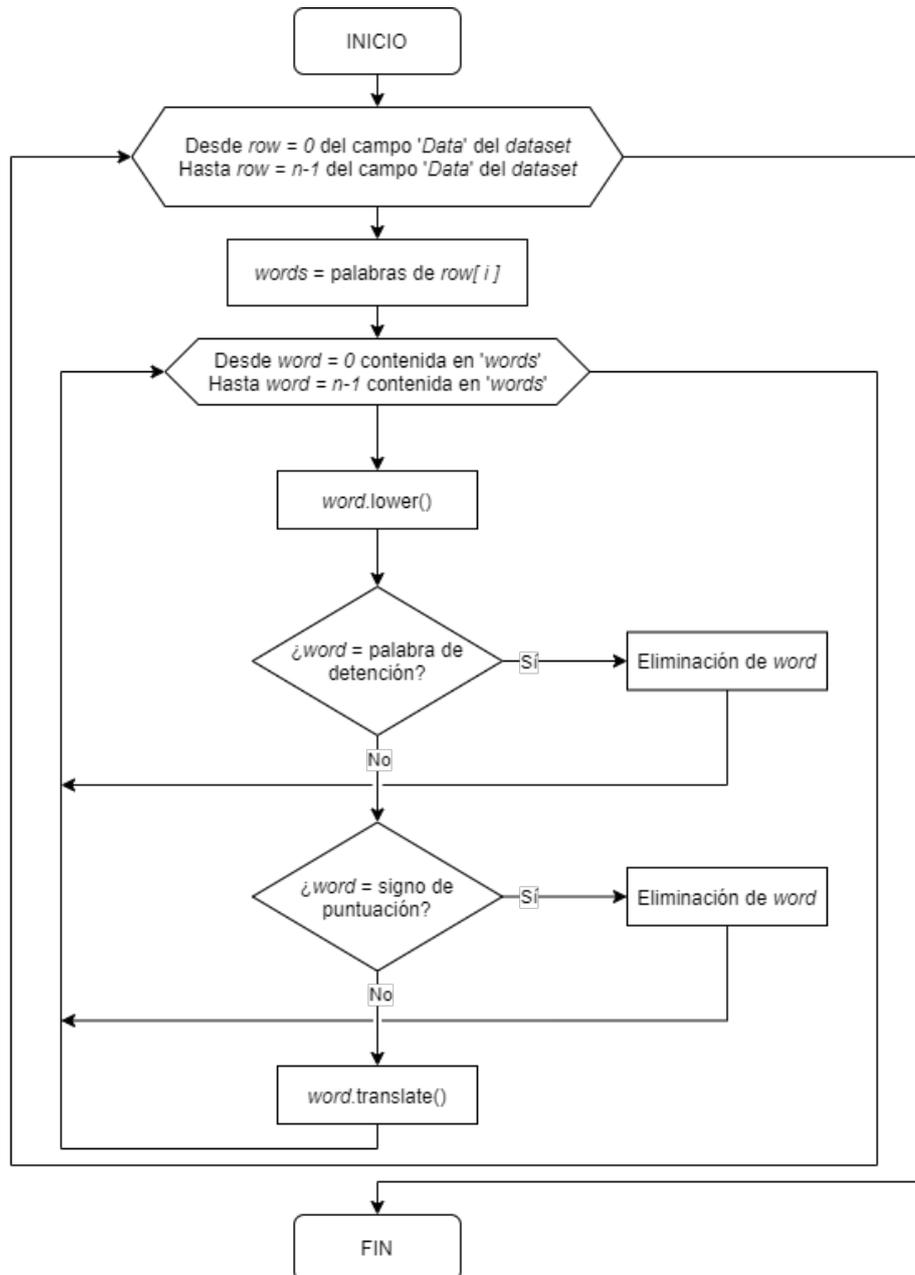
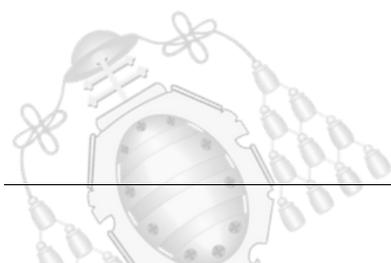


Figura 5.1.- Diagrama de flujo pasos de preprocesamiento

La figura 5.1 muestra el diagrama de flujo de los pasos de preprocesamiento. Tras la realización de este procedimiento, el mensaje “whats your favorite song? :D; I like too many songs to have a favorite” se transforma en “favorito canción me gusta muchos canciones favorito”.





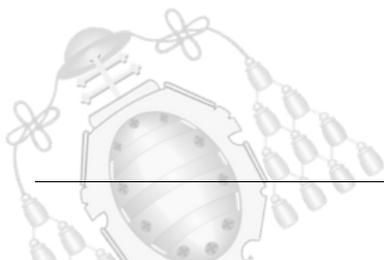
5.1.2.- Generación de datos sintéticos

El dataset de partida contiene un total de 12.158 mensajes, de los cuales sólo 1.958 son mensajes de acoso cibernético.

Para abordar este problema de desequilibrio de clase y mejorar el rendimiento del modelo, se generan artificialmente algunas instancias nuevas. Para ello, se realizan los siguientes pasos:

- Se filtran los 1.958 mensajes de ciberacoso tras el preprocesamiento y se almacenan en una lista.
- Se decide el número adicional de instancias de datos que se desean agregar al conjunto de datos. Se escogen 7.832 para que prácticamente 1/2 del dataset resultante consista en mensajes de intimidación.
- Con el objetivo de lograr el número adicional de instancias de datos se agregan los 1.958 mensajes de acoso cibernético cuadruplicados al conjunto de datos inicial.

