EDITORES:

Manuel A. Serrano - Eduardo Fernández-Medina

Cristina Alcaraz - Noemí de Castro - Guillermo Calvo

Actas de las VI Jornadas Nacionales
(JNIC2021 LIVE)

INVESTIGACIÓN EN ciber segu ridad

Ediciones de la Universidad
de Castilla-La Mancha

Investigación en Ciberseguridad

Actas de las VI Jornadas Nacionales (JNIC2021 LIVE)

Online 9-10 de junio de 2021
Universidad de Castilla-La Mancha

Investigación en Ciberseguridad

Actas de las VI Jornadas Nacionales (JNIC2021 LIVE)

Online 9-10 de junio de 2021
Universidad de Castilla-La Mancha

Editores:
Manuel A. Serrano,
Eduardo Fernández-Medina,
Cristina Alcaraz
Noemí de Castro
Guillermo Calvo

Ediciones de la Universidad
de Castilla-La Mancha

Cuenca, 2021




Edita: Ediciones de la Universidad de Castilla-La Mancha

Colección JORNADAS Y CONGRESOS n.º 34

Esta editorial es miembro de la UNE, lo que garantiza la difusión y comercialización de sus publicaciones a nivel nacional e internacional.

I.S.B.N.: 978-84-9044-463-4
D.O.I.: http://doi.org/10.18239/jornadas_2021.34.00

Hecho en España (U.E.) – *Made in Spain (E.U.)*

Bienvenida del Comité Organizador

Tras la parada provocada por la pandemia en 2020, las VI Jornadas Nacionales de Investigación en Ciberseguridad (JNIC) vuelven el 9 y 10 de Junio del 2021 con energías renovadas, y por primera vez en su historia, en un formato 100% online. Esta edición de las JNIC es organizada por los grupos GSyA y Alarcos de la Universidad de Castilla-La Mancha en Ciudad Real, y con la activa colaboración del comité ejecutivo, de los presidentes de los distintos comités de programa y del Instituto Nacional de Ciberseguridad (INCIBE). Continúa de este modo la senda de consolidación de unas jornadas que se celebraron por primera vez en León en 2015 y le siguieron Granada, Madrid, San Sebastián y Cáceres, consecutivamente hasta 2019, y que, en condiciones normales se habrían celebrado en Ciudad Real en 2020.

Estas jornadas se han convertido en un foro de encuentro de los actores más relevantes en el ámbito de la ciberseguridad en España. En ellas, no sólo se presentan algunos de los trabajos científicos punteros en las diversas áreas de ciberseguridad, sino que se presta especial atención a la formación e innovación educativa en materia de ciberseguridad, y también a la conexión con la industria, a través de propuestas de transferencia de tecnología. Tanto es así que, este año se presentan en el Programa de Transferencia algunas modificaciones sobre su funcionamiento y desarrollo que han sido diseñadas con la intención de mejorarlo y hacerlo más valioso para toda la comunidad investigadora en ciberseguridad.

Además de lo anterior, en las JNIC estarán presentes excepcionales ponentes (Soledad Antelada, del Lawrence Berkeley National Laboratory, Ramsés Gallego, de Micro Focus y Mónica Mateos, del Mando Conjunto de Ciberdefensa) mediante tres charlas invitadas y se desarrollarán dos mesas redondas. Éstas contarán con la participación de las organizaciones más relevantes en el panorama industrial, social y de emprendimiento en relación con la ciberseguridad, analizando y debatiendo el papel que está tomando la ciberseguridad en distintos ámbitos relevantes.

En esta edición de JNIC se han establecido tres modalidades de contribuciones de investigación, los clásicos artículos largos de investigación original, los artículos cortos con investigación en un estado más preliminar, y resúmenes extendidos de publicaciones muy relevantes y de alto impacto en materia de ciberseguridad publicados entre los años 2019 y 2021. En el caso de contribuciones de formación e innovación educativa, y también de transferencias se han considerado solamente artículos largos. Se han recibido para su valoración un total de 86

contribuciones organizadas en 26, 27 y 33 artículos largos, cortos y resúmenes ya publicados, de los que los respectivos comités de programa han aceptado 21, 19 y 27, respectivamente. En total se ha contado con una ratio de aceptación del 77%. Estas cifras indican una participación en las jornadas que continúa creciendo, y una madurez del sector español de la ciberseguridad que ya cuenta con un volumen importante de publicaciones de alto impacto.

