07-008

ONLINE TRAINING APPLIED TO THE INDUSTRIAL SECTOR: LITERATURE REVIEW AND EXISTING FRAMEWORKS

García González, Javier (1); Rodríguez Montequín, Vicente (1); Villanueva Balsera, Joaquín Manuel (1); Díaz Piloñeta, Marina (1); Valdeón Junquera, Ana María (1)

(1) Universidad de Oviedo

Although machine learning techniques have come a long way, currently the prevailing methods are those called off-line learning, in which models are trained by processing the input data set. This entails limitations such as the computational complexity to be able to train the models with large data sets, and the difficulty that the models can adapt to changing circumstances since the model would have to be retrained again. As a counterpart, the use of online learning algorithms is emerging, in which the algorithm learns from each new data that is received. This communication analyses the extend of application of this type of algorithms in the industrial sector and the existing development frameworks.

Keywords: stream data;online training;data mining

APRENDIZAJE ONLINE APLICADO AL SECTOR INDUSTRIAL: ESTADO DEL ARTE Y FRAMEWORKS EXISTENTES

Aunque las técnicas de aprendizaje automático han evolucionado mucho, en la actualidad los métodos que prevalecen son los denominados de aprendizaje off-line, en los cuales los modelos son entrenados procesando el conjunto de datos de entrada. Esto conlleva limitaciones como la complejidad computacional para poder entrenar los modelos con grandes conjuntos de datos, y la dificultad de que los modelos así entrenados se puedan adaptar a las circunstancias cambiantes de los datos pues el modelo debería ser reentrenado de nuevo. Como contrapartida está emergiendo la utilización de algoritmos de aprendizaje online, en los que el algoritmo aprende de cada nuevo dato que es recibido. Esta comunicación analiza la aplicación de este tipo de algoritmos en el sector industrial y los frameworks de desarrollo existentes.

Palabras clave: stream data;online training;data mining

Correspondencia: Vicente Rodríguez Montequín Correo: montequi@uniovi.es

Agradecimientos: Este trabajo ha sido subvencionado a través del programa de “Ayudas para Grupos de Investigación de Organismos del Principado de Asturias” (GRUPIN 2021-2023) de la Fundación para el Fomento de Asturias de la Investigación Científica Aplicada y la Tecnología (FICYT) -Gobierno del Principado de Asturias, (Ref: SV-PA-21-AYUD-2021-50953). Proyecto financiado por la Unión Europea a través del Fondo Europeo de Desarrollo Regional.

1. Introducción

La tipología de entrenamiento empleado hasta el momento para la mayoría de los proyectos de aprendizaje automático basado en datos ha sido el denominado entrenamiento offline (también conocido como aprendizaje por lotes), en el que los algoritmos aprenden de un conjunto de datos resultante de una campaña de captura de datos. Bajo este enfoque, un modelo se construye a partir de todo o una gran parte del conjunto de datos. Esta estrategia tiene la gran ventaja de que el modelo así entrenado se ajusta muy bien al conjunto de situaciones recogidas en la campaña de captura de datos, pero conlleva importantes limitaciones.

Por un lado, el conjunto de datos con el que se ha entrenado debe ser suficientemente representativo del proceso modelado. Aunque los modelos basados en datos tienen una gran capacidad para adaptarse a situaciones para las que no han sido ajustados, las condiciones en un proceso industrial pueden ser muy cambiantes (introducción de nuevos materiales, modificación de las condiciones del proceso, mala calibración de equipos, etc.). En general, cuanto más grande sea el conjunto de datos, mejor preparado estará el modelo para nuevas situaciones; de lo contrario, es necesario volver a entrenar. En cualquier caso, se hace casi imprescindible repetir el tedioso proceso de entrenamiento periódicamente para mantener el modelo bien ajustado. Además, el procesamiento de grandes conjuntos de datos requiere muchas capacidades computacionales en términos de memoria, procesador y tiempo de procesamiento, siendo habitual que el orden de complejidad de los algoritmos aumente exponencialmente respecto al tamaño del problema a resolver.