El nuevo conjunto de datos contiene un total de 19.990 mensajes, de los cuales 10.200 están libres de acoso y 9.790 poseen acoso cibernético.





5.2.- Algoritmos para la Detección de Ciberacoso

El aspecto más notable de los algoritmos empleados es que eliminan tres fases de clasificación llevadas a cabo anteriormente por otros algoritmos de detección de ciberacoso: determinación de características, extracción y selección.

Esto se logra mediante la generación de *word embeddings* (vectores numéricos) para cada palabra en un mensaje, que sirven para alimentar directamente el modelo. Los detalles del *word embedding* se explican en la siguiente sección.

5.2.1.- Word Embeddings

Word Embeddings son una clase de técnicas utilizadas para generar una representación numérica de texto. Una característica llamativa del word embedding es que genera representaciones similares para palabras semánticamente semejantes.

Esta propiedad permite que una máquina entienda realmente el significado del texto en lugar de tratarlo como cadenas de números aleatorios.

Para ilustrar este gran potencial, la figura 5.2 muestra las palabras afines a la palabra 'inteligente', en inglés, junto a su puntaje de similitud utilizando el *word embedding* proporcionado por *Glove* [23].

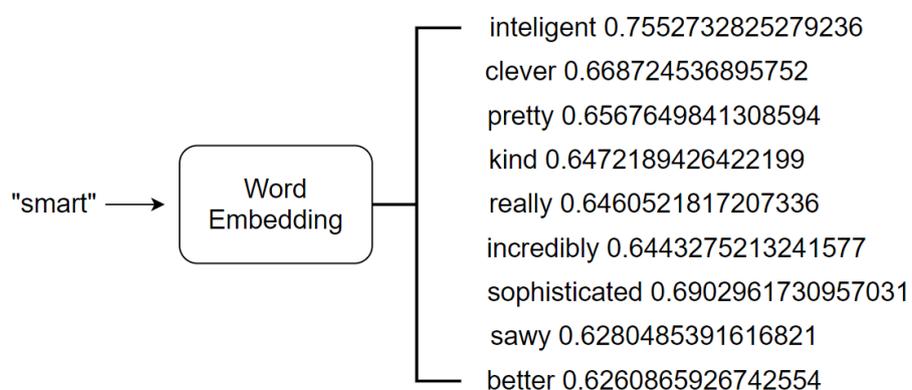
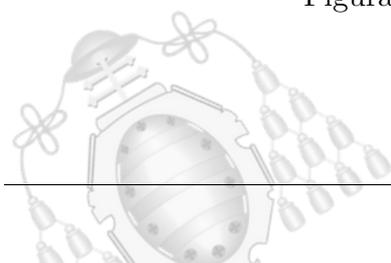


Figura 5.2.- Palabras semejantes a 'smart'





El uso del *word embedding* en el algoritmo propuesto hace que sea más avanzado en comparación con el enfoque tradicional, ya que incorpora la semántica y no solo las características extraídas del texto sin procesar.

Para ello, asignamos cada palabra del vocabulario a un número entero único y codificamos las secuencias de entrada. Realizamos este proceso con la clase *Tokenizer* de la API de *Keras* [24].

5.2.2.- SNN - Red Neuronal Simple

En el caso del modelo de clasificación básico, la red neuronal se define como una secuencia de capas *Dense*.

Las capas *Dense*, encargadas de la clasificación, son un conjunto de capas completamente conectadas [25]. Es decir, cada neurona está conectada a todas las demás neuronas en la siguiente capa, como se observa en la figura 5.3. El número de las mismas en las capas *Dense* varía, sin embargo, la última debe tener 2 neuronas correspondientes al número de clases en este caso (0 = no acoso ; 1 = acoso).

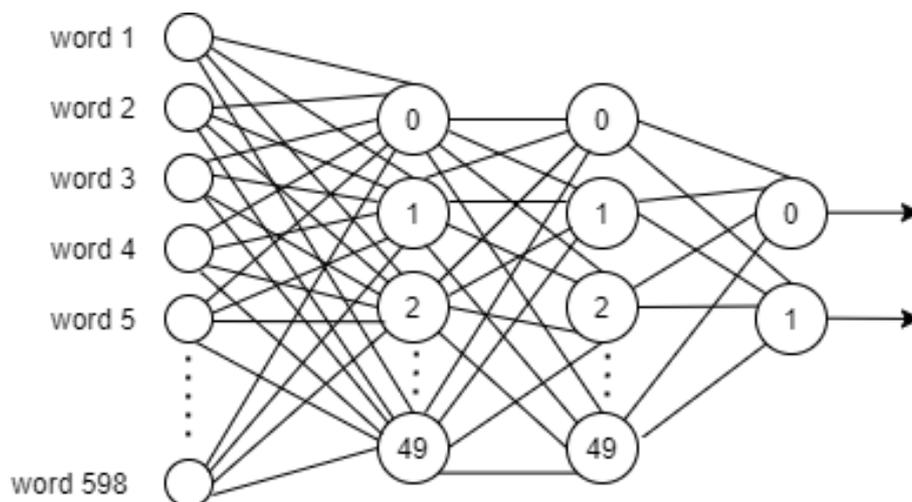
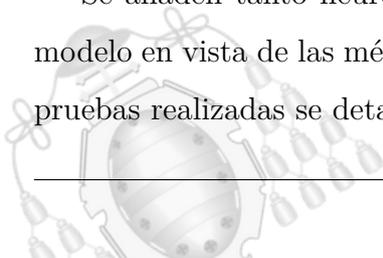


Figura 5.3.- Visualización de la Red Neuronal Simple

Se añaden tanto neuronas como capas *Dense* como sea necesario para mejorar el modelo en vista de las métricas de rendimiento descritas en la sección 5.3. Las diversas pruebas realizadas se detallan en el capítulo 6.





