Trabajo final de evaluación a distancia para la asignatura: "Introducción al Aprendizaje Automático para Ciberseguridad"

curso 2019-20

1. Información previa

Es importante que en primer lugar **lea todo este documento completo detenidamente** y hasta el final, ya que proporciona de forma detallada toda la información necesaria para realizar esta actividad. También es necesario haber leído previamente el Manual Didáctico en el curso virtual de la asignatura y haber seguido las recomendaciones de estudio incluidas en el mismo.

En especial, es importante que cuanto antes se haga el esbozo de planteamiento indicado en el apartado 3.1 para poder realizar la **consulta por email con el coordinador de la asignatura** sobre el problema elegido.

1.1. Resumen

Se propone la realización de un trabajo práctico individual aplicado a un tema elegido por cada estudiante. El trabajo consiste en aplicar los métodos de aprendizaje automático estudiados en la asignatura a un tema relacionado con Ciberseguridad, siguiendo los ejemplos dados en el libro base que se indica en el Manual Didáctico. Se deben seguir las pautas y restricciones indicadas en este documento y consultar con el coordinador de la asignatura, tal como se indica.

1.2. Forma de entrega

Esta es un tarea individual y personal de cada estudiante, no se permiten grupos. El resultado de la actividad debe redactarse en un documento, cuyo contenido y estructura se especifica detalladamente más adelante (sección 1.4), convertido a formato **PDF**. Ese documento y todos los ficheros relacionados con su respuesta (código fuente, datos, etc.; ver 3) se deben empaquetar juntos en un archivo comprimido compatible ZIP o RAR. Ese archivo comprimido se debe subir a través de la tarea correspondiente **dentro del curso virtual** (en la página de "Actividades Evaluables" del menú izquierdo):
`https://2020.cursosvirtuales.uned.es/dotlrn/posgrados/asignaturas/31109097-20/?page_num=3`

Por favor, **no** envíen la respuesta por ningún otro medio, ni por correo electrónico, para evitar confusiones, extravíos y problemas de gestión. Tampoco se aceptarán entregas fuera del plazo establecido para todos los estudiantes.

1.3. Plazo de entrega

La entrega se debe realizar subiendo el fichero correspondiente en la tarea del curso virtual **antes del día 30 de junio de 2020 a las 23:55h.** Se recomienda no esperar hasta el último día, para evitar las saturaciones[1] o problemas puntuales de acceso a internet, etc. Si surge algún problema para el envío, realice lo antes posible una consulta directamente al coordinador del equipo docente por email (pero no envíe el fichero por ese medio) a <jras@dia.uned.es>.

[1] Téngase en cuenta que entre los días **25 de mayo al 13 de junio** en aLF (la plataforma de cursos virtuales), y entre los días **15 al 19 y 22 al 26 de junio** en AvEx (plataforma `unedenlinea.es`), se realizarán de **9h a 20h** pruebas sincronizadas en otras asignaturas, que podrían causar congestiones puntuales. Durante esos días se recomienda subir los ficheros fuera de ese horario (es decir, hacerlo preferiblemente después de las 20h y antes de las 8h.).

La respuesta de la actividad se puede enviar en el curso virtual tantas veces como se quiera hasta el plazo indicado, ya que solamente la **última versión** se guarda y es la única que se tendrá en cuenta para la evaluación al finalizar el plazo.

Se recomienda verificar que el fichero guardado en el servidor es el correcto, descargándolo después del envío para comprobar que es la versión correcta y tiene todos contenidos requeridos.

1.4. Contenidos y estructura del documento de respuesta

El documento de respuesta puede redactarse en cualquier editor, pero se debe **exportar a PDF** para enviarlo. Por ejemplo, se puede usar el editor de LibreOffice que permite exportar a PDF, es gratuito y está disponible para todas las plataformas (ver `http://libreoffice.org/` para obtenerlo). También es muy recomendable el programa LyX (ver `http://lyx.org/`) que puede generar PDF de calidad fácilmente y está disponible en la mayoría de distribuciones de Linux, y también para MS-Windows o Mac.