Como alternativa al aprendizaje offline surge el aprendizaje online. Bajo este enfoque los datos se tratan como un flujo o stream (habitualmente con un orden temporal), de ahí la denominación de *data* stream mining frente al término tradicional data mining. Los streams de datos son una abstracción algorítmica para admitir análisis en tiempo real. Los elementos de datos llegan uno por uno, y los modelos se construyen y mantienen en consecuencia. La estrategia de aprendizaje que se emplea en estos casos es la denominada entrenamiento precuencial o prequential training. Con este enfoque, el algoritmo realiza la predicción usando el nuevo dato, aprende del nuevo elemento y a partir de ahí puede ser descartado (el algoritmo puede mantener una pequeña ventana temporal con los datos más recientes o guardar una estructura que sintetiza estadísticamente los datos). Por lo tanto, el aprendizaje online puede ser más eficiente y adaptarse mejor a los datos.

Aunque el concepto de aprendizaje online no es nuevo y existen algoritmos que se aplican ya desde la década de los noventa, en los últimos años ha ido ganando terreno debido a la aparición de nuevos algoritmos y frameworks especialmente diseñados para esta estrategia. Aunque el auge de estas técnicas se justifica principalmente por la aplicación a fuentes de datos como las procedentes de redes sociales como Twitter, Facebook, etc., su aplicación a entornos industriales puede suponer grandes ventajas. De hecho, otro campo de aplicación natural para estos algoritmos lo supone el despliegue de sistemas de internet de las cosas (IoT), que a su vez constituye una pieza fundamental de la denominada Industria 4.0.

La presente comunicación realiza una presentación general de los puntos principales de esta estrategia de aprendizaje, su estado del arte principalmente en cuanto a frameworks existentes y las aplicaciones incipientes al sector industrial. La comunicación comienza analizando las diferencias entre los dos modos de procesamiento y los desafíos asociados al procesamiento de datos por streaming, para pasar posteriormente a presentar los frameworks y las aplicaciones incipientes y finalizar con las conclusiones.

2. Procesamiento de datos por lotes frente a datos por streaming

El estándar más utilizado para describir los enfoques comunes que afectan a la minería de datos hasta el momento es CRISP-DM (Chapman et al., 2000), el cual delimita el ciclo de vida de estos proyectos en las siguientes fases: entendimiento del negocio, entendimiento de los datos, preparación de los datos, modelado, evaluación y despliegue. Esta metodología asume que la mayoría de los procesos relacionados con la preparación de los datos, su modelado y evaluación se realizan de forma offline, lo que conlleva que si se detecta un fallo una vez que el modelo ha sido desplegado será un experto el que analizará el problema moviéndose de forma cíclica por las diferentes fases hasta dar con una solución que permita volver a entrenar y desplegar el modelo al entorno de producción de forma manual. Sin embargo, con el cambio de enfoque a un escenario en el que se espera una evolución de los datos, la preparación de estos, su modelado y su evaluación pasan a ser automatizadas y realizadas de forma continua tras el propio despliegue mediante la detección de cambios. Cuando se detecta un cambio, los anteriores datos de entrenamiento son descartados y el modelo es actualizado utilizando los datos más recientes.

Esta evolución y cambio a lo largo del tiempo se debe a las características específicas y únicas que diferencian a los datos por streaming de los datos por lotes utilizados en el contexto tradicional del aprendizaje automático. Los datos por streaming son por naturaleza incompatibles con el aprendizaje por lotes por numerosas razones, siendo la principal que en las aplicaciones tradicionales todos los datos están siempre disponibles al encontrarse almacenados por completo en memoria, mientras que, con el nuevo enfoque, los flujos de datos están disponibles de forma secuencial por un corto periodo de tiempo.

