

Date of publication Nov 21, 2025, date of current version Nov 21, 2025.

Digital Object Identifier 10.1109/ACCESS.2024.XXXXX

Predicción de Grupos Relacionados por Diagnóstico (GRD) mediante Aprendizaje Automático

CAMILA EYZAGUIRRE¹, CRISTIAN LORCA¹, CINTYA OLIVARES¹, and ALEJANDRO SUAREZ¹

¹Universidad Andrés Bello, Santiago, Chile (e-mail: c.eyzaguirrevillarro@uandresbello.edu, c.lorcasvargas@uandresbello.edu, c.olivarescisternas@uandresbello.edu, a.suarezsanctilices@uandresbello.edu)

Corresponding author: Camila Eyzaguirre (e-mail: c.eyzaguirrevillarro@uandresbello.edu).

ABSTRACT La predicción precisa de los Grupos Relacionados por Diagnóstico (GRD) es fundamental para la gestión eficiente de recursos hospitalarios y la planificación de cuidados médicos. Este estudio presenta un modelo de aprendizaje automático basado en XGBoost para predecir códigos GRD a partir de información clínica, incluyendo diagnósticos, procedimientos y datos demográficos de pacientes. El dataset utilizado contiene 14,561 registros con 68 características, incluyendo diagnósticos principales y secundarios, procedimientos, y variables demográficas. Se implementó una estrategia de balanceo mediante agrupación de clases minoritarias y utilización de pesos de muestra para abordar el desbalance extremo observado (proporciones de hasta 1218:1 entre clases). El modelo fue entrenado con 205 features derivadas de los top 100 diagnósticos y procedimientos más frecuentes, junto con variables demográficas y características agregadas. Se obtuvo un accuracy de 59.22% en la predicción de 166 categorías GRD agrupadas, mejorando significativamente sobre un modelo aleatorio. Los resultados demuestran la viabilidad de utilizar técnicas de aprendizaje automático para la predicción automática de GRD, contribuyendo a la optimización de procesos hospitalarios.

INDEX TERMS Aprendizaje automático, GRD, predicción médica, XGBoost, clasificación multiclas, datos desbalanceados

I. INTRODUCTION

LOS Grupos Relacionados por Diagnóstico (GRD) constituyen un sistema de clasificación de pacientes ampliamente utilizado en la gestión hospitalaria y la planificación de recursos de salud [1], [2]. Estos códigos agrupan a pacientes con características clínicas y de consumo de recursos similares, facilitando la comparación de eficiencia hospitalaria y la asignación de recursos. La predicción precisa y temprana de los códigos GRD representa un desafío significativo en el ámbito de la salud, ya que estos códigos dependen de múltiples factores clínicos, incluyendo diagnósticos principales y secundarios, procedimientos realizados, edad del paciente, y otros factores demográficos.

El proceso tradicional de asignación de códigos GRD es manual y requiere experiencia clínica especializada, lo que puede resultar en inconsistencias, retrasos en la codificación, y un uso subóptimo de recursos. La automatización de este proceso mediante técnicas de aprendizaje automático ofrece

la promesa de mejorar la precisión, reducir el tiempo de procesamiento, y proporcionar predicciones más consistentes [3], [4].

Este trabajo aborda el problema de predicción de códigos GRD como una tarea de clasificación multiclas, donde cada código GRD representa una clase diferente. El desafío principal radica en el desbalance extremo de las clases, donde algunos códigos aparecen cientos de veces mientras que otros aparecen solo una vez en el dataset. Este desbalance plantea problemas significativos para los algoritmos de aprendizaje automático, que tienden a favorecer las clases mayoritarias en detrimento de las minoritarias [5].

II. LITERATURE REVIEW

La aplicación de técnicas de aprendizaje automático en la predicción de códigos de clasificación médica ha sido ampliamente explorada. Varios estudios han aplicado algoritmos como Random Forest, Support Vector Machines (SVM),

y redes neuronales para la clasificación de diagnósticos y códigos de procedimientos [6], [7].

Chen y Guestrin introdujeron XGBoost como una implementación eficiente y escalable de gradient boosting, demostrando excelente rendimiento en problemas de clasificación multiclase [8]. En el contexto médico, XGBoost ha mostrado resultados prometedores debido a su capacidad para manejar datos heterogéneos, valores faltantes, y clases desbalanceadas mediante el uso de pesos de muestra [9].

El manejo de clases desbalanceadas en problemas de clasificación médica ha sido abordado mediante diversas estrategias, incluyendo undersampling, oversampling, SMOTE, y el uso de pesos de clase [10]. Sin embargo, en casos de desbalance extremo, la agrupación de clases minoritarias ha demostrado ser una estrategia efectiva que preserva la información sin introducir sesgos artificiales [11].

Métricas apropiadas para evaluar modelos en problemas desbalanceados incluyen no solo accuracy, sino también precision, recall, F1-score, y especialmente multi-class logarithmic loss (mlogloss), que penaliza errores en la estimación de probabilidades [12].