En el documento de respuesta (indicado en el apartado 1.2) se deben especificar los resultados de la actividad, y también las explicaciones de cómo se han obtenido éstos, ordenados según el siguiente **esquema**:

1. Datos del estudiante: Nombre, Apellidos, DNI o pasaporte y email de contacto. También se puede incluir opcionalmente un número de teléfono de contacto.

2. Información sobre el entorno, programas y medios usados para realizar esta actividad (incluyendo números de versiones): Sistema operativo, entorno de escritorio, lenguaje de programación, librerías, programas o paquetes adicionales, editor de texto para generar el documento PDF, etc.

3. Breve resumen de la actividad a realizar.

4. Respuestas de cada uno de los apartados pedidos en la actividad (ver apartado 3). Es necesario separar y etiquetar claramente cada apartado y, además de la información o datos específicos requeridos en cada apartado, se recomienda incluir también lo siguiente:

   a) Qué fuentes de información se han consultado.

   b) Qué implementaciones se han usado como ejemplo (código, datos, etc.) y qué modificaciones se han hecho respecto a los originales.

   c) Qué fallos o errores se han encontrado y cómo se han solucionado.

5. Comentarios y opiniones: Dificultades o problemas encontrados, programas o ayudas utilizadas, comentarios sobre la realización de la actividad, etc.

6. Bibliografía o fuentes de información: Documentos, páginas web, libros, etc. consultados para la realización (con datos de título y editorial o URL de localización, etc.).

Es muy importante para la evaluación, seguir el esquema dado e **incluir explicaciones** de cómo se han realizado los procesos de cada etapa (qué lista de datos, operación, resultado, etc.) para que se pueda calificar correctamente.

Utilice un formato sencillo y claro, con una redacción estructurada en apartados numerados en el documento. Incluya diagramas, figuras, e imágenes que expliquen el contenido, pero por favor, no utilice ilustraciones o formatos decorativos innecesarios, pues pueden valorarse negativamente.

No incluya largos listados **de código** fuente, **ni de datos** usados. Los ficheros necesarios, que no se puedan indicar mediante enlaces con descarga pública, se deben añadir fuera del documento PDF, en el fichero comprimido para subir a la actividad.

Para incluir pequeños fragmentos de código, o datos puntuales y resultados concretos obtenidos en comandos de consola de texto, se debe copiar el texto y "pegarlo" como texto en el documento. Se le puede dar formato con un tipo de letra de paso fijo (*fixed font*) y preservar los saltos de línea[2]. No se deben insertar capturas gráficas de la pantalla de las ventanas de terminal de consola que ocupan bastante más y pueden tener peor calidad.

Revise detalladamente, toda la información y los datos específicos pedidos en cada apartado de los resultados requeridos (ver apartado 3 de este enunciado), para asegurarse de que las respuestas se incluyen en el documento en las secciones apropiadas.

[2]En el procesador LyX se puede insertar el texto dentro de un "Listado de Código" que preserva el espaciado.

2. Restricciones y condiciones del material a utilizar

En la elaboración del trabajo se pueden usar únicamente los entornos, software y datos que cumplan los siguientes requisitos:

- El sistema operativo donde se pueda comprobar la ejecución y los resultados debe ser gratuito, y con licencia que permita su uso y distribución libres. Además debe ser una versión reciente que esté disponible para descarga e instalación.
  Se recomienda el uso de alguna distribución de Linux de una versión reciente (tipo Debian, Ubuntu o CentOS), y que no tenga muchos requisitos de memoria, para que pueda ser instalado en máquinas virtuales.

