Predicción conforme para enriquecer los modelos dinámicos de clasificación de aplicaciones maliciosas en escenarios de deriva conceptual

David Escudero García
RIASC. Universidad de León
Edificio MIC. Campus de Vegazana s/n 24071 León (España)
descg@unileon.es

Noemí DeCastro-García
Departamento de Matemáticas. Universidad de León.
Campus de Vegazana s/n 24071 León (España)
ncasg@unileon.es

***Resumen*—El aprendizaje automático es uno de los principales enfoques utilizados para la detección de *malware*, ya permite obtener modelos más adaptables que las soluciones basadas en firmas. Uno de los principales desafíos en la aplicación del aprendizaje automático en la detección de *malware* es la presencia de la deriva conceptual, que es un cambio en la distribución de los datos a lo largo del tiempo. Para abordar la deriva, es común aplicar modelos *online* que se pueden actualizar dinámicamente. Sin embargo, estos modelos requieren nuevas instancias etiquetadas con las que actualizar el modelo. Por lo general, las etiquetas disponibles son escasas, costosas de obtener y no están disponibles de forma inmediata, por lo que la construcción de un modelo efectivo es complicada. En este trabajo, proponemos la construcción de modelos enriquecidos con predicción conforme, que disponen de garantías estadísticas en la predicción, para obtener pseudoetiquetas fiables que puedan utilizarse para actualizar el modelo y compensar la ausencia de datos etiquetados. Los resultados muestran que la predicción conforme puede mejorar el rendimiento predictivo, aunque la mejora depende del modelo base utilizado.**

***Index Terms*—aprendizaje automático, malware, predicción conforme, deriva conceptual**

Tipo de contribución: *Investigación original*

I. Introducción

La clasificación del malware es un problema que ha recibido mucha atención debido al aumento anual de amenazas [1]. Antivirus y otros métodos basados en firmas permiten identificar rápidamente muestras sospechosas, pero requieren que la muestra haya sido identificada previamente, lo que limita su aplicación a nuevas amenazas. Por ello, los esfuerzos de investigación se dirigen a abordar este problema proponiendo diferentes estrategias de identificación de *malware*. Una de las soluciones más comunes en la literatura es la aplicación de técnicas de aprendizaje automático. No obstante, la aplicación de aprendizaje automático tiene ciertas limitaciones.

Una de las principales suposiciones realizadas en el aprendizaje automático es que la distribución de probabilidad de los datos es estacionaria, es decir, permanece constante en el tiempo. Esta situación no se da en entornos realistas [1]. Nuevas familias de *malware* aparecen o modifican su comportamiento; o incluso reaparecen tras un tiempo. Estos cambios en la distribución de los datos se conocen como deriva conceptual [2] y comprometen la eficacia de los modelos, ya que la distribución de los nuevos datos sobre los que se predice no es la misma que la de los datos que se utilizaron para entrenar el modelo. Por ejemplo, un modelo entrenado principalmente con muestras maliciosas de tipo *botnet* probablemente tendría problemas para detectar *ransomware* con exactitud.

En la literatura, se han propuesto diferentes enfoques para afrontar estos cambios en la distribución de los datos [3], [4]. Buena parte de estos trabajos se centran en la aplicación de modelos *online*, que pueden ser actualizados de forma dinámica sin necesidad de reentrenar el modelo desde cero. Muchos trabajos aplican una metodología *precuencial*, donde se asume disponibilidad de etiquetas para actualizar modelos. En [3] se aplica una estrategia de adaptación pasiva mediante procesamiento por lotes, manteniendo un conjunto de modelos; el de peor rendimiento es sustituido cuando se procesa cierto número de instancias. Esta técnica mejora la tasa de acierto en un promedio de 0.3 en comparación con modelos estáticos. Otros trabajos aplican una estrategia de adaptación activa frente a la deriva conceptual, incluyendo algoritmos de detección de deriva. Aunque existen trabajos que con esta aproximación aportan mejoras de 0.15, [4], la mayoría de estos algoritmos también requieren etiquetas para los nuevos datos. Otros enfoques parten de la utilización de *pseudoetiquetas*. En [5], se propone un modelo que utiliza un ensamblado de modelos lineales y *pseudoetiquetas* generadas a partir de predicciones anteriores. Aunque este método mejora la tasa de acierto en 0.1 en comparación con modelos tradicionales, carece de un mecanismo para determinar la fiabilidad de las *pseudoetiquetas*, lo que en general suele causar una degradación del rendimiento en el tiempo, a medida que el modelo aprende de ejemplos mal etiquetados [6].

Los trabajos citados presentan resultados competitivos, pero su principal limitación es que es necesario disponer de etiquetas (determinar si la muestra es maliciosa o no, o a qué familia de malware pertenece) para actualizar el modelo. La necesidad de etiquetado manual limita su aplicabilidad [2] ya que requiere de conocimiento experto y es costoso, por lo que la proporción de datos que se podrán etiquetar será muy pequeña y tendrá una cierta demora antes de que se pueda utilizar para actualizar el modelo.

Con el objetivo de obtener etiquetas de forma rápida y fiable, en este trabajo planteamos la aplicación de predicción conforme. La predicción conforme es un paradigma que permite controlar, de forma estadísticamente rigurosa, la incertidumbre de las predicciones de un predictor, sin imponer ninguna suposición sobre el modelo o la distribución

de los datos [7]. Nuestra hipótesis es que aplicando predicción conforme, se pueden obtener *pseudoetiquetas* fiables con las que actualizar el modelo, mejorando así su rendimiento, aún cuando no se disponga de suficientes ejemplos etiquetados por un experto. Además, sería posible evitar el problema de la degradación del modelo por la acumulación de errores en las pseudoetiquetas [6]. No existen muchos trabajos que apliquen predicción conforme al problema de la detección de malware. En [8], se aplica predicción conforme en combinación con un modelo *Random Forest*. Los resultados muestran que las garantías respecto a la tasa de error de la predicción conforme se cumplen en la práctica. No obstante, el trabajo no tiene cuenta la deriva conceptual. Por otro lado, el trabajo en [9], aplica aprendizaje conforme para detectar predicciones poco fiables (por ejemplo instancias sometidas a deriva). Una vez detectada, la instancia se podría analizar manualmente o enviar a otros análisis automatizados. Los resultados muestras que la diferencia en el F1-score entre las predicciones fiables y no fiables según el algoritmo puede llegar a ser mayor a 0.5, por lo que resulta eficaz en la detección de predicciones poco fiables.