5.2.3.- RNN - Red Neuronal Recurrente

Se implementa una Red Neuronal Recurrente sencilla. Su arquitectura consta de tres tipos de capas: *Embedding*, *Long Short-Term Memory* y *Dense*.

Se adopta la capa *Embedding* proporcionada por *Keras* [26] debido a que recoge todo el texto (mensajes limpios en este caso) y genera un espacio vectorial de vocabulario. Por lo tanto, el cálculo es más sencillo, tanto en tiempo como en recursos. Esta capa requiere establecer tres parámetros de entrada para la construcción del espacio vectorial:

- Input dimension: especifica el número total de palabras en el vocabulario (corpus completo). Este número deriva de lo siguiente, siendo M todos los mensajes del corpus.

$$M = \{m_1, m_2, m_3, m_4 \dots m_n\}, n = \text{número de mensajes}$$

$$\text{Input dimension} = \text{length}(\text{Tokenized}(M))$$

- Output dimension: especifica el tamaño del vector de salida de esta capa.
- Input length: longitud de cada vector (número máximo de palabras por mensaje).

La longitud de entrada se calcula como sigue:

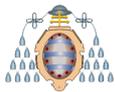
$$\text{Input length} = \max(\text{length for } m \text{ in } M)$$

La siguiente capa es una capa *LSTM* proporcionada por *Keras* [27]. La característica principal de las *LSTM* es que la información puede persistir introduciendo bucles en el diagrama de la red, por lo que pueden “recordar” estados previos y utilizar esta información para decidir cuál será el siguiente.

Debido a que nuestro objetivo es la clasificación binaria, la última capa es una capa *Dense* con una función de activación *sigmoid*.

Se añaden tantos nodos en la capa LSTM como sea necesario para mejorar el modelo en vista de las métricas de rendimiento descritas en la sección 5.3. Las diversas pruebas realizadas se detallan en el capítulo 6.





5.3.- Métricas de rendimiento para problemas de clasificación en Machine Learning

Tras implementar el modelo y obtener los resultados se evalúa la calidad de la predicción mediante las siguientes métricas de rendimiento:

1) Matriz de confusión

Es una de las métricas que más se utiliza para observar la corrección y precisión del modelo. Se emplea para los problemas de clasificación donde la salida puede ser de dos o más tipos de clases. En el trabajo desarrollado se resuelve un problema de clasificación en el cual se predice si un mensaje contiene *bullying* o no.

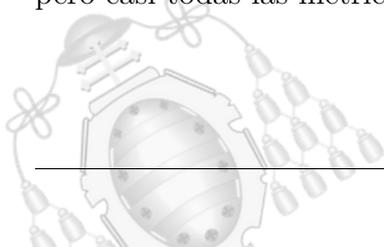
La matriz tiene la estructura que se muestra en la tabla 5.2 :

		Predicción	
		Positivos	Negativos
Real	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Tabla 5.2.- Estructura matriz de confusión

- **VP** es la cantidad de *positivos* que fueron *clasificados correctamente* como positivos por el modelo.
- **VN** es la cantidad de *negativos* que fueron *clasificados correctamente* como negativos por el modelo.
- **FN** es la cantidad de *positivos* que fueron *clasificados incorrectamente* como negativos.
- **FP** es la cantidad de *negativos* que fueron *clasificados incorrectamente* como positivos.

La matriz de confusión en sí misma no es una medida de rendimiento como tal, pero casi todas las métricas de rendimiento se basan en ella.





2) Exactitud (Accuracy)

La precisión o exactitud en los problemas de clasificación es el número de predicciones correctas realizadas por el modelo sobre todo tipo de predicciones realizadas.

$$Accuracy = \frac{VP + VN}{Total} \quad (5.1)$$

3) Precision (Precisión)

La precisión indica qué proporción de mensajes que se clasifican como *bullying* en realidad contenían ciberacoso.

$$Precision = \frac{VP}{VP + FP} \quad (5.2)$$

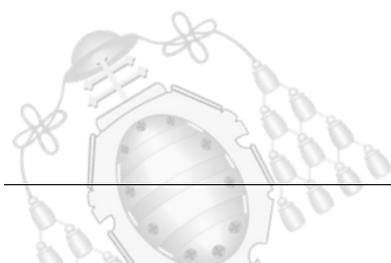
4) Recall or Sensitivity (Sensibilidad)

La sensibilidad indica qué proporción de mensajes que realmente contienen ciberacoso fueron clasificados por el algoritmo como *bullying*.

$$Recall = \frac{VP}{VP + FN} \quad (5.3)$$

5) F1 Score

F1 score considera tanto la precisión como la sensibilidad de la prueba para calcular la puntuación: el número de resultados positivos correctos dividido por el número de todos los resultados positivos devueltos por el clasificador.



6. Trabajo realizado y resultados obtenidos

6.1.- Evaluación de los modelos de clasificación

A continuación evaluamos la calidad de la predicción de los diferentes modelos mediante las métricas de rendimiento mencionadas en la sección 5.3. Para poder utilizar dicho método de evaluación se opta por dos técnicas: separación de muestras y validación cruzada.

1) Separación de muestras: el conjunto de datos se divide en datos de entrenamiento (67%) y datos de prueba (33%). El subconjunto de entrenamiento contiene una salida conocida y el modelo aprende sobre estos datos para ser generalizado a otros más adelante. El subconjunto de datos de prueba se emplea para probar la predicción del modelo en ese subconjunto.