El formato online de esta edición de las jornadas nos ha motivado a organizar las jornadas de modo más compacto, distinguiendo por primera vez entre actividades plenarias (charlas invitadas, mesas redondas, sesión de formación e innovación educativa, sesión de transferencia de tecnología, junto a inauguración y clausura) y sesiones paralelas de presentación de artículos científicos. En concreto, se han organizado 10 sesiones de presentación de artículos científicos en dos líneas paralelas, sobre las siguientes temáticas: detección de intrusos y gestión de anomalías (I y II), ciberataques e inteligencia de amenazas, análisis forense y cibercrimen, ciberseguridad industrial, inteligencia artificial y ciberseguridad, gobierno y riesgo, tecnologías emergentes y entrenamiento, criptografía, y finalmente privacidad.

En esta edición de las jornadas se han organizado dos números especiales de revistas con elevado factor de impacto para que los artículos científicos mejor valorados por el comité de programa científico puedan enviar versiones extendidas de dichos artículos. Adicionalmente, se han otorgado premios al mejor artículo en cada una de las categorías. En el marco de las JNIC también hemos contado con la participación de la Red de Excelencia Nacional de Investigación en Ciberseguridad (RENIC), impulsando la ciberseguridad a través de la entrega de los premios al *Mejor Trabajo Fin de Máster en Ciberseguridad* y a la *Mejor Tesis Doctoral en Ciberseguridad*. También se ha querido acercar a los jóvenes talentos en ciberseguridad a las JNIC, a través de un CTF (Capture The Flag) organizado por la Universidad de Extremadura y patrocinado por Viewnext.

Desde el equipo que hemos organizado las JNIC2021 queremos agradecer a todas aquellas personas y entidades que han hecho posible su celebración, comenzando por los autores de los distintos trabajos enviados y los asistentes a las jornadas, los tres ponentes invitados, las personas y organizaciones que han participado en las dos mesas redondas, los integrantes de los distintos comités de programa por sus interesantes comentarios en los procesos de revisión y por su colaboración durante las fases de discusión y debate interno, los presidentes de las sesiones, la Universidad de Extremadura por organizar el CTF y la empresa Viewnext por patrocinarlo, los técnicos del área TIC de la UCLM por el apoyo con la plataforma de comunicación, los voluntarios de la UCLM y al resto de organizaciones y entidades patrocinadoras, entre las que se encuentra la Escuela Superior de Informática, el Departamento de Tecnologías y Sistemas de Información y el Instituto de Tecnologías y Sistemas de Información, todos ellos de la Universidad de Castilla-La Mancha, la red RENIC, las cátedras (Telefónica e Indra) y aulas (Avanttic y Alpinia) de la Escuela Superior de Informática, la empresa Cojali, y muy especialmente por su apoyo y contribución al propio INCIBE.

Manuel A. Serrano, Eduardo Fernández-Medina
Presidentes del Comité Organizador
Cristina Alcaraz
Presidenta del Comité de Programa Científico
Noemí de Castro
Presidenta del Comité de Programa de Formación e Innovación Educativa
Guillermo Calvo Flores
Presidente del Comité de Transferencia Tecnológica

Índice General

Comité Ejecutivo

Juan Díez González	INCIBE
Luis Javier García Villalba	Universidad de Complutense de Madrid
Eduardo Fernández-Medina Patón	Universidad de Castilla-La Mancha
Guillermo Suárez-Tangil	IMDEA Networks Institute
Andrés Caro Lindo	Universidad de Extremadura
Pedro García Teodoro	Universidad de Granada. Representante de red RENIC
Noemí de Castro García	Universidad de León
Rafael María Estepa Alonso	Universidad de Sevilla
Pedro Peris López	Universidad Carlos III de Madrid

Comité Organizador

Presidentes del Comité Organizador

Eduardo Fernández-Medina Patón	Universidad de Castilla-la Mancha
Manuel Ángel Serrano Martín	Universidad de Castilla-la Mancha

Finanzas

David García Rosado	Universidad de Castilla-la Mancha
Luis Enrique Sánchez Crespo	Universidad de Castilla-la Mancha

Actas

Antonio Santos-Olmo Parra	Universidad de Castilla-la Mancha

Difusión

Julio Moreno García-Nieto	Universidad de Castilla-la Mancha
José Antonio Cruz Lemus	Universidad de Castilla-la Mancha
María A Moraga de la Rubia	Universidad de Castilla-la Mancha

Webmaster

Aurelio José Horneros Cano	Universidad de Castilla-la Mancha

Logística y Organización

Ignacio García-Rodriguez de Guzmán	Universidad de Castilla-la Mancha
Ismael Caballero Muñoz-Reja	Universidad de Castilla-la Mancha
Gregoria Romero Grande	Universidad de Castilla-la Mancha
Natalia Sanchez Pinilla	Universidad de Castilla-la Mancha