En esta investigación se han aplicado algoritmos *online* a un conjunto de datos de detección de *malware*, y se han comparado con la implementación de los mismos pero enriqueciendo los datos con predicción conforme. Se ha analizado el comportamiento de los segundos, y establecido una comparativa de ganancia con los primeros. Todos los experimentos se llevan a cabo utilizando el conjunto de datos KronoDroid [10], que incluye una muestra representativa de aplicaciones Android benignas y maliciosas de 2008 a 2020, haciendo posible evaluar el rendimiento de los modelos en una situación relativamente realista, en la que el comportamiento de las aplicaciones cambia en el tiempo. Se han incluido diferentes versiones del conjunto de datos, variando la proporción de etiquetas verdaderas disponibles, y de muestras maliciosas.

Este documento se organiza de la siguiente manera: en la sección 2 se incluye una breve descripción de la predicción conforme. En la sección 3, se incluyen todos los detalles de la experimentación. En la sección 4, se discuten los resultados. Finalmente, se incluyen las conclusiones y las referencias.

II. Predicción conforme

En el caso de problemas de clasificación, la predicción conforme se utiliza para producir, en lugar de una única predicción de clase a la que pertenece una instancia x_j, un conjunto de clases (el conjunto de predicción) $\mathcal{C}(x_j)$ tal que la clase correcta de la instancia y_j pertenece a $\mathcal{C}(x_j)$ con una determinada probabilidad, que el usuario puede fijar. Para un problema multiclase, una mayor cardinalidad del conjunto de predicción implica una menor certeza del modelo en la predicción. En el caso más extremo, el conjunto de predicción puede estar vacío, lo que significa que la predicción conforme no puede garantizar que ninguna de las clases sea la correcta de acuerdo con el modelo con una suficiente probabilidad.

La predicción conforme requiere de un conjunto de datos de calibración $D_{calib} = \{(x_i, y_1), \ldots, (x_{nc}, y_{nc})\}$ diferente al usado para entrenar el modelo, donde las instancias x_i son vectores de características y y_i es la clase asociada a la instancia. Por ejemplo, para un problema de detección de malware, x_i podría contener información como el número de llamadas a diferentes funciones de la API del sistema y y_i podría indicar si la muestra correspondiente es benigna o maliciosa. Utilizando D_{calib} y las predicciones del modelo, la probabilidad de que la verdadera clase de la instancia esté contenida en el conjunto de predicción será aproximadamente $1 - \alpha$, donde α es un nivel de riesgo que el usuario puede fijar.

En primer lugar, la predicción conforme requiere establecer una *métrica de disconformidad* s_i, que modela la incertidumbre de la predicción del modelo $\hat{f}$ sobre la instancia x_i. Es posible elegir diferentes métricas utilizando la probabilidad que el propio modelo asigna a su predicción o la distancia a otras instancias de la misma clase. En este trabajo fijamos $s_i = 1 - \hat{f}(x_i)_{y_i}$, es decir, 1 menos la probabilidad que el modelo asigna a la verdadera clase y_i de la instancia x_i. Consecuentemente, si el modelo realiza una predicción incorrecta, $\hat{f}(x_i)_{y_i}$ tendrá una magnitud pequeña y s_i estará cercana a su valor máximo, que es 1. Durante la calibración, se computa $S = (s_1, \ldots, s_{nc})$ (la disconformidad de todos los elementos de D_{calib}) y se fija $\hat{q} = Cuantil(S, \frac{\lceil (nc+1)(1-\alpha) \rceil}{nc})$. $\hat{q}$ es esencialmente el cuantil $1 - \alpha$ de S con una pequeña corrección. Para una nueva instancia x_j sobre la que predecir, el conjunto de predicción es $\mathcal{C}(x_j) = \{y : \hat{f}(x_j)_y \geq 1 - \hat{q}\}$. Es decir, todas las clases para las cuales el modelo asigna una probabilidad lo suficientemente alta.

La única suposición que la predicción conforme requiere para ser válida es la *intercambiabilidad*, que exige que cualquier permutación de los datos originales presente la misma distribución. Si los datos presentan deriva conceptual, esta suposición puede no cumplirse, por lo que las predicciones pueden perder validez estadística. Para determinar en qué casos son fiables, se puede hacer uso de dos métricas: la *confianza* y *credibilidad* de la predicción.

$confianza = \sup\{1 - \alpha : |\mathcal{C}(x_j)| = 1\}$

$credibilidad = \sup\{\alpha : |\mathcal{C}(x_j)| = 0\}$

La confianza indica cómo de distinguible es x_j de instancias de otras clases y la credibilidad cómo de similar es x_j a instancias de la clase que se le ha asignado. Estas dos magnitudes se pueden utilizar para seleccionar las predicciones más fiables.

III. Sección experimental

En esta sección, se describen los conjuntos de datos utilizados en la experimentación, así como los modelos de aprendizaje utilizados, y el protocolo de evaluación.

III-A. Protocolo de investigación

Como hemos mencionado, la experimentación se basa en estudiar el comportamiento de la predicción conforme como enriquecimiento en un escenario de clasificación mediante modelos de aprendizaje automático *online*.

Las preguntas de investigación son:

1. ¿Cómo funcionan los modelos basados en predicción conforme?
2. ¿Existen diferencias entre el comportamiento obtenido por los modelos enriquecidos con predicción conforme en función del modelo base con el que trabajan, las proporciones de etiquetas disponibles o la proporciones de *malware*?

3. ¿Hay mejoras en el ajuste que aportan los modelos basados en predicción conforme en comparación con los modelos *online*?
4. En caso afirmativo, ¿están las mejoras relacionadas con la proporción de *malware*, de etiquetas, o con el modelo *online* con el que comparamos?

III-B. Modelos de aprendizaje

Los modelos de aprendizaje utilizados en este trabajo son tres: *Adaptive Hoeffding Tree* (AHT) [11], *Adaptative Random Forest* (ARF) [12] y *Logistic Regression* (LR) [13]. Se utiliza la implementación de la librería river [14] del lenguaje de programación Python. Se seleccionan estos tres modelos por ser comúnmente usados en la literatura de procesado de datos *online*.

El rango de hiperparámetros utilizado para los modelos se muestra en la Tabla I.