2) Validación cruzada: este método proporciona una estimación sólida del rendimiento de un modelo en datos no vistos. Para ello, se divide el conjunto de datos en k subconjuntos. Se usa $k-1$ subconjuntos para entrenar el modelo y se reserva un subconjunto como datos de prueba. El proceso se repite hasta que todos los subconjuntos han sido el conjunto de validación.

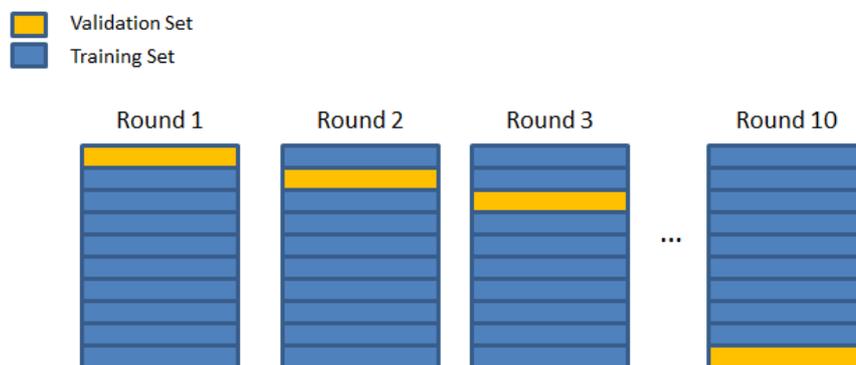
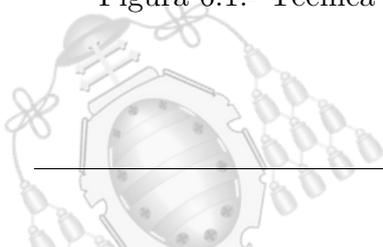


Figura 6.1.- Técnica validación cruzada (Fuente: Towards Data Science [28])





En el proceso de entrenamiento del modelo pueden surgir dos cuestiones desfavorables, ya que afectan a la predicción: el sobre-entrenamiento y el sub-entrenamiento.

1) Sobre-entrenamiento

El sobre-entrenamiento generalmente ocurre cuando el modelo es demasiado complejo y sólo se ajusta a los casos particulares siendo incapaz de reconocer nuevos datos de entrada. En el conjunto de datos de entrada a menudo se introducen muestras atípicas o con “ruido” en alguna de sus dimensiones, o muestras que pueden no ser del todo representativas. Cuando se produce sobre-entrenamiento, el algoritmo considera como válidos sólo los datos idénticos a los del conjunto de datos de entrenamiento, incluidos sus defectos, y es incapaz de distinguir entradas correctas como fiables si se salen de los rangos ya preestablecidos.

2) Sub-entrenamiento

El sub-entrenamiento se refiere a un modelo que no puede modelar los datos de entrenamiento y por lo tanto, no logra generalizar a nuevos datos. Este efecto destruye la precisión del modelo; suele suceder cuando los datos son insuficientes para construir el mismo.

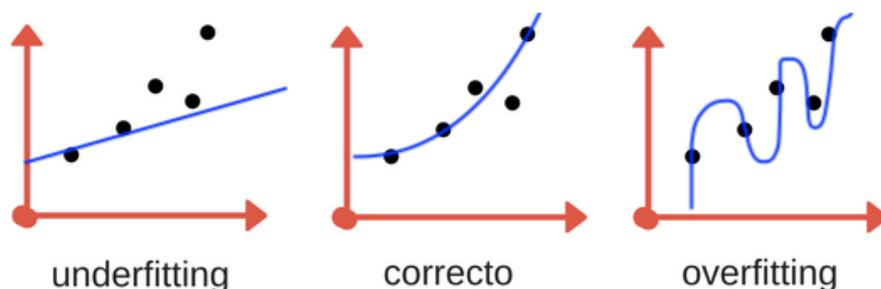
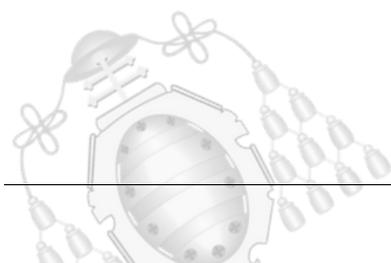


Figura 6.2.- Underfitting y overfitting (Fuente: Aprende Machine Learning [29])





6.2.- Características del PC utilizado en las pruebas

En la tabla 6.1 se muestran las especificaciones del PC donde se han llevado a cabo las distintas pruebas del estudio realizado.

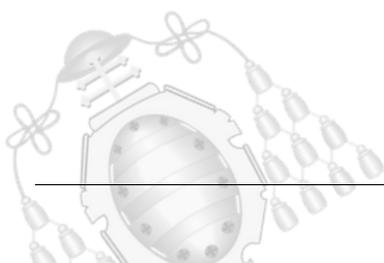
Especificaciones Dell Inspiron 5590

Procesador	Intel(R) Core(TM) i7-10510U CPU 1.80GHz 2.30GHz
RAM	8,00GB (7,79GB usable)
GPU	NVIDIA GeForce MX250
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64

Tabla 6.1.- Especificaciones equipo de pruebas

En inteligencia artificial no siempre se escoge el mejor algoritmo pues también hay que tener en cuenta el tiempo. El problema suele radicar en el tiempo de generación del modelo, por ello, muchos fabricantes están añadiendo procesadores especializados para estas tareas: las NPUs.

Una NPU o *Neural Processing Unit* es un circuito que comprende todos los componentes de control y lógica aritmética necesarios para ejecutar algoritmos de aprendizaje automático. Las NPU están diseñadas para acelerar el rendimiento de las tareas comunes de aprendizaje automático, como la clasificación de imágenes, la traducción automática, la detección de objetos y otros modelos predictivos.