Comité de Programa Científico

Presidenta

Cristina Alcaraz Tello	Universidad de Málaga

Miembros

Aitana Alonso Nogueira	INCIBE
Marcos Arjona Fernández	ElevenPaths
Ana Ayerbe Fernández-Cuesta	Tecnalia
Marta Beltrán Pardo	Universidad Rey Juan Carlos
Carlos Blanco Bueno	Universidad de Cantabria
Jorge Blasco Alís	Royal Holloway, University of London
Pino Caballero-Gil	Universidad de La Laguna
Andrés Caro Lindo	Universidad de Extremadura
Jordi Castellà Roca	Universitat Rovira i Virgili
José M. de Fuentes García-Romero de Tejada	Universidad Carlos III de Madrid
Jesús Esteban Díaz Verdejo	Universidad de Granada
Josep Lluis Ferrer Gomila	Universitat de les Illes Balears
Dario Fiore	IMDEA Software Institute
David García Rosado	Universidad de Castilla-La Mancha
Pedro García Teodoro	Universidad de Granada
Luis Javier García Villalba	Universidad Complutense de Madrid
Iñaki Garitano Garitano	Mondragon Unibertsitatea
Félix Gómez Mármol	Universidad de Murcia
Lorena González Manzano	Universidad Carlos III de Madrid
María Isabel González Vasco	Universidad Rey Juan Carlos I
Julio César Hernández Castro	University of Kent
Luis Hernández Encinas	CSIC
Jorge López Hernández-Ardieta	Banco Santander
Javier López Muñoz	Universidad de Málaga
Rafael Martínez Gasca	Universidad de Sevilla
Gregorio Martínez Pérez	Universidad de Murcia

David Megías Jiménez	Universitat Oberta de Cataluña
Luis Panizo Alonso	Universidad de León
Fernando Pérez González	Universidad de Vigo
Aljosa Pasic	ATOS
Ricardo J. Rodríguez	Universidad de Zaragoza
Fernando Román Muñoz	Universidad Complutense de Madrid
Luis Enrique Sánchez Crespo	Universidad de Castilla-La Mancha
José Soler	Technical University of Denmark-DTU
Miguel Soriano Ibáñez	Universidad Politécnica de Cataluña
Victor A. Villagrá González	Universidad Politécnica de Madrid
Urko Zurutuza Ortega	Mondragon Unibertsitatea
Lilian Adkinson Orellana	Gradiant
Juan Hernández Serrano	Universitat Politécnica de Cataluña

Comité de Programa de Formación e Innovación Educativa

Presidenta

Noemí De Castro García	Universidad de León

Miembros

Adriana Suárez Corona	Universidad de León
Raquel Poy Castro	Universidad de León
José Carlos Sancho Núñez	Universidad de Extremadura
Isaac Agudo Ruiz	Universidad de Málaga
Ana Isabel González-Tablas Ferreres	Universidad Carlos III de Madrid
Xavier Larriva	Universidad Politécnica de Madrid
Ana Lucila Sandoval Orozco	Universidad Complutense de Madrid
Lorena González Manzano	Universidad Carlos III de Madrid
María Isabel González Vasco	Universidad Rey Juan Carlos
David García Rosado	Universidad de Castilla - La Mancha
Sara García Bécares	INCIBE

Comité de Transferencia Tecnológica

Presidente

Guillermo Calvo Flores	INCIBE

Miembros

José Luis González Sánchez	COMPUTAEX
Marcos Arjona Fernández	ElevenPaths
Victor Villagrá González	Universidad Politécnica de Madrid
Luis Enrique Sánchez Crespo	Universidad de Castilla – La Mancha

http://doi.org/10.18239/jornadas_2021.34.10

Diseño y evaluación de modelos de aprendizaje automático no supervisado para la detección de anomalías en un sistema Spark.

Farid Bagheri-Gisour Marandyn, Xavier. Larriva-Novo, Víctor A. Villagrá
Universidad Politécnica de Madrid (UPM). DIT, ETSI Telecomunicaciones. Avda. Complutense, 30. 28040 Madrid
farid.academ@gmail.com, {xavier.larriva.novo, victor.villagra }@upm.es

Resumen- En el presente artículo se muestran los resultados obtenidos al diseñar y evaluar un modelo de aprendizaje automático no supervisado como es el K-Means para la detección de anomalías en tiempo real sobre múltiples sensores dentro de un sistema Spark utilizando un threshold para delimitar esas posibles anomalías. Los resultados obtenidos del modelo (aún en fase de desarrollo y mejora) demuestran la capacidad de poder detectar tres tipos de eventos: eventos no anómalos, eventos anómalos por características y eventos anómalos por aspectos temporales. Este comportamiento presenta características estimulantes para poder aplicar este tipo de algoritmos en un entorno real donde los datos no tienen ningún tipo de etiquetado. Todo ello, sumado a la capacidad que ofrece Spark para realizar el procesado de grandes volúmenes de datos en tiempo real, da como resultado un sistema prometedor capaz de clasificar eventos procedentes de diversos sensores de manera inmediata.