Tabla I
MODELOS Y SUS HIPERPARÁMETROS

Modelo	Hiperparámetro	Rango
AHT	Criterio de divisón	$[gini, hellinger, information\ gain]$
	Profundidad	$[8, 10, 12, 14]$
ARF	Criterio de divisón	$[gini, hellinger, information\ gain]$
	Número de modelos	$[10, 25, 50, 100]$
	Profundidad	$[8, 10, 12, 14]$
LR	Regularización L2	$[0, 0.001, 0.025, 0.05, 0.1, 0.125]$

III-C. Conjuntos de datos

En este trabajo se han utilizado el conjunto de datos KronoDroid [10]. Este conjunto de datos incluye aplicaciones maliciosas y benignas de Android del periodo 2008-2020. Contiene 76720 muestras, 40936 maliciosas y 35784 benignas. El conjunto de datos tiene 489 variables obtenidas a través de análisis estático (permisos e intents) y dinámico (llamadas a API).

Para poder realizar la experimentación, se ha modificado este conjunto de datos, variando la proporción real de muestras maliciosas (0.05,0.1,0.5336), y la proporción de instancias etiquetadas con las que actualizar los modelos ($0, 0.05$ y 0.1). Por lo tanto, se han obtenido 9 subconjuntos de datos. Respecto a la proporción de muestras maliciosas, el valor 0.5336 es el presente en Kronodroid. En entornos realistas, la proporción de muestras maliciosas tiende a ser inferior a la proporción de muestras benignas [1] por lo que en los experimentos reducimos esa proporción para evaluar el rendimiento con datos desbalanceados. Por otro lado, fijamos una proporción de datos etiquetados relativamente limitada, respetando la suposición de que la mayor parte de los datos no dispondrán de etiquetas para actualizar el modelo. Además, para simular que las etiquetas no estarán disponibles de forma inmediata, fijamos una demora de 2000 instancias para la actualización del modelo. Es decir, si una instancia está etiquetada, no se utilizará para actualizar el modelo hasta que no haya realizado un predicción sobre las 2000 instancias posteriores.

Además, se utilizarán dos conjuntos de datos para los análisis. El primero, D_1, incluye los resultados obtenidos de aplicar los modelos online enriquecidos con predicción conforme a los subconjuntos de datos obtenidos desde $KronoDroid$ variando las proporciones de malware y de etiquetas. Las variables de respuesta que incluye este conjunto de datos son el coeficiente de correlación de Matthews o MCC ([15]), la sensibilidad y la especificidad. Se han realizado cinco iteraciones para cada modelo y subconjunto, por lo tanto, el conjunto de datos tiene 135 filas. El segundo, D_2, tiene una estructura idéntica a D_1, pero incluye los resultados obtenidos de aplicar los modelos online sin predicción conforme.

III-D. Protocolo de creación de modelos

El procedimiento que se sigue es el siguiente:

- Primero, se ajusta el conjunto de datos según la proporción de *malware* seleccionada. Esto se lleva a cabo con una muestra aleatoria sobre las instancias maliciosas, para reducirlas al número fijado.
- Se utiliza el 10 % de datos más antiguos del conjunto de datos (las instancias se procesan respetando el orden cronológico) para entrenar los modelos. Para los modelos conformes, se utiliza el 30 % de estos datos como conjunto de calibración.
- El modelo realiza predicciones sobre las instancias una a una. Si la instancia tiene disponible una etiqueta, se utilizará para actualizar el modelo tras la demora fijada.
- En el caso de que se utilice un modelo con predicción conforme, se evalúa la predicción. Si el nivel de confianza es superior a 0.95 y la credibilidad a 0.9 (estos valores se fijan en base a experimentos preliminares) la predicción se considera fiable y se toma como *pseudoetiqueta* para actualizar el modelo.
- Se computan métricas en base a las predicciones del modelo.

III-E. Análisis

Las variables respuesta analizadas son tres: el MCC [16] que se define de la siguiente manera:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (1)$$

donde TP, TN, FP, y FN denotan verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, respectivamente. La sensibilidad y la especificidad se denotan por S y E. En los experimentos se considera como clase positiva que la muestra sea maliciosa. La sensibilidad se refiere a la capacidad del modelo para identificar correctamente los casos positivos, minimizando los falsos negativos; la especificidad es la capacidad de detectar verdaderos negativos, minimizando falsos positivos. La preferencia por sensibilidad o especificidad dependerá del caso de uso. Utilizamos el MCC porque es una métrica más informativa que la tasa de acierto o F1-score en conjuntos de datos con un importante desbalanceo.

El resto de análisis llevados a cabo se enumeran a continuación:

1. El primer análisis estadístico realizado es la prueba de normalidad de Kolmogorov–Smirnov con la corrección de Lilliefors. Al salir significativa, los análisis inferenciales serán no paramétricos y se utilizará la mediana como medida de centralización.
2. Se realiza un estudio descriptivo de D_1.

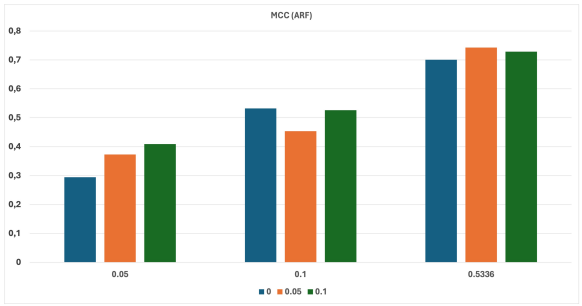

Figura 1. Aplicación de predicción conforme junto con ARF. Eje X: Proporción de *malware*.Eje Y: MCC

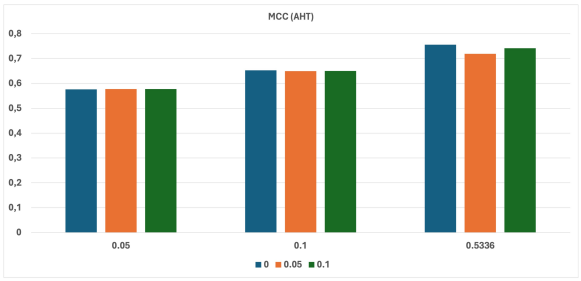

Figura 2. Aplicación de predicción conforme junto con AHT. Eje X: Proporción de *malware*.Eje Y: MCC