6.3.- Evaluación de diferentes configuraciones de redes neuronales

6.3.1.- Evaluación de la Red Neuronal Simple (SNN)

En el primer experimento, la red neuronal se define como una secuencia de tres capas *Dense*, donde se añaden neuronas para mejorar el modelo. En la tabla 6.2 se muestran los resultados obtenidos en las distintas pruebas evaluadas mediante separación de muestras mientras que en la tabla 6.3 se muestran los resultados obtenidos en las distintas pruebas evaluadas mediante validación cruzada.

La primera columna de las tablas mencionadas contiene el número de neuronas por capa, se emplea el separador “/” para indicar el número de neuronas pertenecientes a cada capa que compone el modelo. Las siguientes columnas corresponden a las métricas de rendimiento descritas en la sección 5.3. Por último, las columnas restantes contienen el tiempo de generación y clasificación del modelo (en segundos) y el porcentaje de uso de memoria.

N neuronas	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso memoria (%)
25/25/2	41	39	27	32	101.731	0.057	12.20
50/50/2	48	49	57	53	111.207	0.062	12.36
75/75/2	57	56	63	60	120.349	0.060	12.43
100/100/2	57	55	90	68	138.853	0.069	12.43
125/125/2	74	78	68	73	154.745	0.064	12.44
150/150/2	64	85	35	50	173.490	0.071	12.45
175/175/2	58	64	37	47	186.038	0.075	12.49

Tabla 6.2.- Evaluación SNN de 3 capas *Dense* con separación de muestras

N neuronas	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso memoria (%)
25/25/2	49.74	49.23	57.42	52.60	102.359	0.100	11.62
50/50/2	48.49	47.60	53.06	49.86	77.0219	0.049	12.79
75/75/2	36.67	21.78	15.20	16.20	56.046	0.044	13.46
100/100/2	53.45	41.22	58.51	47.85	63.130	0.045	13.46
125/125/2	56.79	49.90	69.22	56.35	71.419	0.049	13.47
150/150/2	42.22	22.74	37.05	26.73	81.301	0.049	13.47
175/175/2	49.79	29.38	60.00	39.45	98.896	0.059	13.47

Tabla 6.3.- Evaluación SNN de 3 capas *Dense* con validación cruzada



En el segundo experimento, la red neuronal se define como una secuencia de cuatro capas *Dense*, donde se añaden neuronas para mejorar el modelo. En la tabla 6.4 se muestran los resultados obtenidos en las distintas pruebas evaluadas mediante separación de muestras mientras que en la tabla 6.5 se muestran los resultados obtenidos en las distintas pruebas evaluadas mediante validación cruzada.

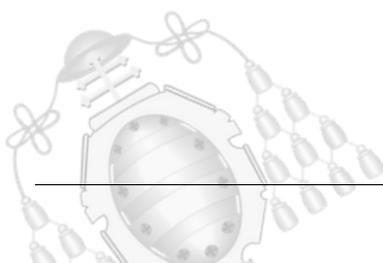
La primera columna de las tablas mencionadas contiene el número de neuronas por capa, se emplea el separador “/” para indicar el número de neuronas pertenecientes a cada capa que compone el modelo. Las siguientes columnas corresponden a las métricas de rendimiento descritas en la sección 5.3: exactitud, precisión, sensibilidad y F1 score. Por último, las columnas restantes contienen el tiempo de generación y clasificación del modelo (en segundos) y el porcentaje de uso de memoria.

N neuronas	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso memoria (%)
25/25/25/2	24	32	43	37	115.869	0.058	12.15
50/50/50/2	77	78	75	77	133.408	0.065	12.20
75/75/75/2	68	90	41	57	136.377	0.071	12.36
100/100/100/2	57	58	54	56	160.372	0.074	12.43
125/125/125/2	21	26	30	28	186.010	0.080	12.49

Tabla 6.4.- Evaluación SNN de 4 capas *Dense* con separación de muestras

N neuronas	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso memoria (%)
25/25/25/2	45.28	28.53	40.07	33.12	52.354	0.047	13.41
50/50/50/2	24.55	4.91	3.16	3.84	56.556	0.051	13.42
75/75/75/2	44.97	33.46	41.60	36.61	60.174	0.049	13.42
100/100/100/2	49.36	30.72	42.93	35.46	75.027	0.056	13.43
125/125/125/2	49.63	34.56	45.69	39.13	85.235	0.058	13.48

Tabla 6.5.- Evaluación SNN de 4 capas *Dense* con validación cruzada





En el tercer experimento, la red neuronal se define como una secuencia de cinco capas *Dense*, donde se añaden neuronas para mejorar el modelo. En la tabla 6.6 se muestran los resultados obtenidos en las distintas pruebas evaluadas mediante separación de muestras mientras que en la tabla 6.7 se muestran los resultados obtenidos en las distintas pruebas evaluadas mediante validación cruzada.

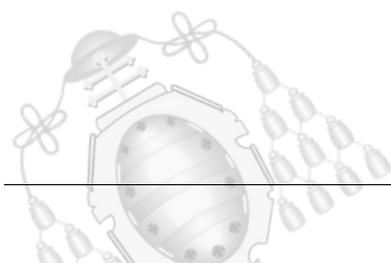
La primera columna de las tablas mencionadas contiene el número de neuronas por capa, se emplea el separador “/” para indicar el número de neuronas pertenecientes a cada capa que compone el modelo. Las siguientes columnas corresponden a las métricas de rendimiento descritas en la sección 5.3: exactitud, precisión, sensibilidad y F1 score. Por último, las columnas restantes contienen el tiempo de generación y clasificación del modelo (en segundos) y el porcentaje de uso de memoria.