- Cualquier software (programas, aplicaciones, compiladores, bibliotecas o librerías, etc.) necesario, debe ser también gratuito y de libre distribución, que además se pueda descargar, usar e instalar en el sistema operativo utilizado (según las restricciones indicadas antes). Es preferible que el software tenga licencias compatibles con la GNU-GPL (*General Public License*) de GNU.
  Los programas o scripts elaborados por el estudiante (o modificaciones de otros), se deben adjuntar en el archivo comprimido a entregar en la tarea.

- Todos los datos que se requieran para el funcionamiento, entrenamiento, prueba y ejecución de los programas utilizados deben ser, o bien **adjuntados** en la entrega del trabajo, o bien deben estar **disponibles para descarga** con licencia de uso libre o de dominio público.

- Por supuesto, todos los datos utilizados deben haber sido **obtenidos de forma legal**, con permiso y cumpliendo todas las normativas y legislación aplicables.
  En especial, se recuerda que cualquier conjunto de datos, que pueda incluir datos personales, debe cumplir el "Reglamento General de Protección de Datos" y la correspondiente "Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales".

3. Resultados requeridos

En esta actividad se pide que cada estudiante elija un problema relacionado con la Ciberseguridad y que aplique para resolverlo alguna de las técnicas (agrupamiento, clasificación, modelos probabilísticos o *Deep Learning*) explicadas en la asignatura, con los métodos o algoritmos (K-means, DBSCAN, Árboles de Decisión, Naïve Bayes, Modelo de Mezclas Gausianas, Redes Neuronales Recurrentes o Convolucionales) incluidos en los ejemplos dados en el libro base. Aunque el problema al que se aplique debe ser diferente de los que se muestran como ejemplos en el libro base, sí puede hacerse uso de parte de los *scripts* e infraestructura que se proporciona en el repositorio oficial del libro, `https://www.cylance.com/intro-to-ai`. De la misma forma, el problema puede ser similar, pero no igual, a otros que se resuelvan como ejemplo, o de los que se proporcione implementación completa, disponibles en Internet. Aunque, también se puede hacer uso parcial modificado de ellos, siempre que se den las referencias de los originales y se indiquen claramente las diferencias y modificaciones realizadas.

Se recomienda, repasar los datos disponibles en el repositorio original de datos usado en algunos apartados en el libro base, "*SecRepo.com: Samples of Security Related Data*": `https://www.secrepo.com/`. En este repositorio se incluyen muchos ejemplos de registros de accesos de diversas aplicaciones y servicios que pueden contener información de posibles ataques informáticos. Incluye también los registros de acceso a su propio servidor web que se utilizan en el libro base para sus ejemplos de agrupamiento.

Adicionalmente, se puede encontrar gran cantidad de información sobre aprendizaje automático en Ciberseguridad que puede servir de inspiración en el repositorio de GitHub "*Machine Learning for Cyber Security*", y en especial en su apartado de *Datasets* con varios repositorios de datos para ejemplo, entrenamiento y pruebas: `https://github.com/wtsxDev/Machine-Learning-for-Cyber-Security#-datasets`. También hay varios Datasets sobre Ciberseguridad en: `https://www.unb.ca/cic/datasets/index.html`.

Es muy importante que cada estudiante realice, después de leer el resto de este documento, un primer esbozo rápido sobre el problema elegido, con los datos a utilizar y los algoritmos que se intentarán aplicar, y se **envíe, lo antes posible por email <jras@dia.uned.es>, una consulta al coordinador de la asignatura** con esa información, para que le pueda asesorar sobre la adecuación y dificultades del problema y los métodos elegidos, o sugerir alguna modificación o alternativas factibles y apropiadas.

3.1. Planteamiento

El planteamiento inicial del problema elegido a resolver, y los métodos elegidos para hacerlo, son la parte más importante de este trabajo, y de hecho la puntuación de esta parte constituirá **la mitad de la calificación** del trabajo.