Index Terms- machine learning, K-Means, threshold, Spark, detección anomalías, ciberseguridad

Tipo de contribución: Investigación en desarrollo

I. Introducción

Hasta hace no mucho tiempo, la tarea de realizar un análisis y posterior extracción de información de la vasta amalgama de datos que una entidad empresarial posee era una tarea faraónica e inabarcable para los equipos, metodologías y mecanismos que clásicamente se venían usando en el área de ciberseguridad, más aún si las necesidades requerían que el trabajo de detección de anomalías se realizase de manera casi instantánea, a tiempo real.

Por ello, para poder solventar esos requisitos de seguridad, desde el mundo de la ciberseguridad se empezó a plantear utilizar métodos y mecanismos de una de las ramas de la computación que durante los últimos años ha experimentado un crecimiento exponencial, el campo del machine learning y el big data.

En el campo del machine learning, los modelos de aprendizaje no supervisado son los que más interés suscitan, ya que los datos que se les suministra no poseen ningún etiquetado, por lo que el conocimiento tiene que ser adquirido de la propia metaestructura de los datos entregados. Esta característica es fundamental para el análisis de la gran mayoría de los datos que se generan por los equipos, ya que dichos datos no poseen etiquetado alguno.

Dentro del big data se desarrollaron diversos mecanismos y modelos que permiten realizar el manejo y procesado de grandes volúmenes de datos a tiempo real. Dentro de las diversas propuestas se destaca Apache Spark, sistema bien conocido open-source, escalable, que permite procesar grandes volúmenes de datos mediante la paralelización del trabajo y en tiempo real gracias a Spark Streaming.

Por tanto, el objetivo del proyecto se centrará en utilizar las bondades que ofrece la herramienta Apache Spark para poder realizar un sistema de autoaprendizaje no supervisado que permita detectar anomalías de un conjunto de datos provenientes de varios sensores a tiempo real. A través de una de las API de python que ofrece Spark, denominada PySpark y que ofrece a su vez una librería de modelos de machine learning, se puede ejecutar e integrar con todo el ecosistema de Spark el algoritmo de aprendizaje no supervisado K-Means, para realizar esa tarea de detección de anomalías en tiempo real. Así mismo, el entorno provee todas las herramientas necesarias para el preprocesamiento de los datos y la posterior conexión a un sistema ELK (ElasticSearch, Logstash y Kibana) donde se podrá visualizar y analizar los resultados. Se propone ese modelo, ya que es uno de los modelos de aprendizaje no supervisado mejor conocidos que permita agrupar los datos en clústeres (conjunto de datos que se agrupan como similares).

El entrenamiento del modelo se realizará con datasets sin anomalías, que serán generados por cada fuente de datos. El correcto modelado del sistema requerirá del preprocesamiento de los datos a través de las herramientas de Pyspark mencionadas, para que el algoritmo K-Means pueda ser correctamente entrenado.

La arquitectura de detección de anomalías se basa en el diseño de un sistema de threshold junto al algoritmo K-Means que permita clasificar, en base a los clústeres formados por el propio algoritmo previamente, qué datos provenientes de las distintas fuentes de datos son anómalos. Se harán uso de técnicas que permitan optimizar el modelo a través de sus hiperparámetros y tras el diseño del sistema, se pasará a evaluar su comportamiento.

Las pruebas que se realizan determinan si el modelo efectúa una correcta clasificación de los datos en anomalía/no anomalía en función de las características de los datos, así como de la marca temporal del mismo.

Por último, se integra al sistema, sensores que proporcionan las fuentes de datos, así como un sistema ELK en donde pueden ser almacenados, gestionados y analizados los resultados. Se comprobará que el sistema cumpla los requisitos planteados en el escenario real, exponiendo aquellas ventajas y desventajas ofrecidas por la arquitectura, así como posibles mejoras y líneas futuras que puedan surgir del proyecto.