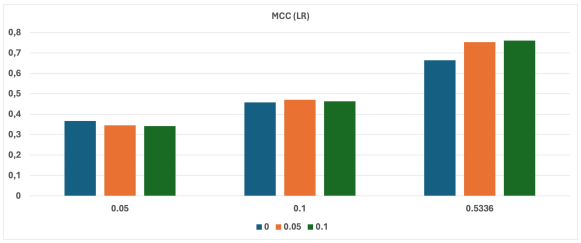

Figura 3. Aplicación de predicción conforme junto con LR. Eje X: Proporción de *malware*.Eje Y: MCC

3. Debido a que la aplicación de los modelos enriquecidos con predicción conforme modifica el conjunto de datos, los conjuntos D_1 y D_2 se consideran independientes. Para realizar la comparación entre ellos y deducir si las posibles diferencias detectadas son significativas, se ha aplicado el test U de Mann-Whitney para 2 muestras independientes. Si han existido diferencias, utilizamos un estudio descriptivo para determinar con qué conjunto obtenemos mejores resultados de clasificación. Se realizarán de manera general y segmentando los conjuntos de datos por las variables de las que queremos medir el efecto: tipo de modelo base, proporción de *malware* y proporción de etiquetas.
4. Para estudiar el efecto que tienen los grupos en aquellos casos en los que sí aparecen diferencias significativas, se ha utilizado la d de Cohen mediante la estandarización de las diferencias de medias. Para interpretar el índice, se utiliza la escala descrita en Eq. 2 ([17]):

$$\text{Efecto} = \begin{cases} \text{Pequeño} & \text{si } d \in [0, 0.3] \\ \text{Medio} & \text{si } d \in [0.5, 0.8] \\ \text{Grande} & \text{si } d > 0.8 \end{cases} \quad (2)$$

5. El último análisis que se realiza es el estudio de las posibles correlaciones entre las ganancias obtenidas en MCC, sensibilidad y especificidad, y los resultados en esas tres variables que se obtenían por los modelos enriquecidos con predicción conforme y los modelos *online* sin enriquecimiento. Como todas las variables son cuantitativas, se ha calculado el coeficiente de correlación de Pearson junto con el p-valor asociado. Cabe recordar que el coeficiente de correlación de Pearson toma valores entre -1 y 1. Cuando más se acerca a la unidad, hay más correlación. El signo representa si la relación es directa o inversa.

Todos los análisis inferenciales se realizan con $\alpha = 0.05$.

IV. Resultados y discusión

Los resultados se organizan en función de las preguntas de investigación planteadas.

IV-A. Comportamiento de los modelos con predicción conforme

Se incluyen en las Figuras 1, 2 y 3 los ajustes de MCC obtenidos por los modelos enriquecidos con predicción conforme en cada uno de los subconjuntos de datos.

Como podemos observar, el algoritmo AHT es el que mejor resultados aporta en la mayoría de los casos ($MCC_{ARF} = 0.5291$, $MCC_{AHT} = 0.6553$ y $MCC_{LR} = 0.5137$), siendo únicamente superado por LR en los casos de mayor proporción de malware y etiquetas (53.56 % de malware y 5 % y 10 % de proporción de etiquetas, respectivamente). Posiblemente, esta ventaja se deba a la mayor regularización que AHT incluye para hacer frente a la deriva conceptual [11]: una política de actualización más conservadora previene el impacto de cambios estacionarios en los datos, pero puede limitar el aprendizaje en condiciones más favorables. Cabe destacar que, para todos los algoritmos y modelos construidos, los mejores ajustes se han obtenido con la mayor proporción de malware. Esto no resulta sorprendente, ya que los modelos *online* son particularmente vulnerables al desbalanceo de los datos [18]. No hay tendencias claras en el caso de la proporción de datos etiquetados. Esto se puede achacar a que se seleccionan aleatoriamente: las instancias cuya clasificación sea más ambigua serán más informativas para el modelo, pero es posible que otra menos informativas resulten etiquetadas.

Se incluyen en las Figuras 4, 5 y 6 la sensibilidad obtenida por los modelos con predicción conforme en cada uno de los subconjuntos de datos de $Kronodroid$ ($S_{ARF} = 0.4584$, $S_{AHT} = 0.5640$ y $S_{LR} = 0.7670$). La sensibilidad presenta una tendencia creciente en términos de proporción de malware, llegando a obtener valores mayores de 0.80 para la proporción más alta de aplicaciones maliciosas. Esta tendencia es menos visible para el algoritmo de LR. Sin embargo, la mínima sensibilidad obtenida en este caso es de 0.72, por lo tanto, el rango de crecimiento es menor. En relación con la proporción de etiquetas, el comportamiento de la sensibilidad es homogéneo.

Se incluyen en las Figuras 7, 8 y 9 la especificidad obtenida por los modelos enriquecidos con predicción conforme en cada uno de los conjuntos de datos ($E_{ARF} = 0.9559$,

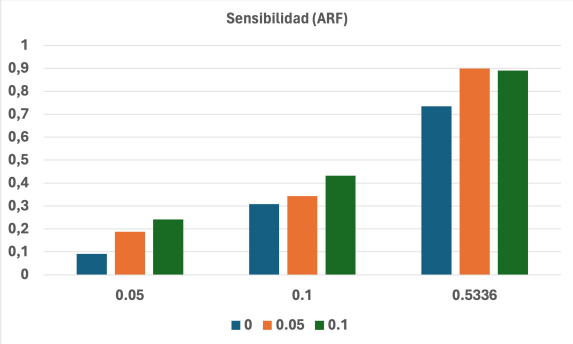

Figura 4. Aplicación de predicción conforme junto con ARF. Eje X: Proporción de *malware*.Eje Y: Sensibilidad

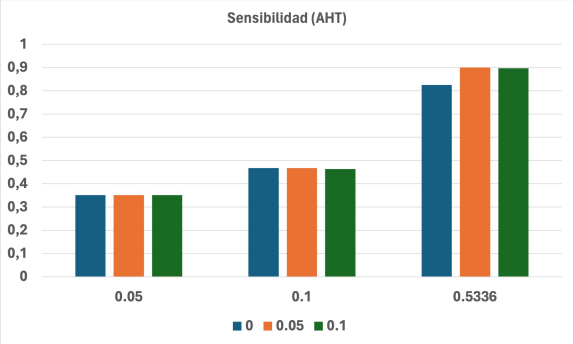

Figura 5. Aplicación de predicción conforme junto con AHT. Eje X: Proporción de *malware*.Eje Y: Sensibilidad