Se debe incluir en el documento de respuesta de la actividad (explicado en el apartado 1.4 de este enunciado), un apartado donde se deben indicar:

1. La descripción y planteamiento del problema elegido que se pretende resolver. Se debe explicar en qué cuáles son las dificultades para resolverlo por métodos tradicionales. Si es posible, dar ejemplos de otros problemas similares y también de otros métodos conocidos para resolverlos, etc.

2. Una descripción de los datos a utilizar, tanto para entrenamiento o ajuste, como para ejemplos de pruebas, etc. Se debe indicar cuál es la fuente de la que se obtendrían los datos en un caso real y cuáles son los que se van a usar en este trabajo (artificiales o ficticios, fabricados o ejemplos antiguos, etc.) que cumplan con los requisitos del apartado 2.

3. Cuáles son los algoritmos de aprendizaje automático que se pretenden utilizar y los métodos que se van a aplicar para resolver el problema. Es necesario explicar cómo se realizará el proceso: qué transformación de los datos iniciales se hará, cómo se extraerá de ellos la información que se utilizará para el entrenamiento o ajuste, y cómo se debe interpretar esa información (casos, etiquetas, elementos, vectores, coordenadas, etc.) para aplicar los algoritmos elegidos.

4. Los resultados esperados del proceso, después de la extracción de datos y su utilización en el entrenamiento o ajuste de los algoritmos. Se debe explicar cuál es la interpretación de los resultados que indique la solución del problema planteado inicialmente.

3.2. Implementación

Una vez realizado el planteamiento detallado del problema elegido, junto con los datos y métodos de resolución apropiados, se debe realizar una implementación en el lenguaje de programación apropiado, o utilizando los entornos software correspondientes, que cumplan con los requisitos del apartado 2.

Se debe explicar el desarrollo sobre cómo se ha implementado la solución: lenguaje, entorno de programación, fuentes, *scripts*, etc. Aunque se recomienda que se utilicen entornos similares a los utilizados en el libro base de la asignatura (scikit-learn, Python, etc.), también es posible realizarlos en otros lenguajes de programación habituales.

Como respuesta de este apartado (en el documento de la actividad explicado en el apartado 1.4), se deben indicar:

1. El lenguaje o entorno de programación utilizado y la estructura de los programas o ficheros necesarios (no incluir listados en el documento, sino adjuntar los ficheros en el mismo archivo comprimido .zip).

2. La explicación de la estructura original de los datos y qué modificaciones se han realizado, incluyendo las instrucciones, código, *scripts*, etc. que se han escrito o desarrollado para esas modificaciones y preprocesado.

3. La forma en la que se deben compilar (si fuera necesario) o ejecutar los programas desarrollados y cuáles son los argumentos, opciones o parámetros que se pueden usar con ellos.

4. Indicar cuáles son las limitaciones o restricciones en la implementación (valores extremos o fuera de rango, etc.) y cuáles son los valores por omisión (por defecto) de los parámetros y porqué se han elegido así.

3.3. Pruebas y resultados

Aunque no es imprescindible que al final se obtengan unos resultados concluyentes que resuelvan definitivamente el problema planteado, sí es interesante hacer un análisis de las pruebas, experimentos o intentos realizados junto con una interpretación de los resultados (tanto positivos como negativos).

Se debe incluir en el documento de respuesta de la actividad (explicado en el apartado 1.4 de este enunciado), un apartado donde se deben indicar:

1. Cuál es el proceso que se debe seguir para obtener los datos iniciales y cómo se deben procesar para realizar los ajustes o entrenamiento previos.

2. Cómo se aplican los programas o *scripts* desarrollados para utilizar los datos en la solución del problema.

3. Qué casos se han comprobado y con qué valores de datos iniciales y de parámetros, indicando porqué se han usado esos valores.

4. Cuáles son los resultados obtenidos y qué interpretación tienen en el contexto de la solución del problema, indicando si son, o no, los valores esperados o deseables.

5. Qué otros valores es podrían comprobar, o qué otros casos sería posible estudiar, indicando qué modificaciones o añadidos serían necesarios en los programas y en los datos.