II. Metodología

El sistema de aprendizaje automático propuesto se enmarca en una arquitectura de adquisición, procesamiento y tratamiento de datos de un sistema de conciencia cibersituacional y es el

encargado de la detección de anomalías a partir de los datos de entrada del sistema. Los datos de entrada se componen de diversos conjuntos de valores numéricos en diversos formatos provenientes de distintos sensores de actividad. Estos datos no pueden ser inyectados directamente al algoritmo K-Means, sino que deben ser tratados con anterioridad para que puedan ser procesados. A su vez, el algoritmo requiere de una fase previa de entrenamiento y validación para que el sistema no muestre resultados aleatorios. Por último, los datos de salida del sistema están formadas por el conjunto de características respectivas al evento de entrada más el etiquetado por un valor 'False' o 'True' indicando si se ha detectado como anomalía o no

Por todo ello, se realizó una arquitectura compuesta por diferentes fases y subsistemas que se listan a continuación:

- Recepción de datos de los sensores.
- Subsistema de generación de datos sintéticos.
- Subsistema de preprocesamiento.
- Subsistema de entrenamiento y validación.
- Subsistema de procesamiento a tiempo real.
- Envío/almacenado de los resultados obtenidos.

Cada uno de los subsistemas está compuesto a su vez de diferentes módulos para poder realizar su tarea asignada.

La Figura 1 muestra la arquitectura del sistema realizado, con las conexiones entre los diferentes subsistemas que conforman el sistema completo.

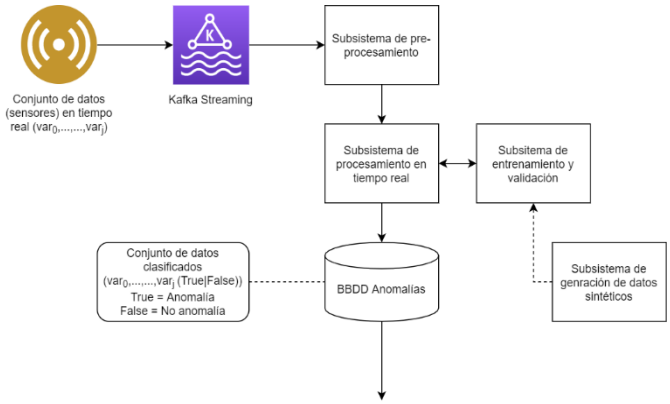

Figura 1: Arquitectura modular del sistema de aprendizaje automático

A. Subsistema de generación de datos sintéticos

Los datos que recibe la arquitectura son datos con formato json donde, dependiendo de su procedencia, tendrá campos diferentes propios de cada sensor. Estos datos serán consumidos por la arquitectura mediante los Topic de Kafka.

B. Subsistema de generación de datos sintéticos

El subsistema de generación de datos sintéticos está destinado a la creación de datasets sintéticos en aquellos modelos en los que no se posea un dataset acorde a las posteriores necesidades del modelo, ya que hay ciertos sensores que no pueden recolectar las suficientes entradas de datos distintas por estar desplegadas en un entorno no real.

La arquitectura del subsistema se muestra en la Figura 2, donde se puede observar los diferentes módulos que la componen.

El subsistema utiliza los datos que se quiera proporcionar a su entrada y comienza a establecer las relaciones entre los atributos necesarias para que no se produzcan entradas no válidas. A su vez, se definen los perfiles de generación de eventos, estableciéndose el número de eventos que se producen por paso de reloj y seguidamente, se añade la probabilidad de que cada perfil se ejecute por paso de reloj.

La configuración del reloj se establece con anterioridad, indicando el tiempo de simulación, así como el tiempo entre pasos. Esta configuración estará condicionada por un perfil temporal, en donde se indica las características de generación de eventos según la hora (p. ej. mayor cantidad de pasos en horas laborables).

Por último, se definen los atributos a generar ya con todas las configuraciones, relaciones, perfiles y se pasa a su generación. Cabe indicar que se establecen dos configuraciones distintas relativas a la generación de un dataset de tipo normal y otro de tipo anómalo.

Todas estas configuraciones son relativas a la librería para la generación de datasets realistas Trumania [1], librería que ofrece las herramientas necesarias para poder generar sintéticamente datos, definir estructuras internas del dataset e impedir o permitir la generación de eventos que se ajusten a las siguientes necesidades:

- Tipos adecuados para todos los valores del dataset, facilitando su posterior modificación.
- Estructura de creación de eventos no uniforme
- Estructura temporal verosímil respecto a escenarios reales normales y anómalos.
- La no posibilitación de creación de eventos que no puedan darse en entornos reales.