$E_{AHT} = 0.9566$ y $E_{LR} = 0.8717$).Como puede observarse, el nivel de especificidad alcanzado es alto y mayor que la sensibilidad, resultando superior para ARF y AHT. En el caso de las proporciones de *malware* 0.05 y 0.1, una mayor especificidad es el resultado esperado: la mayoría de instancias con las que se entrena el modelo serán benignas. Para la proporción de *malware* 0.5336, esta diferencia se justifica porque las instancias benignas están menos afectadas por la deriva conceptual [4], [3]. Aún con datos balanceados y etiquetas disponibles, cabe esperar más errores en la predicción de instancias maliciosas. Para los resultados de especificidad de LR, prácticamente no hay diferencias encontradas entre los niveles de maliciosidad del conjunto de datos. Sin embargo, para ARF y AHT, sí que existe un mejor ajuste de especificidad en los niveles bajos de proporción de *malware* y de etiquetado. En general, un incremento de la sensibilidad tiende a incrementar el número de falsos positivos (y por lo tanto a reducir la especificidad). En el caso de AHT y ARF, los mecanismos de adaptación a la deriva introducen cambios en la estructura del modelo (eliminar modelos en ARF, poda de ramas en AHT) para asegurar que la nueva instancia se clasifica correctamente. Esto conduce a un cierto sesgo. Aunque LR también se actualiza, no dispone de estos mecanismos, por lo que el impacto sobre la especificidad no es tan marcado.

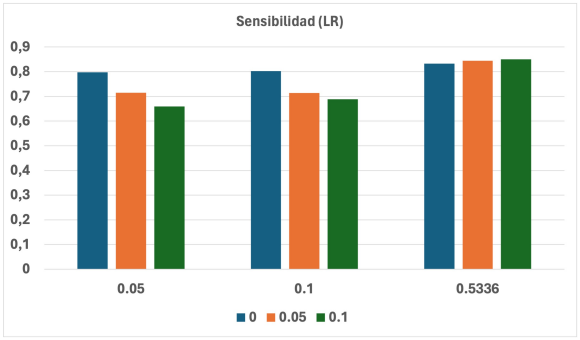

Figura 6. Aplicación de predicción conforme junto con LR. Eje X: Proporción de *malware*.Eje Y: Sensibilidad

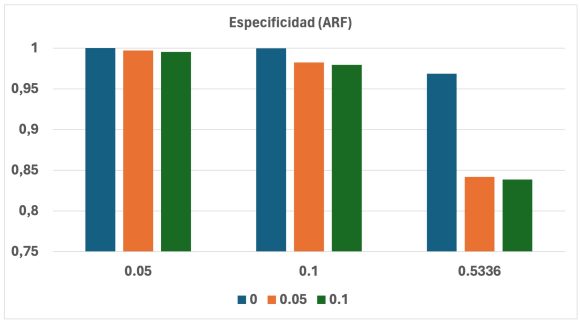

Figura 7. Aplicación de predicción conforme junto con ARF. Eje X: Proporción de *malware*.Eje Y: Especificidad

IV-B. Comparativa entre la aplicación de los algoritmos con y sin predicción conforme

Aunque vemos que existen tendencias prometedoras en el comportamiento de la aplicación de los modelos enriquecidos con predicción conforme, resulta de especial interés comprobar si su uso aporta alguna mejora con respecto a la aplicación de los modelos base. Para poder extraer conclusiones, se ha realizado un contraste de hipótesis cuyos resultados se incluyen en la Tabla II.

Tabla II
RESULTADOS DEL TEST U DE MANN- WHITNEY PARA COMPARAR LOS MODELOS BASE VS MODELOS CON PREDICCIÓN CONFORME

Variable	U de Mann-Whitney	ρ-valor	Mejor modelo	d-Cohen
MCC	$Z = -4.07$	$\rho < .001$	Conformes	0.595
S	$Z = -2.171$	0.029	Conformes	0.309
E	$Z = -0.9812$	0.3268		0.023

Como podemos observar, existen diferencias significativas en el MCC y en la sensibilidad, aunque no para la especificidad. En el caso del MCC y la sensibilidad, el efecto puede considerarse mediano. El mayor impacto en el MCC se debe a que es una métrica relativamente exigente, por lo que se ve afectada en mayor medida por diferencias en tasas de predicciones correctas.

A continuación analizamos los resultados más en profundidad, segmentando D_1 y D_2 por modelo, proporción de malware y proporción de etiquetas. Los resultados según el algoritmo base se incluyen en la Tabla III. Se puede observar que AHT es el modelo que más sale beneficiado por la apli-

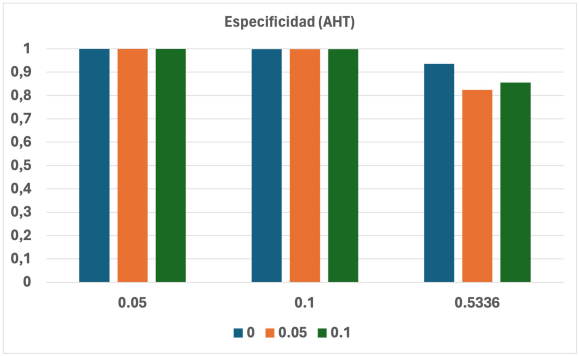

Figura 8. Aplicación de predicción conforme junto con AHT. Eje X: Proporción de *malware*.Eje Y: Especificidad

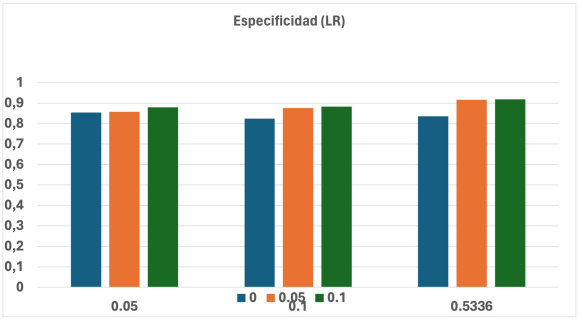

Figura 9. Aplicación de predicción conforme junto con LR. Eje X: Proporción de *malware*.Eje Y: Especificidad

cación de predicción conforme, con mejoras estadísticamente significativas para las tres métricas. También existen mejoras significativas para la sensibilidad en LR (MCC y especificidad mejoran, pero no de forma estadísticamente significativa), pero en ninguna métrica para ARF.