La siguiente tabla resume las características de los datos por lotes frente a los datos en streaming:

Tabla 1. Comparación de características de datos por lotes y datos en streaming

Características	Datos por lotes	Datos por streaming
Acceso	Aleatorio	Secuencial
Estado	Persistente	Transitorio
Procesamiento	Totalidad	Muestras
Disponibilidad	Constante	Limitada
Resultados	Precisos	Aproximados
Tamaño	Fijo	Sin límite
Estadísticos	Conocidos	Impredecibles

3. Desafíos asociados a los datos por streaming

El uso de entornos de datos por streaming implica tener en consideración una serie de desafíos que afectan a la hora de aplicar técnicas de minería de datos. Estos desafíos también afectan a los modelos de datos por lotes, pero en los siguientes apartados se centrará la problemática en el enfoque online.

3.1. Concept drift

El término concept drift hace referencia a un cambio en la distribución de los datos a lo largo del tiempo. Este efecto también puede recibir los nombres de dataset shift si se relaciona con detección de patrones o no estacionarios en el campo de tratamiento de señales (Žliobaitė et al., 2016). La detección de estos cambios supone uno de los mayores retos en los entornos streaming.

Su categorización se basa en la velocidad a la que se produce este cambio (Mittal & Kashyap, 2015), pudiendo ser abrupta cuando el cambio se produce rápidamente en un pequeño número de muestras, gradual o incremental cuando la transición ocurre a lo largo de numerosas muestras o recurrente cuando los cambios reaparecen después de un largo periodo.

Un detector de drift es un algoritmo que estima las posiciones de estos cambios para sustituir el modelo base y con ello mantener o mejorar el nivel de precisión del modelo desplegado. Estos detectores deben ser capaces de detectar lo antes posible el punto en el que la distribución ha cambiado, superar el ruido que se produce durante el cambio de modelo base y ser lo suficientemente rápidos computacionalmente para trabajar con la tasa de llegada de los flujos de datos (Wares et al., 2019).

Los detectores más utilizados son (Barros & Santos, 2018): Drift Detection Method (DDM), Early Drift Detection Method (EDDM), Adaptive Windowing (ADWIN), Statistical Test of Equal Proportions (STEPD), EWMA for Concept Drift Detection (ECDD),Paired Learners (PL), Sequential Drift (SeqDrift), Drift Detection Methods based on Hoeffding's Bounds (HDDM), Fast Hoeffding Drift DetectionMethod (FHDDM), Reactive Drift Detection Method (RDDM), Wilcoxon Rank Sum Test Drift Detector (WSTD) y Fisher Test Drift Detector (FTDD).

La detección de cambios es una tarea compleja debido al compromiso entre detectar cambios reales y evitar falsas alarmas. Las métricas (Bifet et al., 2013) que deben ser usadas para la elección de un detector de drift adecuado al problema a solucionar son:

- Media de tiempo entre falsas alarmas (MTFA). - Frecuencia con la que las falsas alarmas son lanzadas. Debe ser lo más grande posible.
- Tasa de falsa alarma (FAR). - Valor obtenido de 1/MTFA.
- Media de tiempo de detección (MTD). - Como de rápido se detecta la aparición de un drift. Debe ser lo más pequeño posible.
- Longitud promedio de ejecución (ARL). - Representa el tiempo en el que se produce una alarma tras un cambio de tamaño determinado.

No existe un único detector de drift que sea mejor que el resto en todas las situaciones. Además, los detectores más citados como son DDM, EDDM, ADWIN, ECDD y STEPD no siempre ofrecen los mejores resultados (Barros & Santos, 2018).

3.2. Datos desbalanceados

Se dice que los datos están desbalanceados cuando se tiene un mayor número de datos de un tipo que de otro (Alfhaid & Abdullah, 2021). Es una situación habitual en el mundo real, por ejemplo, en cualquier monitorización de fallos el número de estos siempre será muy inferior al número de muestras representadas por un funcionamiento correcto.

Los tres enfoques más utilizados para tratar esta situación son el muestreo, modelos conjuntos y los costes asociados (Ferreira et al., 2019). El muestreo puede ser de dos tipos, creación de muestras sintéticas (oversampling) o eliminación de muestras del tipo mayoritario (undersampling). El uso de modelos conjuntos busca la combinación de sus

decisiones de modo que para los tipos de datos que hay en menor número se puedan obtener mejores resultados. Por otro lado, los costes asociados utilizan como criterio el coste generado por clasificar mal cada uno de los tipos disponibles.