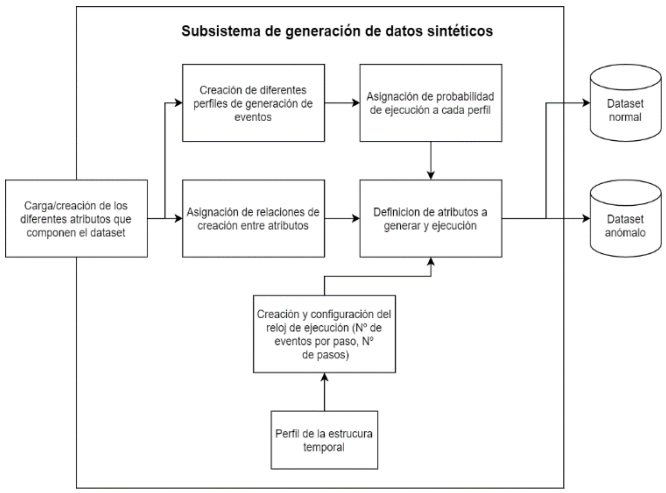

Figura 2: Arquitectura modular del subsistema de generación de datos sintéticos

Los datos de salida del subsistema corresponden a la pareja de dataset generados para un modelo específico, un dataset con características normales y un dataset con características anómalas. El tamaño de estos datasets creados son del orden de 100.000 entradas.

C. Subsistema de preprocesamiento de datos

El subsistema de preprocesamiento de datos es el encargo de aplicar modelos algorítmicos y funciones matemáticas de acuerdo con los datos con los que trabajen los diferentes modelos de machine learning de cada sensor.

El subsistema de preprocesamiento de datos queda definido en la Figura 3, donde se puede observar los diferentes subpartes que la componen.

Este preprocesamiento es previo al entrenamiento de los algoritmos de machine learning y al procesamiento en tiempo real. El subsistema se compone de:

- Estructurado de los datos.

- Aplicación de los módulos y funciones de preprocesamiento.
- Confección de vector de características final.

Los datos de entrada del subsistema de preprocesamiento se componen del conjunto de valores de datos respectivos a cada uno de los sensores, que deben ser procesados por los algoritmos de machine learning.

Esos datos son primeramente estructurados según el tipo de valor que tengan, teniendo así una definición del tipo de datos que aparecerá en los datos de entrada.

Seguidamente entran en juego los módulos de preprocesamiento ofrecidas por Pyspark y las funciones matemáticas definidas. Entre los diferentes módulos usados (dependiendo del dato con el que se lidie) se encuentran:

String Indexer. Módulo de preprocesamiento que codifica strings a índices numéricos.

Min Max Scaler. Módulo de preprocesamiento que normaliza por defecto valores numéricos al rango [0, 1].

One Hot Encoder (OHE). Módulo de preprocesamiento que asigna a valores categóricos un vector binario. La longitud del vector binario dependerá de la cantidad de valores categóricos distintos que posean los datos a procesar.

Regex Tokenizer. Módulo de preprocesamiento que realiza la separación del texto en múltiples subgrupos según la expresión regex definida.

Count Vectorizer. Módulo de preprocesamiento que transforma un conjunto de strings a vectores de token.

TF-IDF. Módulo de preprocesamiento que refleja la importancia de los términos de un texto dentro de un corpus.

Módulos de preprocesamiento específicos de cada modelo. Este conjunto hace referencia a aquellas transformaciones concretas necesarias para adecuar los datos de entrada de cada modelo.

Vector Assembler. Módulo de preprocesamiento que combina una lista de columnas dadas en un único vector columna. Esto es el paso final del preprocesamiento.

Los datos de salida del subsistema de preprocesamiento corresponden a vectores unidimensionales que recogen las características preprocesadas de los datos de entrada ofrecida por los sensores y/o datasets.

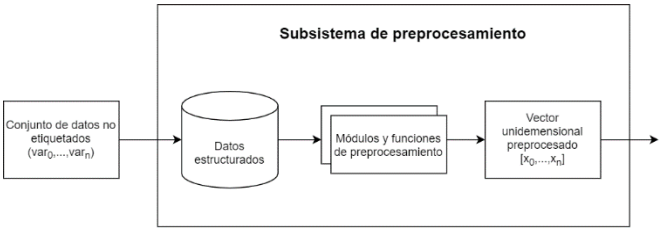

Figura 3: Arquitectura modular del subsistema de preprocesamiento

D. Subsistema de entrenamiento y validación

El subsistema de entramiento y validación es el encargado de la preparación y la supervisión del correcto funcionamiento del modelo de machine learning para la detección de anomalías por cada tipo de sensor.

Para ello, se utilizan los módulos de selección de hiperparámetros, métricas y datos de validación que se pueden observar en la Figura 4, en el que se muestra la arquitectura completa de este subsistema.

Los datos de entrada serán esos vectores ofrecidos por el subsistema de preprocesamiento, que pasarán a ser usados para entrenar y validar las diferentes propuestas de configuración del modelo de machine learning.