Tabla III
RESULTADOS DEL TEST U DE MANN- WHITNEY PARA COMPARAR LOS MODELOS ONLINE VS MODELOS CON PREDICCIÓN CONFORME SEGÚN ALGORITMO BASE

Variable	Estadístico	ρ-valor	Mejor modelo
MCC_{ARF}	$Z = -1.819$	.069	
S_{ARF}	$Z = -1.141$	.254	
E_{ARF}	$Z = 0.891$	.373	
MCC_{AHT}	$Z = -5.003$	$< .001$	Conformes
S_{AHT}	$Z = -2.489$	.012	Conformes
E_{AHT}	$Z = -2.788$	.005	Conformes
MCC_{LR}	$Z = -1.327$	.185	
S_{LR}	$Z = -2.489$	.012	Conformes
E_{LR}	$Z = -1.868$	.062	

Los resultados de la comparativa cuando segmentamos los conjuntos de datos por proporción de malware se incluyen en la Tabla IV. En este caso, se puede observar que los modelos enriquecidos con predicción conforme mejoran significativamente el MCC para las proporciones de malware 0.05 y 0.1; para 0.05, esta mejora afecta principalmente a la especificidad y para 0.1 a la sensibilidad. En la proporción de malware de 0.5336, los modelos con predicción conforme no proporcionan ninguna mejora y, en algunos casos, empeoran el rendimiento.

Los resultados de la comparativa cuando segmentamos los conjuntos de datos por proporción de etiquetas se incluyen en la Tabla V. En este caso, únicamente se producen ganancias significativas en MCC y sensibilidad en el caso en el que no hay etiquetas disponibles.

Tabla IV
RESULTADOS DEL TEST U DE MANN- WHITNEY COMPARANDO LOS MODELOS ONLINE VS MODELOS CON PREDICCIÓN CONFORME SEGÚN PROPORCIÓN DE MALWARE

Variable	Estadístico	ρ-valor	Mejor modelo
$MCC_{0.05}$	$Z = -5.108$	$< .001$	Conformes
$S_{0.05}$	$Z = -1.843$	.065	
$E_{0.05}$	$Z = -2.598$	.009	Conformes
$MCC_{0.1}$	$Z = -5.999$	$< .001$	Conformes
$S_{0.1}$	$Z = -2.489$	.012	Conformes
$E_{0.1}$	$Z = -1.4404$	.150	
$MCC_{0.5336}$	$Z = 1.533$	.126	
$S_{0.5336}$	$Z = 0.112$	.913	
$E_{0.5336}$	$Z = 0.689$	.492	

Tabla V
RESULTADOS DEL TEST U DE MANN- WHITNEY COMPARANDO LOS MODELOS ONLINE VS MODELOS CON PREDICCIÓN CONFORME SEGÚN PROPORCIÓN DE ETIQUETAS

Variable	Estadístico	ρ-valor	Mejor modelo
MCC_0	$Z = -5.338$	$< .001$	Conformes
S_0	$Z = -3.631$	$< .001$	Conformes
E_0	$Z = -0.504$	.615	
$MCC_{0.05}$	$Z = -1.638$	.102	
$S_{0.05}$	$Z = -0.496$	.622	
$E_{0.05}$	$Z = -1.061$	.290	
$MCC_{0.1}$	$Z = -0.544$	.588	
$S_{0.1}$	$Z = -0.181$	.859	
$E_{0.1}$	$Z = -0.117$	.910	

En general, la aplicación de modelos conformes resulta útil principalmente en situaciones de alto desbalanceo de los datos, pero sólo para AHT la magnitud de la mejora es notable. En el resto de modelos no hay mejora significativa.

IV-C. Estudio de la ganancia y de las correlaciones existentes

En la Tabla VI se incluyen los estadísticos descriptivos de las ganancias en términos de MCC, sensibilidad y especificidad. En general, los modelos con predicción conforme mejoran el rendimiento respecto a los modelos base, aunque como se ha visto en la sección anterior, las ganancias no son estadísticamente significativas en ciertos casos. En la Tabla VII se muestran las ganancias medias segmentadas por modelo y proporciones de malware y datos etiquetados. Se puede observar que AHT es el modelo con mayores ganancias, seguido de ARF y LR.

A continuación, entramos a valorar las posibles correlaciones existentes entre las ganancias y el resto de variables. El estudio de las correlaciones se ha realizado analizando si existe una relación, y de qué tipo, entre las ganancias obtenidas, y los ajustes, sensibilidad y especificidad de los modelos, tanto aplicando los algoritmos con predicción conforme, como solo aplicando los algoritmos base. En casos en los que existe una ganancia si introducimos la predicción conforme podría ocurrir que ésta fuera constante, sea cual sea el comportamiento de los modelos base con los que se

Tabla VI
DESCRIPTIVOS SOBRE GANANCIAS APORTADAS POR LOS MODELOS CONFORMES

Estadístico	Ganancia MCC	Ganancia S	Ganancia E
Media	0.109	0.081	0.001
Mediana	0.054	0.048	0.002
Desviación típica	0.140	0.107	0.025
Rango	0.593	0.368	0.117
Mínimo	−0.057	−0.057	−0.061
Máximo	0.535	0.311	0.056
Percentil 25	0.0005	0.005	−0.009
Percentil 50	0.054	0.048	0.002
Percentil 75	0.205	0.112	0.018

Tabla VII
GANANCIAS MEDIAS SEGMENTADAS EN MODELOS CONFORMES

Segmentación	MCC	S	E
ARF	0.119	0.078	-0.012
AHT	0.166	0.109	0.001
LR	0.043	0.057	0.016
Proporción *malware* = 0.05	0.145	0.081	0.011
Proporción *malware* = 0.1	0.146	0.102	0.007
Proporción *malware* = 0.5336	0.037	0.060	-0.014
Proporción etiquetas = 0.0	0.238	0.188	0.009
Proporción etiquetas = 0.05	0.056	0.031	0.004
Proporción etiquetas = 0.1	0.034	0.024	-0.008

trabaja, o con las proporciones de *malware* y etiquetas que se han estudiado. Sin embargo, en el caso de que no lo sea, sería interesante saber si existen condiciones en las que la aplicación de la predicción conforme aporta una ganancia significativa. El estudio se lleva a cabo por separado para las tres métricas fijadas: MCC (ajuste), sensibilidad y especificidad. Es posible que la aplicación de aprendizaje conforme pueda resultar más útil en ciertos casos en los que se desee maximizar la detección de muestras maliciosas o minimizar los falsos positivos.