Los métodos de pre-procesado más utilizados se encuentran clasificados en: Clasificación binaria desbalanceada, Clasificación multiclase desbalanceada, Clasificación de multi-etiqueta y multi-instancia desbalanceada, regresión desbalanceada, Aprendizaje desbalanceado semi-supervisado y no supervisado, aprendizaje automatizado desde datos desbalanceados y big data desbalanceado (Alfhaid & Abdullah, 2021; Korycki & Krawczyk, 2021; Krawczyk, 2016).

3.3. Valores atípicos

La detección de anomalías (outliers) tiene como objetivo descubrir elementos que presentan una desviación significativa respecto al comportamiento esperado (Ahmad et al., 2017), por lo que resultan datos incoherentes. Su detección se realiza principalmente por dos razones (Kontaki et al., 2016): proceder a su eliminación para que no influencien a la precisión del modelo o detectarlos para su análisis ya que pueden suponer información que merece ser explorada.

Como se ha comentado en apartados anteriores, las aplicaciones de datos en streaming no pueden almacenar todos los datos recibidos debido al volumen y a la rapidez con los que son generados, por lo que la detección de valores atípicos se realiza sobre un número de muestras fijados por una ventana deslizante o haciendo uso de valores estadísticos de conjuntos previos (Ahmad et al., 2017; Chen et al., 2020; Duraj & Szczepaniak, 2021; Kontaki et al., 2016).

3.4. Dimensionalidad

La reducción de dimensionalidad se ha convertido en una técnica esencial debido a los escenarios compuestos por grandes conjuntos de datos. El objetivo que buscan estos procesos (Hu et al., 2018) es la reducción de la complejidad inherente al conjunto de datos, de modo que estos puedan procesarse más fácilmente por los modelos de minería de datos finalmente desplegados.

Las técnicas de reducción (Ramírez-Gallego et al., 2017) buscan eliminar las características/instancias redundantes o que generan ruido. De este modo, se obtienen modelos más precisos, rápidos y adaptables a la vez que se reduce la complejidad computacional (Abdulhammed et al., 2019).

3.5. Privacidad

Las limitaciones analizadas hasta ahora no pueden permitir que la privacidad y la sensibilidad comercial aportadas por los datos en streaming sea sacrificada cuando los flujos de información son compartidos con terceros (Sopaoglu & Abul, 2021). Para poder mantener esta privacidad es necesario hacer uso de métodos para anonimizar los flujos de datos, alterar los datos (Chamikara et al., 2018) y proteger los servicios de localización (Stephanie et al., 2022).

4. Frameworks y ejemplos de aplicaciones

Desde el punto de vista del desarrollo de software, un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. En el caso del aprendizaje online, los frameworks incluyen colecciones de algoritmos para realizar mediante aprendizaje online algoritmos de clasificación, regresión, clustering, reglas de asociación y *concept* drift, junto con herramientas para lanzar estos algoritmos bien de manera gráfica o a través de línea de comandos, y

adicionalmente poder integrar estos algoritmos en aplicaciones. También integran métricas para la evaluación de este tipo de algoritmos.

Uno de los primeros y más populares frameworks en el campo del aprendizaje online es MOA (**M**assive **O**nline **A**nalysis), lanzado en 2010 (Bifet et al., 2010). Se trata de un proyecto que emula el entorno y forma de trabajo de WEKA, de manera que proporciona un entorno de trabajo para probar de manera sencilla los distintos algoritmos que integra. Este framework tiene un enfoque principal como herramienta para ensayar distintos algoritmos y comparar resultados entre ellos. En este sentido destaca por la gran cantidad de algoritmos que incorpora. Sin embargo, su utilidad como librería que pueda ser empleada en una integración con una aplicación en tiempo real parece más discutida.

En 2015 aparece streamDM C++, un framework para procesamiento de streams basado en C++. Sin embargo, no ha tenido evolución desde entonces y los algoritmos que incorpora son muy limitados.