Seguidamente, el subsistema de selección de hiperparámetros y métricas permite optimizar y analizar el rendimiento que el algoritmo de machine learning muestra al variar los hiperparámetros, en un proceso iterativo, que lo componen. Son dos las métricas que se usan para medir el grado de mejora:

WSSSE. Mide por cada punto del dataset cuan lejano está del centroide de los clústeres. Se busca reducir esta medida lo máximo posible sin llegar al overfitting.

$$\sum_{k=1}^{K} \sum_{i \in S_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2 \qquad (1)$$

donde,
k = Clúster
S_k = Datos del cluster k,
x_{kj} = valor j del vector centroid del clúster k

Silhouette. Métrica que permite conocer la cohesión dentro de un clúster a la vez que la separación con otros clústeres, midiendo así cuan bien un dato está clasificado en un clúster. El rango de valores esta entre [-1,1] donde altos valores indican que los datos están bien asignados (alta cohesión a su clúster, alta separación con otros clústeres).

$$\mathbf{s(i)} = \frac{\mathbf{b(i)} - \mathbf{a(i)}}{\mathbf{max(b(i), a(i))}} \qquad (2)$$

donde,

a(i): Distancia media con todos los puntos de su clúster
b(i): Distancia media con todos los puntos del clúster más cercano
s(i): Coeficiente Silhouette para el iésimo punto

La selección del hiperparámetro optimo será, por tanto, el resultado del compendio del valor la métrica WSSSE y el valor que proporcione la métrica Silhouette.

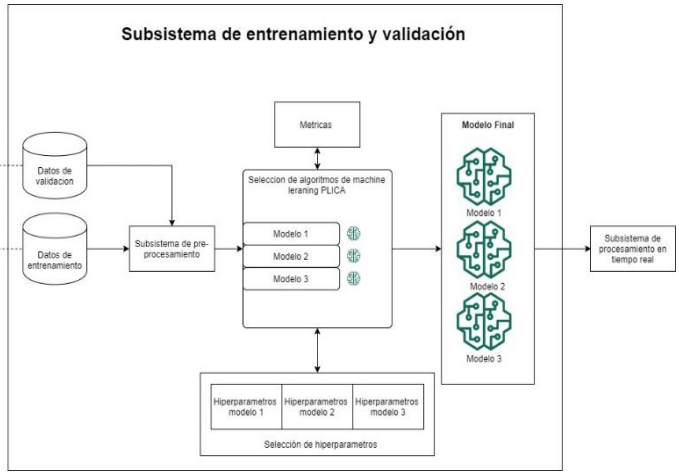

Figura 4: Arquitectura modular del subsistema de entrenamiento y validación

C.1. Modelo de Machine Learning.

El algoritmo que se usará para la realización del modelo de identificación de anomalías será el K-Means. Este algoritmo permite realizar agrupar los diferentes datos en clústeres de características similares de manera no supervisada.

La elección de este modelo radica principalmente a su amplio uso en el estado del arte del clustering y la posibilidad de su uso en la librería de Mlib de Pyspark.

Para la detección de anomalías en este modelo, se pasará a usar un threshold por cada clúster formado que permita delimitar que datos son lo suficientemente diferentes al resto de datos del clúster para que pueda ser considerado como una anomalía.

El threshold se considera como el límite de cada uno de los clústeres formados. El threshold es calculado como el punto más lejano al centroide de cada uno de los clústeres (aunque puede ser ajustado como el i-ésimo punto más lejano). Todos los puntos más lejanos a esos thresholds serán clasificados como anomalías.

Por tanto, el modelo es entrenado con un conjunto de datos que se consideran no anómalos para que en la detección de nuevos eventos se pueda comparar y observar si esos nuevos datos son lo aptos para detectarse como anomalías o no. Esta implementación permite entre otras características, tener en cuenta desplazamientos temporales de los datos, ya que la marca temporal es una característica propia de los datos.

E. Subsistema de procesamiento a tiempo real

El subsistema de procesamiento en tiempo real es el sistema que realiza finalmente la funcionalidad de la identificación de anomalías a tiempo real, mediante la integración de todas las funciones y modelos de preprocesamiento descritos, y el cargado del modelo de machine learning ya entrenado y optimizado en el paso anterior.

El sistema obtendrá los eventos de los sensores mediante un Topic de Kafka (un Topic por sensor), los preprocesará y los enviará el modelo de machine learning donde este lo clasificará según sea una posible anomalía o no.

Por último, los resultados se enviarán a elasticsearch, donde se almacenarán y podrán ser visualizados mediante kibana.

III. Resultados

Para observar el funcionamiento del sistema, y por tanto, que bien se comporta a las competencias que se le quiere asignar, se realizaron tres tipos de pruebas por cada modelo asociado a cada sensor. Se consideraron 6 sensores distintos, proporcionando datos de actividad referentes a señales Wifi, Bluetooth, redes de telefonía móvil, señales de radiofrecuencia, así como logs de firewalls y SIEMs. Las pruebas son:

- Identificación como datos normales aquellos datos que se ha visto durante el entrenamiento.
- Identificación como posibles datos anómalos aquellos datos nunca vistos y que se diferencien lo suficiente de los datos de entrenamiento.
- Identificación como anomalías aquellos datos vistos por el modelo durante el entrenamiento, pero desplazados temporalmente a una hora no común (por ejemplo, madrugada).