En la Tabla VIII se muestran las correlaciones entre el rendimiento de los modelos y las ganancias en el ajuste. Cabe destacar que, aunque existen correlaciones entre las ganancias en MCC y el MCC que tenían los modelos con predicción conforme, no siempre la correlación tiene el mismo signo, y no en todos los casos llega a ser significativa. Sin embargo, hay una evidente correlación significativa e inversa entre el MCC de los modelos base y las ganancias. Este hecho lleva a la conclusión de que la ganancia es más independiente de los modelos con predicción conforme, pero no de los modelos base. Por lo tanto, la predicción conforme resulta más eficaz cuando el rendimiento de los modelos base es bajo. Esto puede tener una justificación en el criterio de selección utilizado para las *pseudoetiquetas* (sección III-D): una alta credibilidad implica un mayor parecido con las instancias utilizadas para entrenar el modelo. Sin embargo, las instancias con mayores diferencias en la distribución, que son en las que el modelo previsiblemente cometerá más errores (y por lo tanto resultarían más informativas), no recibirán pseudoetiquetas. Si el modelo tiene un rendimiento bajo, debido por ejemplo a un desbalanceo significativo de los datos, las instancias pseudoetiquetadas pueden contribuir positivamente, pero a medida que el rendimiento mejora, las *pseudoetiquetas* pueden no ser lo suficientemente informativas.

En la Tabla IX se muestran las correlaciones de ganancias en sensibilidad. En este caso, solo existe una correlación

Tabla VIII
CORRELACIONES ENTRE LAS GANANCIAS Y LA VARIABLE MCC

Segmentación		Ganancia MCC
ARF	MCC_{D_1}	$(r = -0.217, \rho = 0.574)$
	MCC_{D_2}	$(r = -0.830, \rho < .001)$
AHT	MCC_{D_1}	$(r = -0.533, \rho = 0.139)$
	MCC_{D_2}	$(r = -0.926, \rho < .001)$
LR	MCC_{D_1}	$(r = -0.687, \rho < 0.040)$
	MCC_{D_2}	$(r = -0.823, \rho < .001)$
Proporción *malware* = 0.05	MCC_{D_1}	$(r = 0.392, \rho = 0.296)$
	MCC_{D_2}	$(r = -0.529, \rho = 0.142)$
Proporción *malware* = 0.1	MCC_{D_1}	$(r = 0.455, \rho = 0.217)$
	MCC_{D_2}	$(r = -0.873, \rho = 0.002)$
Proporción *malware* = 0.5336	MCC_{D_1}	$(r = -0.212, \rho = 0.583)$
	MCC_{D_2}	$(r = -0.971, \rho < .001)$
Proporción etiquetas= 0	MCC_{D_1}	$(r = -0.092, \rho = 0.812)$
	MCC_{D_2}	$(r = -0.715, \rho = 0.030)$
Proporción etiquetas=0.05	MCC_{D_1}	$(r = -0.121, \rho = 0.754)$
	MCC_{D_2}	$r = -0.589, \rho = 0.094)$
Proporción etiquetas= 0.1	MCC_{D_1}	$(r = -0.277, \rho = 0.470)$
	MCC_{D_2}	$(r = -0.649, \rho = 0.048)$

significativa para LR y la proporción de *malware* de 0.5336 y es en ambos casos inversa: un menor rendimiento en el modelo base se traduce en una mayor ganancia.

Tabla IX
CORRELACIONES ENTRE LAS GANANCIAS Y LA VARIABLE DE SENSIBILIDAD

Segmentación		Ganancia Sensibilidad
ARF	S_{D_1}	$(r = 0.016, \rho = 0.967)$
	S_{D_2}	$(r = -0.397, \rho = 0.290)$
AHT	S_{D_1}	$(r = 0.018, \rho = 0.964)$
	S_{D_2}	$(r = -0.420, \rho = 0.260)$
LR	S_{D_1}	$(r = -0.658, \rho = 0.054)$
	S_{D_2}	$(r = -0.916, \rho = 0.001)$
Proporción *malware* = 0.05	S_{D_1}	$(r = 0.601, \rho = 0.087)$
	S_{D_2}	$(r = 0.401, \rho = 0.285)$
Proporción *malware* = 0.1	S_{D_1}	$(r = -0.171, \rho = 0.660)$
	S_{D_2}	$(r = -0.673, \rho = 0.047)$
Proporción *malware* = 0.5336	S_{D_1}	$(r = -0.723, \rho = 0.028)$
	S_{D_2}	$(r = -0.980, \rho < .001)$
Proporción etiquetas= 0	S_{D_1}	$(r = 0.001, \rho = 0.999)$
	S_{D_2}	$(r = -0.382, \rho = 0.310)$
Proporción etiquetas= 0.05	S_{D_1}	$(r = -0.186, \rho = 0.633)$
	S_{D_2}	$(r = -0.391, \rho = 0.299)$
Proporción etiquetas= 0.1	S_{D_1}	$(r = -0.109, \rho = 0.780)$
	S_{D_2}	$(r = -0.296, \rho = 0.439)$

Los resultados del estudio de correlaciones entre la ganancia en especificidad y los resultados de la aplicación de cada modelo se incluye en la Tabla X. En términos de especificidad, no existe ningún patrón. Hay una correlación positiva entre el rendimiento del modelo base y la ganancia para ARF y AHT y negativa para LR. Para el resto de segmentaciones, solo hay una correlación negativa con la proporción de etiquetas 0.