Dentro de la familia Spark surge streamDM for Spark Streaming como framework para el procesamiento masivo de streams. Se trata de una extensión de la API de Spark que permite el procesamiento de los streams dividiendo los datos en lotes que luego pueden ser procesados por Spark para generar los resultados. Este framework fue lanzado en 2016, pero al igual que el anterior, no ha tenido evolución desde entonces.

En entorno Python surge en 2018 scikit-multiflow (Montiel et al., 2018). El framework incorpora generadores de datos, algoritmos para aprendizaje supervisado, aprendizaje no supervisado y detección de concep drift. También en Python surge en 2020 Creme (Halford, 2020), que incorpora algunas novedades como guardar las características en diccionarios, lo que posibilita escalar datasets con millones de características. Sin embargo, el recorrido de estos dos frameworks es limitado, ya que se fusionan en 2020 en un nuevo framework denominado River (Montiel et al., 2021). Al igual que las anteriores, es un framework de propósito general que aborda problemas de clasificación, regresión, aprendizaje no supervisado y concept drift. Las técnicas implementadas tratan de gestionar de la manera más eficiente posible aspectos como la memoria y el tiempo de cómputo, dado el carácter de los procesos a aplicar. Este framework mantiene actualizaciones hasta el momento, aunque la documentación existente es bastante limitada, al igual que el número de algoritmos que implementa. Además, algunos experimentos realizados a modo comparativo entre MOA y River con el mismo conjunto de datos empleando el mismo algoritmo de aprendizaje arrojan resultados dispares, lo cual hace dudar de la implementación de algunos de estos algoritmos. Esto se ha podido comprobar al menos con el algoritmo AMRules (Duarte et al., 2016).

Tabla 2. Comparativa de frameworks de data streaming

Framework	Lanzamiento	Última actualización	Entorno	URL
MOA	2010	2021	Java	https://moa.cms.waikato.ac.nz/
StreamDM C++	2015	-	C++	http://huawei-noah.github.io/streamDM-Cpp/
StreamDM for Spark Streaming	2016	-	Spark	http://huawei-noah.github.io/streamDM/
Scikit-multiflow	2018	2020	Python	https://scikit-multiflow.github.io/
RIVER	2020	2021	Python	https://riverml.xyz/latest/

En cuanto a las aplicaciones en entornos reales, el número de referencias bibliográficas que se encuentran es muy reducido. Aunque el campo lleva en crecimiento continuo la última década, los ejemplos de aplicaciones concretas al sector industrial son muy reducidas. Las páginas webs de los distintos frameworks analizados no reportan en sus referencias ninguna aplicación en un entorno real industrial. Para realizar el análisis se ha optado por buscar las publicaciones que citan a los artículos que sirven para referenciar los frameworks y aplicar una estrategia de bola de nieve. Así por ejemplo el artículo correspondiente a MOA (Bifet et al., 2010) devuelve 1835 resultados. Es con diferencia el framework con más citaciones. River dispone de 31 citaciones. A continuación, se relacionan algunas de las aportaciones encontradas.

Entre las aplicaciones reportadas en publicaciones se pueden citar varios ejemplos dentro del sector de consumo energético, como la predicción de eficiencia energética (Martín et al., 2015), que emplea el algoritmo denominado Very Fast Decision Tree (VFDT). También se reportan aplicaciones para predecir eventos inusuales de consumo energético a partir de sensores domésticos (Fong et al., 2018). En este trabajo emplean una mejora del algoritmo VFDT.

También se han encontrado aplicaciones relativas a la calibración de sensores de calidad de aire (Bagkis et al., 2022). En este caso se utilizan algoritmos para el entrenamiento que tienen en cuenta el concept drift de los datos.

En el campo de la logística se han encontrado también ejemplos de aplicación. Así por ejemplo se desarrolla un sistema para procesar los datos logísticos en tiempo real y se compara con el sistema offline (AlShaer et al., 2019).

En el área de la ciberseguridad se han encontrado diversas aplicaciones también. Así por ejemplo el trabajo de Nakagawa et al. (2021) se enfoca a la detección de ataques en redes IoT domésticas, pero la aplicación podría ser fácilmente asimilable a entornos industriales.