Los resultados arrojados por las pruebas realizadas para la confección del modelo y la medición de su desempeño son las siguientes:

De la Tabla 1 se puede aprecia que lo hiperparámetros de la tolerancia como de las máximas iteraciones son iguales. Esto se debe a que se comprobó que tales hiperparámetros no marcaban ninguna diferencia al variarse, por lo que se dejo por defecto. Por tanto, el hiperparámetro que influye considerablemente es el número de clústeres que tiene.

Tabla I
HIPERPARÁMETROS ÓPTIMOS POR CADA MODELO

Modelo del sensor	N.º de clústeres	Tolerancia	Máximas iteraciones
RM	25	1e-4	100
RF	17	1e-4	100
Bluetooth	15	1e-4	100
Wifi	17	1e-4	100
Firewall	18	1e-4	100
Siem	18	1e-4	100

Tabla II
PRECISIÓN (%) EN LA DETECCIÓN DE LOS DIFERENTES DATOS

Modelo del sensor	Normales	Anómalos por características	Anómalos temporales
RM	100.00 %	96.00 %	56.65 %
RF	94.56 %	100.00 %	71.84 %
Bluetooth	100.00 %	100.00 %	94.44 %
Wifi	100.00 %	100.00 %	72.89 %
Firewall	100.00 %	100.00 %	76.97 %
Siem	99.0 %	100.00 %	46.51 %

La Tabla 2 arroja el resultado de medir la precisión del sistema en un entorno en el que se definió que datos son normales y que datos son posibles anomalías. El resultado muestra que casi todos los datos considerados normales y posibles anomalías por características son clasificados como tal. En contrapartida, se puede ver que el rendimiento del modelo a la hora de definir anomalías temporales es mucho menor. No obstante, estos valores se deben en parte a que la distribución de detección de anomalías no es uniforme. Casi todas las anomalías detectadas son en un rango de horas de 1 a 5 AM, siendo el número de eventos aquí menor. Por tanto, no se podría hablar de un mal funcionamiento del modelo, sino un incompleto reflejo del desempeño del modelo.

IV. Conclusiones

A la vista de los datos observados se puede ver que los modelos realizados por cada sensor no tienen más complicación en realizar una detección de lo que se considero para las pruebas datos normales y posibles datos anómalos por características.

Por contraparte, el desempeño en la detección de posibles datos anómalos temporales, aún teniendo en cuenta esa distribución no uniforme, cae bastante con respecto a los otros dos y es punto de mejora de siguientes versiones.

Cabe recalcar que hay que coger estos datos con cuidado, ya que se está en fase de pruebas y se necesitan más tests para corroborar estos resultados.

V. Referencias

[1] Sv3ndk, Milanvdm, FHachez, Thomas-jakemeyn, Petervandenabeele: "Trumania", https://github.com/RealImpactAnalytics/trumania, 2020

[2] Erdem, Yadigar & Ozcan, Caner: "Fast Data Clustering and Outlier Detection using K-Means Clustering on Apache Spark", 2017

[3] Berthold, Michael & Höppner, Frank: "On Clustering Time Series Using Euclidean Distance and Pearson Correlation", 2016

[4] Peng, Kai & Huang, Qingjia: Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System over Big Data", 2018

[5] Zhang, Tao & Li, Haibin & Xu, Lexi & Gao, Jie & Guan, Jian & Cheng, Xinzhou: "Comprehensive IoT SIM Card Anomaly Detection Algorithm Based on Big Data", 2019

[6] Kumari, R. & Sheetanshu, & Singh, M. & Jha, R. & Singh, N.K.: "Anomaly detection in network traffic using K-mean clustering", 2016

[7] Han, Li.: "Using a Dynamic K-means Algorithm to Detect Anomaly Activities", 2011

[8] Lima, M.F., Zarpelão, B., Sampaio, L.D., Rodrigues, J., Abrão, T., & Proença, M.L: "Anomaly detection using baseline and K-means clustering", 2010

[9] Wazid, M., Das, A.K.:" An Efficient Hybrid Anomaly Detection Scheme Using K-Means Clustering for Wireless Sensor Networks", 2016

VI. Agradecimientos

La investigación presentada en este artículo ha sido parcialmente financiada por el proyecto PLICA dentro del programa coincidente del Ministerio de Defensa del Gobierno de España.