N neuronas	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso memoria (%)
25/25/25/25/2	77	97	56	71	124.026	0.064	12.15
50/50/50/50/2	22	26	30	28	127,177	0.069	12.20
75/75/75/75/2	23	28	33	30	137.402	0.083	12.36
100/100/100/100/2	21	26	30	28	166.291	0.078	12.36

Tabla 6.6.- Evaluación SNN de 5 capas *Dense* con separación de muestras

N neuronas	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso memoria (%)
25/25/25/25/2	65.28	54.82	77.31	64.05	56.718	0.051	13.51
50/50/50/50/2	50.42	29.49	37.76	32.96	61.107	0.054	13.52
75/75/75/75/2	59.84	52.70	78.92	62.62	67.258	0.049	13.53
100/100/100/100/2	34.24	16.68	21.29	18.35	82.616	0.059	13.53

Tabla 6.7.- Evaluación SNN de 5 capas *Dense* con validación cruzada





6.3.2.- Evaluación de la Red Neuronal Recurrente (RNN)

En el cuarto experimento, la red neuronal se define con una capa *Embedding*, una capa *LSTM* y una capa *Dense*, donde se añaden nodos en la capa *LSTM* para mejorar el modelo. En la tabla 6.8 se muestran los resultados obtenidos en las distintas pruebas evaluadas mediante separación de muestras mientras que en la tabla 6.9 se muestran los resultados obtenidos en las distintas pruebas evaluadas mediante validación cruzada.

La primera columna de las tablas mencionadas contiene el número de nodos y neuronas por capa, se emplea el separador “/” para indicar el número de nodos en la capa *LSTM* y neuronas en la capa *Dense*. La capa *Embedding* no aparece representada numéricamente en esta notación debido a que no está compuesta por neuronas o nodos, por lo tanto figura como “-”.

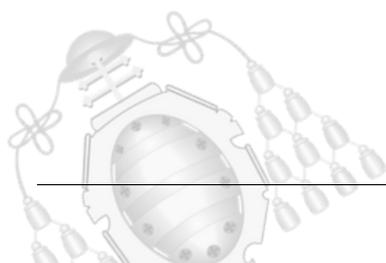
Las siguientes columnas corresponden a las métricas de rendimiento descritas en la sección 5.3: exactitud, precisión, sensibilidad y F1 score. Por último, las columnas restantes contienen el tiempo de generación y clasificación del modelo (en segundos) y el porcentaje de uso de memoria.

N nodos/neur	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso memoria (%)
-/25/1	92	98	86	92	407.496	7.081	15.02
-/50/1	92	98	87	92	655.764	12.108	15.03
-/100/1	93	99	87	92	1774.367	19.998	12.84
-/150/1	92	98	85	91	2519.677	26.863	12.84

Tabla 6.8.- Evaluación RNN con separación de muestras

N nodos/neur	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso memoria (%)
-/25/1	90.45	84.37	98.86	91.03	996.333	4.650	16,.32
-/50/1	87.80	80.93	98.57	88.84	1123.232	9.823	16.34
-/100/1	88.13	81.25	98.50	89.04	1627.286	16.800	16.34
-/150/1	89.68	83.69	98.47	90.40	1848.022	24.991	16.36

Tabla 6.9.- Evaluación RNN con con validación cruzada





6.4.- Discusión de los resultados

Las pruebas realizadas en los distintos modelos muestran que el aumento de neuronas o nodos en las capas no necesariamente equivale a un mejor rendimiento de los mismos.

Esto se debe a que se provoca un sobre-entrenamiento, resultado de un modelo excesivamente complicado, como se detalla al principio del capítulo. Para evitar este efecto se ajustan los múltiples modelos y se utiliza la validación cruzada para comparar las precisiones predictivas en los datos de prueba.

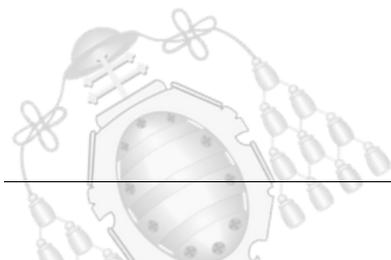
A continuación, en la tabla 6.10, se muestran los resultados más favorables de los distintos experimentos con validación cruzada.

Algoritmo	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)	T. gen. (s)	T. clasif. (s)	Uso mem. (%)
SNN(3)	56.79	49.90	69.22	56.35	71.419	0.049	13.47
SNN(4)	49.63	34.56	45.69	39.13	85.235	0.058	13.48
SNN(5)	65.28	54.82	77.31	64.05	56.718	0.051	13.51
RNN	90.45	84.37	98.86	91.03	1123.232	4.650	16.32

Tabla 6.10.- Comparativa de los experimentos más favorables

En vista de los resultados obtenidos, se opta por la Red Neuronal Recurrente (RNN) configurada con 25 nodos en la capa *LSTM*. Esto se debe a que los experimentos realizados muestran que RNN es el modelo más favorable, con una exactitud en la mejor prueba del 90.45 %. El rendimiento del modelo RNN incrementa durante las épocas, debido a que el aprendizaje evoluciona con cada una de ellas. El modelo comienza con una precisión del 65 %, y aumenta al 90 % después de 10 épocas.

Cuando se compara RNN con el resto de enfoques (SNN(x)), genera mejores resultados en las cuatro métricas de rendimiento: exactitud, precisión, sensibilidad y F1 score. Entre los cuatro modelos de clasificación estudiados, el RNN muestra la diferencia más notable.