Las técnicas de aprendizaje online han sido aplicadas también para abordar la supervisión y operación de parques eólicos (Pargmann et al., 2018). En este caso los datos procedentes de distintos sensores son encauzados a través de protocolo MQTT para el almacenamiento en la nube y el análisis online. En el campo de la agricultura también se han reportado aplicaciones, por ejemplo para la toma de decisiones en granjas (Wangen et al., 2021). Sin embargo, aunque los trabajos presentados abordan el tratamiento de datos como streams, realmente se limitan a la parte de captura y

almacenamiento de datos, no constituyendo verdaderos ejemplos de aplicaciones que utilicen modelos que empleen entrenamiento online.

5. Conclusiones

El número de dispositivos IoT y los datos que estos generan van a seguir aumentando en los próximos años. Resulta necesario seguir mejorando los modelos de datos por streaming para superar los desafíos a los que este tipo de minería de datos se debe enfrentar.

Los modelos de datos por streaming no tienen como objetivo sustituir a los métodos de modelización tradicionales. Estos modelos de datos por lotes van a seguir siendo necesarios para determinadas circunstancias, pero las necesidades generadas por el análisis en tiempo real hacen necesario adaptarse a este nuevo escenario.

De la prospectiva realizada se puede concluir que los frameworks existentes se encuentran en un estado poco maduro. Nuestra percepción parece encajar con las conclusiones alcanzadas por artículos de revisión recientes que resaltan que los esfuerzos de investigación deben orientarse hacia el desarrollo de frameworks y algoritmos escalables que se adapten al modo de computación de flujo de datos, la estrategia de asignación de recursos efectiva y los problemas de paralelización para hacer frente al tamaño y la complejidad cada vez mayores de los datos (Kolajo et al., 2019).

La aplicabilidad de la minería de datos por streaming puede adaptarse a múltiples sectores. Cualquier entorno personal o laboral con capacidad de generar algún tipo de dato medible de forma continua es candidato a poder implementar este tipo de soluciones. El sector industrial por sus características parece encajar bien en este contexto. Sin embargo, hasta el momento las aplicaciones reales reportadas son escasas.

6. Referencias

Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A. (2019). Features dimensionality reduction approaches for machine learning based network intrusion detection. *Electronics*, *8*(3), 322.

Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, *262*, 134–147.

Alfhaid, M. A., & Abdullah, M. (2021). Classification of Imbalanced Data Stream: Techniques and Challenges. *Artificial Intelligence*, *9*(2), 36–52.

AlShaer, M., Taher, Y., Haque, R., Hacid, M.-S., & Dbouk, M. (2019). IBRIDIA: A hybrid solution for processing big logistics data. *Future Generation Computer Systems*, *97*, 792–804.

Bagkis, E., Kassandros, T., & Karatzas, K. (2022). Learning Calibration Functions on the Fly: Hybrid Batch Online Stacking Ensembles for the Calibration of Low-Cost Air Quality Sensor Networks in the Presence of Concept Drift. *Atmosphere*, *13*(3), 416.

Barros, R. S. M., & Santos, S. G. T. C. (2018). A large-scale comparison of concept drift detectors. *Information Sciences*, *451*, 348–370.

Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., & Seidl, T. (2010). Moa: Massive online analysis, a framework for stream classification and clustering. *Proceedings of the First Workshop on Applications of Pattern Analysis*, 44–50.

Bifet, A., Read, J., Pfahringer, B., Holmes, G., & Žliobaitė, I. (2013). CD-MOA: Change detection framework for massive online analysis. *International Symposium on Intelligent Data Analysis*, 92–103.

Chamikara, M. A. P., Bertók, P., Liu, D., Camtepe, S., & Khalil, I. (2018). Efficient data perturbation for privacy preserving and accurate data stream mining. *Pervasive and Mobile Computing*, *48*, 1–19.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. *SPSS Inc*, *9*, 13.