IV-D. Discusión

El uso de predicción conforme para la obtención de etiquetas adicionales para los modelos mejora en cierta medida la eficacia de las predicciones. La mejora es más clara para el modelo AHT (el único para es la mejora es estadísticamente significativa en todas las métricas), aunque ARF y LR se benefician en menor medida. Hay dos limitaciones principales: en primer lugar, las ganancias no siempre son estadísticamente significativas. En segundo lugar, cuando los datos están balanceados y se incrementa el número de instancias etiquetadas,

Tabla X
CORRELACIONES ENTRE LAS GANANCIAS Y LA VARIABLE DE ESPECIFICIDAD

Segmentación		Ganancia Especificidad
ARF	E_{D_1}	$(r = 0.858, \rho = 0.003)$
	E_{D_2}	$(r = 0.704, \rho = 0.034)$
AHT	E_{D_1}	$(r = 0.907, \rho = 0.001)$
	E_{D_2}	$(r = 0.787, \rho = 0.012)$
LR	E_{D_1}	$(r = -0.178, \rho = 0.648)$
	E_{D_2}	$(r = -0.705, \rho = 0.034)$
Proporción *malware* = 0.05	E_{D_1}	$(r = -0.370, \rho = 0.327)$
	E_{D_2}	$(r = -0.564, \rho = 0.113)$
Proporción *malware* = 0.1	E_{D_1}	$(r = 0.120, \rho = 0.758)$
	E_{D_2}	$(r = -0.156, \rho = 0.689)$
Proporción *malware* = 0.5336	E_{D_1}	$(r = 0.309, \rho = 0.418)$
	E_{D_2}	$(r = -0.326, \rho = 0.392)$
Proporción etiquetas= 0	E_{D_1}	$(r = -0.676, \rho = 0.046)$
	E_{D_2}	$(r = -0.811, \rho = 0.008)$
Proporción etiquetas= 0.05	E_{D_1}	$(r = 0.616, \rho = 0.077)$
	E_{D_2}	$(r = 0.406, \rho = 0.278)$
Proporción etiquetas= 0.1	E_{D_1}	$(r = 0.803, \rho = 0.009)$
	E_{D_2}	$(r = 0.489, \rho = 0.182)$

los modelos conformes llegan a obtener peores resultados (aunque no de forma estadísticamente significativa). Por un lado, las instancias pseudoetiquetadas serán aquellas con menor incertidumbre en la predicción y por lo tanto son menos informativas para el modelo. Por otro lado, inevitablemente alguna *pseudoetiqueta* será errónea. Cuando el modelo tiene peor rendimiento (que coincide en los experimentos con datos desbalanceados y sin etiquetas) el enriquecimiento de los modelos conformes puede compensar estas limitaciones, lo que no ocurre en situaciones más favorables.

V. CONCLUSIONES

En este trabajo se ha llevado a cabo una evaluación de modelos enriquecidos con predicción conforme combinados con modelos *online* para obtener *pseudoetiquetas* fiables que puedan utilizarse para actualizar los modelos *online* y mejorar así su rendimiento. Los resultados concluyen que la integración de predicción puede resultar efectiva en ciertos casos, particularmente en los que el rendimiento del modelo base es más bajo. En situaciones en las que, por ejemplo, el conjunto de datos está balanceado y los modelos tienen buen rendimiento de base, los posibles errores de las *pseudoetiquetas* y su limitación a instancias con menor incertidumbre pueden resultar en una ausencia de mejora o incluso empeoramiento. Por otro lado, AHT es el único modelo cuyas ganancias resultan estadísticamente significativas en todas las métricas, lo que implica una dependencia del modelo o los datos utilizados en los resultados.

Una de las limitaciones en este trabajo es que los resultados están vinculados al conjunto de datos. Sería adecuado entonces replicar los experimentos con otros conjuntos de datos de similares y diferentes características para poder determinar que las conclusiones son más generales e, incluso, poder encontrar patrones de comportamiento de los modelos conformes. Por otro lado, es posible introducir mejoras en la predicción conformal. Las garantías respecto a la tasa de error (véase sección II) se aplica al total de los datos. Si por ejemplo hay un 90 % de instancias benignas y 10 % de maliciosas, todas ellas clasificadas incorrectamente por el modelo, la predicción conforme podría mantener una certeza de 90 % en las predicciones. Para adaptarse al desbalanceo de los datos se puede llevar a cabo un ajuste mondriano [7] para que las garantías de certeza en la predicción se apliquen a cada clase individualmente.

AGRADECIMIENTOS

Este trabajo se realiza en el marco de los fondos del Plan de Recuperación, Transformación y Resiliencia, financiados por la Unión Europea (Next Generation), en el proyecto estrátegico *Ciencia de datos para un modelo de inteligencia artificial en ciberseguridad*.

REFERENCIAS

[1] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *J. of Netw. and Comput. Appl.*, vol. 153, p. 102526, Mar. 2020.
[2] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.
[3] A. Guerra-Manzanares, M. Luckner, and H. Bahsi, "Android malware concept drift using system calls: Detection, characterization and challenges," *Expert Systems with Applications*, vol. 206, p. 117200, 2022.
[4] F. Ceschin, M. Botacin, H. M. Gomes, F. Pinagé, L. S. Oliveira, and A. Grégio, "Fast & furious: On the modelling of malware detection as an evolving data stream," *Expert Systems with Applications*, vol. 212, p. 118590, 2023.
[5] K. Xu, Y. Li, R. Deng, K. Chen, and J. Xu, "Droidevolver: Self-evolving android malware detection system," in *2019 IEEE European Symposium on Security and Privacy (EuroSP)*, 2019, pp. 47–62.
[6] Z. Kan, F. Pendlebury, F. Pierazzi, and L. Cavallaro, "Investigating labelless drift adaptation for malware detection," in *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, ser. AISec '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 123–134.
[7] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," 2022.
[8] H. Papadopoulos, N. Georgiou, C. Eliades, and A. Konstantinidis, "Android malware detection with unbiased confidence guarantees," *Neurocomputing*, vol. 280, pp. 3–12, 2018, applications of Neural Modeling in the new era for data and IT.
[9] F. Barbero, F. Pendlebury, F. Pierazzi, and L. Cavallaro, "Transcending transcend: Revisiting malware classification in the presence of concept drift," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 805–823.
[10] A. Guerra-Manzanares, H. Bahsi, and S. Nõmm, "Kronodroid: Time-based hybrid-featured dataset for effective android malware detection and characterization," *Computers & Security*, vol. 110, p. 102399, 2021.
[11] A. Bifet and R. Gavaldà, "Adaptive learning from evolving data streams," in *Advances in Intelligent Data Analysis VIII*, N. M. Adams, C. Robardet, A. Siebes, and J.-F. Boulicaut, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 249–260.
[12] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, pp. 1469–1495, 2017.
[13] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 20, no. 2, pp. 215–242, 1958.
[14] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem *et al.*, "River: machine learning for streaming data in python," 2021.
[15] B. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophys. Acta (BBA) - Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975.
[16] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, 2020.
[17] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers., 1988.
[18] W. P. Kegelmeyer, K. Chiang, and J. Ingram, "Streaming malware classification in the presence of concept drift and class imbalance," in *Proc. of 12th Int. Conf. on Mach. Learn. and Appl.*, vol. 2, 2013, pp. 48–53.