7. Conclusiones y trabajos futuros

El ciberacoso es uno de los delitos más perjudiciales de Internet, sus consecuencias son críticas para las víctimas: desde el suicidio hasta la disminución de la autoestima. Por ello, el control del acoso cibernético ha sido el foco de muchas investigaciones psicológicas y técnicas.

En el estudio desarrollado, se ha abordado el tema de la detección automática del acoso cibernético a través de plataformas de mensajería instantánea. El objetivo principal de la investigación fue desarrollar un algoritmo de detección de ciberacoso eficiente capaz de abordar la semántica y el significado de las palabras. Posteriormente, se implementa en la plataforma de mensajería instantánea *Telegram* para comprobar la eficacia del mismo. Para ello, se ha considerado eliminar la extracción de características del proceso de detección y automatizar el mismo con una RNN. El algoritmo propuesto ocasiona que la detección del acoso cibernético sea una tarea totalmente automatizada, sin intervención humana, a la vez que garantiza un buen resultado. Las exhaustivas pruebas demostraron la viabilidad del enfoque.

Este modelo simple genera a una exactitud del 90% para un conjunto de datos pequeño. Por lo que, como trabajo futuro se plantea:

- Aumentar el conjunto de datos
- Mejorar la etapa de preprocesamiento
- Investigar nuevos algoritmos de detección de ciberacoso más complejos (como pueden ser las Redes Neuronales Convolucionales (CNN))

Además, en un futuro se pretende implementar un sistema distribuido de detección de ciberacoso que utilice el algoritmo de aprendizaje automático para detectar mensajes de *bullying* y descartarlos antes de que alcancen su objetivo y que puedan causar cualquier tipo de daño, físico o mental, al usuario.



8. Presupuesto

8.1.- Recursos Humanos

Para la realización del estudio se ha requerido la participación tanto de un investigador como de un programador junior. En la tabla 8.1 se especifican los costes asociados a los recursos humanos para las horas de trabajo efectuadas.

Tipo de RRHH	Cantidad (horas)	Coste unitario (€/h)	Coste total (€)
Investigador	140	11,27	1.577,80
Programador Junior	360	9,57	3.445,20
SUBTOTAL			5.003,00

Tabla 8.1.- Costes de recursos humanos

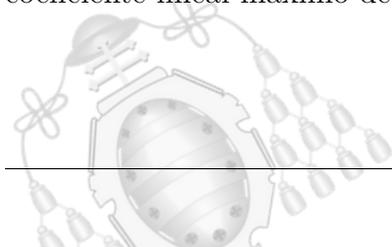
8.2.- Equipamiento Hardware

A continuación se muestran en la tabla 8.2 los costes asociados a la adquisición del equipamiento hardware necesario para la elaboración del estudio realizado.

Equipamiento hardware	Concepto	Amortización (%)	Coste total (€)
Dell Inspiron 15 5000	PC portátil	20	175,80
Samsung S9	Dispositivo móvil	20	65 80
SUBTOTAL			241,60

Tabla 8.2.- Costes de equipamiento hardware

Para la amortización se han utilizado los porcentajes fijados por hacienda [30]. El coeficiente lineal máximo de equipos electrónicos es el 20%.





8.3.- Equipamiento Software

A continuación se muestran en la tabla 8.3 los costes de las diversas licencias de software empleadas durante el desarrollo del estudio.

Equipamiento software	Cantidad	Amortización (%)	Coste total (€)
Licencia Windows 10 Pro	1	33	85,47
Licencia Office Hogar y Estudiantes	1	33	49,17
API Translation: Basic Edition	1	33	11,54
Python 3.7.4	1	33	00,00
Anaconda Individual Edition	1	33	00,00
Eclipse Committers 2019-09	1	33	00,00
TeXnicCenter: Latex Editing	1	33	00,00
SUBTOTAL			146,18

Tabla 8.3.- Costes de equipamiento software

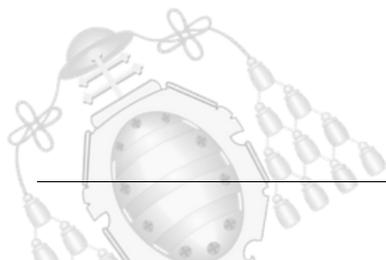
Para la amortización se han utilizado los porcentajes fijados por hacienda [30]. El coeficiente lineal máximo de sistemas y programas informáticos es el 33%.

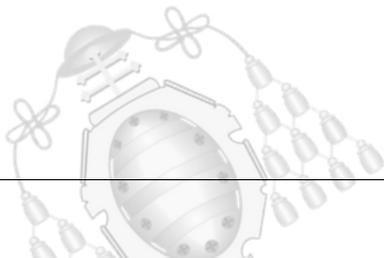
8.4.- Presupuesto Final

El coste total del proyecto es de CINCO MIL TRESCIENTOS NOVENTA CON SETENTA Y OCHO EUROS, que ha sido obtenido de los costes anteriormente desglosados.

Tipo de coste	Coste total(€)
Recursos Humanos	5.003,00
Equipamiento Hardware	241,60
Equipamiento Software	146,18
COSTE TOTAL PROYECTO	5.390,78