Chen, L., Gao, S., & Cao, X. (2020). Research on real-time outlier detection over big data streams. *International Journal of Computers and Applications*, *42*(1), 93–101.

Duarte, J., Gama, J., & Bifet, A. (2016). Adaptive model rules from high-speed data streams. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *10*(3), 1–22.

Duraj, A., & Szczepaniak, P. S. (2021). Outlier Detection in Data Streams—A Comparative Study of Selected Methods. *Procedia Computer Science*, *192*, 2769–2778.

Ferreira, L. E. B., Gomes, H. M., Bifet, A., & Oliveira, L. S. (2019). Adaptive random forests with resampling for imbalanced data streams. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–6.

Fong, S., Li, J., Song, W., Tian, Y., Wong, R. K., & Dey, N. (2018). Predicting unusual energy consumption events from smart home sensor network by data stream mining with misclassified recall. *Journal of Ambient Intelligence and Humanized Computing*, *9*(4), 1197–1221.

Halford, M. (2020, March 26). *Machine learning for streaming data with creme*. Medium. https://towardsdatascience.com/machine-learning-for-streaming-data-with-creme-dacf5fb469df

Hu, X., Zhou, P., Li, P., Wang, J., & Wu, X. (2018). A survey on online feature selection with streaming features. *Frontiers of Computer Science*, *12*(3), 479–493.

Kolajo, T., Daramola, O., & Adebiyi, A. (2019). Big data stream analysis: A systematic literature review. *Journal of Big Data*, *6*(1), 1–30.

Kontaki, M., Gounaris, A., Papadopoulos, A. N., Tsichlas, K., & Manolopoulos, Y. (2016). Efficient and flexible algorithms for monitoring distance-based outliers over data streams. *Information Systems*, *55*, 37–53.

Korycki, Ł., & Krawczyk, B. (2021). Concept drift detection from multi-class imbalanced data streams. *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 1068–1079.

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221–232.

Martín, E. G., Lavesson, N., & Grahn, H. (2015). Energy efficiency in data stream mining. *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 1125–1132.

Mittal, V., & Kashyap, I. (2015). Online methods of learning in occurrence of concept drift. *International Journal of Computer Applications*, *117*(13).

Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H. M., Read, J., & Abdessalem, T. (2021). *River: Machine learning for streaming data in Python*.

Montiel, J., Read, J., Bifet, A., & Abdessalem, T. (2018). Scikit-multiflow: A multi-output streaming framework. *The Journal of Machine Learning Research*, *19*(1), 2915–2914.

Nakagawa, F. H., Junior, S. B., & Zarpelão, B. B. (2021). Attack Detection in Smart Home IoT Networks using CluStream and Page-Hinkley Test. *2021 IEEE Latin-American Conference on Communications (LATINCOM)*, 1–6.

Pargmann, H., Euhausen, D., & Faber, R. (2018). Intelligent big data processing for wind farm monitoring and analysis based on cloud-technologies and digital twins: A quantitative approach. *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 233–237.

Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., & Herrera, F. (2017). A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing*, *239*, 39–57.

Sopaoglu, U., & Abul, O. (2021). Classification utility aware data stream anonymization. *Applied Soft Computing*, *110*, 107743.

Stephanie, V., Chamikara, M. A. P., Khalil, I., & Atiquzzaman, M. (2022). Privacy-preserving location data stream clustering on mobile edge computing and cloud. *Information Systems*, *107*, 101728.

Wangen, S. R., Zhang, F., Fadul-Pacheco, L., da Silva, T. E., & Cabrera, V. E. (2021). Improving farm decisions: The application of data engineering techniques to manage data streams from contemporary dairy operations. *Livestock Science*, *250*, 104602.

Wares, S., Isaacs, J., & Elyan, E. (2019). Data stream mining: Methods and challenges for handling concept drift. *SN Applied Sciences*, *1*(11), 1–19.

Žliobaitė, I., Pechenizkiy, M., & Gama, J. (2016). An overview of concept drift applications. *Big Data Analysis: New Algorithms for a New Society*, 91–114.

Comunicación alineada con los Objetivos de Desarrollo Sostenible