Tabla 8.4.- Costes totales

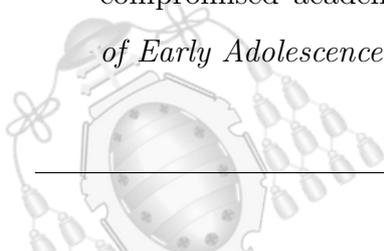






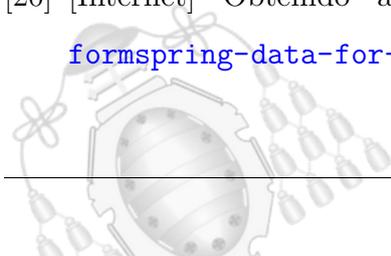
Bibliografía

- [1] P. K. Smith, J. Mahdavi, M. Carvalho, S. Fisher, S. Russell y N. Tippett. (2008). “Cyberbullying: Its nature and impact in secondary school pupils.” *Journal of child psychology and psychiatry*, vol. 49, pp. 376–385.
- [2] [Internet] B. Belsey. (2009). Cyberbullying. Creador de la página web <https://cyberbullying.org/> [citado 8 Jul 2020]
- [3] D. Halpern, M. Piña y J. Vázquez. (2017). “Soledad, bienestar social e individual: hacia una conceptualización de los efectos del cyberbullying.” *Cultura y Educación*, vol. 29, no. 4, pp. 703–727.
- [4] K. Kumpulainen, E. Räsänen y K. Puura. (2001). “Psychiatric disorders and the use of mental health services among children involved in bullying.” *Aggressive behavior*, vol. 27, no. 2, pp. 102–110.
- [5] A.-H. Luukkonen, K. Riala, H. Hakko y P. Räsänen. (2010). “Bullying behaviour and substance abuse among underage psychiatric inpatient adolescents.” *European psychiatry*, vol. 25, no. 7, pp. 382–389.
- [6] A. B. Klomek, A. Sourander y M. Gould. (2010). “The association of suicide and bullying in childhood to young adulthood: a review of crosssectional and longitudinal research findings.” *The Canadian Journal of Psychiatry*, vol. 55, no. 5, pp. 282–288.
- [7] W. M. Craig. (1998). “The relationship among bullying, victimization, depression, anxiety, and aggression in elementary school children.” *Personality and individual differences*, vol. 24, no. 1, pp. 123–130.
- [8] J. Juvonen, Y. Wang y G. Espinoza. (2011). “Bullying experiences and compromised academic performance across middle school grades.” *The Journal of Early Adolescence*, vol. 31, no. 1, pp. 152–173.



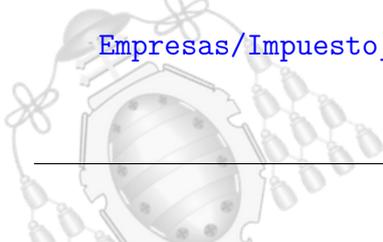


- [9] G. Álvarez Idarraga. (2018). “Educación y Futuro.” *Revista de investigación aplicada y experiencias educativas*, no. 38, pp. 111–112.
- [10] R. Donegan. (2012). “Bullying and cyberbullying: History, statistics, law, prevention and analysis.” *Elon J. Undergrad. Res. Commun.*, vol. 3, no. 1, pp. 33–42.
- [11] K. Van Royen, K. Poels, W. Daelemans y H. Vandebosch. (2014). “Telematics and Informatics Automatic monitoring of cyberbullying on social networking sites: From technological feasibility to desirability.” *Telemat. INFORMATICS*.
- [12] S. Salawu, Y. He y J. Lumsden. (2017). “Approaches to Automated Detection of Cyberbullying: A Survey.” *vol. 3045*, no. c, pp. 1–20.
- [13] H. Sanchez y S. Kumar. (2011). “Twitter bullying detection.” *ser. NSDI*, vol. 12, p. 15.
- [14] V. Nahar, X. Li y C. Pang. (2013). “An effective approach for cyberbullying detection.” *Commun. Inf. Sci. Manag. Eng.*, vol. 3, no. 5, p. 238.
- [15] C. Paper. (2016). “Methods for detection of cyberbullying: A survey.”
- [16] E. A. Abozinadah, A. V Mbaziira y J. H. J. Jr. (2015). “Detection of Abusive Accounts with Arabic Tweets.” *vol. 1*, no. 2.
- [17] M. Dadvar, D. Trieschnigg, R. Ordelman y F. De Jong. “Improving cyberbullying detection with user context.” *pp. 2–5*.
- [18] V. Nahar, S. Al-Maskari, X. Li y C. Pang. (2014). “Semi-supervised learning for cyberbullying detection in social networks.” *in Australasian Database Conference*, pp. 160–171.
- [19] P. K. Atrey. (2014). “Cyber Bullying Detection Using Social and Textual Analysis.” *pp. 3–6*.
- [20] [Internet] Obtenido a través de <https://www.kaggle.com/swetaagrawal/formspring-data-for-cyberbullying-detection> [citado 11 May 2020]





- [21] S. Bird and E. Loper, “Nltk: the natural language toolkit”, in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, p. 31, Association for Computational Linguistics, 2004.
- [22] [Internet] Google Cloud Platform: API Translate <https://cloud.google.com/translate> [citado 20 May 2020]
- [23] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [24] [Internet] Página oficial de Keras <https://keras.io/api/preprocessing/text/> [citado 7 Jun 2020]
- [25] [Internet] Página oficial de Keras https://keras.io/api/layers/core_layers/dense/ [citado 7 Jun 2020]
- [26] [Internet] Página oficial de Keras https://keras.io/api/layers/core_layers/embedding/ [citado 14 Jun 2020]
- [27] [Internet] Página oficial de Keras https://keras.io/api/layers/recurrent_layers/lstm/ [citado 14 Jun 2020]
- [28] [Internet] Towards Data Science: “Train/Test Split and Cross Validation in Python.” <https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6> [citado 8 Jul 2020]
- [29] [Internet] Aprende Machine Learning: “Qué es overfitting y underfitting y cómo solucionarlo.” <https://www.aprendemachinlearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/> [citado 9 Jul 2020]
- [30] [Internet] Página oficial Agencia Tributaria https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_





[de_1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml](#) [citado 13 Jul 2020]