III. OBJECTIVES

El objetivo principal de este estudio es desarrollar un modelo de aprendizaje automático capaz de predecir códigos GRD a partir de información clínica y demográfica de pacientes. Los objetivos específicos incluyen:

- 1) Analizar la calidad y distribución de los datos disponibles para identificar características relevantes y detectar problemas de desbalance.
- 2) Implementar una estrategia de preprocessamiento que maneje adecuadamente el desbalance extremo de clases mediante agrupación y pesos de muestra.
- 3) Desarrollar un modelo de clasificación multiclase utilizando XGBoost optimizado para el problema de predicción de GRD.
- 4) Evaluar el rendimiento del modelo utilizando métricas apropiadas para problemas multiclase desbalanceados.

IV. METHODOLOGY

A. DATASET DESCRIPTION

El dataset utilizado en este estudio contiene 14,561 registros de pacientes, cada uno caracterizado por 68 variables. Las características principales incluyen:

- **Diagnósticos:** Un diagnóstico principal y hasta 29 diagnósticos secundarios, cada uno con su código y descripción.
- **Procedimientos:** Múltiples procedimientos realizados, cada uno con código y descripción.
- **Variable objetivo:** Código GRD que incluye el código base más un dígito de severidad (MCC, CC, o sin CC).
- **Datos demográficos:** Edad del paciente, sexo, y otras características demográficas.

El dataset inicialmente contenía 14,561 registros, pero después de la eliminación de duplicados, se mantuvieron

todos los registros únicos para el análisis. La variable objetivo (GRD) presenta un desbalance extremo, con códigos que aparecen hasta 1,218 veces y otros que aparecen solo una vez.

B. METHODOLOGY OVERVIEW

El proceso de desarrollo del modelo siguió una metodología estructurada que incluyó las siguientes etapas:

1) Data Preprocessing

El preprocessamiento de datos consistió en:

- 1) **Limpieza de códigos GRD:** Se extrajo el código base eliminando el último dígito que representa la severidad (MCC/CC), resultando en códigos de 5 dígitos.
- 2) **Limpieza de diagnósticos y procedimientos:** Se extrajeron solo los códigos, eliminando las descripciones textuales.
- 3) **Eliminación de duplicados:** Se identificaron y eliminaron registros duplicados.
- 4) **Agrupación de clases minoritarias:** Códigos GRD con frecuencia menor a 5 fueron agrupados en una categoría "OTROS_GRD", reduciendo las clases de 210 a 166.

2) Feature Engineering

Se crearon 205 features mediante:

- **Features binarias de diagnósticos:** 100 features indicando presencia/ausencia de los top 100 códigos de diagnóstico más frecuentes.
- **Features binarias de procedimientos:** 100 features indicando presencia/ausencia de los top 100 códigos de procedimiento más frecuentes.
- **Features demográficas:** Edad (numérica), sexo (binario codificado como dos variables).
- **Features agregadas:** Número total de diagnósticos y número total de procedimientos por paciente.

Esta estrategia permitió capturar la información más relevante mientras se mantenía la dimensionalidad del problema manejable.

3) Machine Learning Techniques

a: XGBoost Selection and Justification

XGBoost (eXtreme Gradient Boosting) fue seleccionado como algoritmo principal por las siguientes razones:

- **Manejo de clases desbalanceadas:** XGBoost permite asignar pesos de muestra (*sample weights*) que balancean la importancia relativa de clases minoritarias y mayoritarias durante el entrenamiento.
- **Clasificación multiclase nativa:** Soporta directamente problemas de clasificación multiclase mediante la función objetivo *multi:softprob*, evitando la necesidad de estrategias one-vs-rest o one-vs-one.
- **Manejo de valores faltantes:** XGBoost maneja automáticamente valores faltantes en los datos, lo cual es común en datos clínicos.

- **Interpretabilidad:** Proporciona medidas de importancia de features, facilitando la comprensión de qué factores influyen más en las predicciones.
- **Rendimiento:** Ha demostrado excelente rendimiento en competencias de machine learning y problemas similares en el dominio médico.
- **Regularización incorporada:** Incluye parámetros de regularización L1 y L2 que ayudan a prevenir sobreajuste, crucial cuando el número de features es grande.

b: Training Configuration

El modelo XGBoost fue configurado con los siguientes hiperparámetros:

- **objective:** *multi:softprob* para clasificación multiclas con probabilidades.
- **eval_metric:** *mlogloss* (multi-class logarithmic loss).
- **max_depth:** 6 niveles de profundidad en los árboles.
- **learning_rate:** 0.1 para un aprendizaje balanceado.
- **subsample:** 0.8 (submuestreo de filas para regularización).
- **colsample_bytree:** 0.8 (submuestreo de columnas).
- **min_child_weight:** 3 (control de sobreajuste en hojas).
- **reg_alpha:** 0.1 (regularización L1).
- **reg_lambda:** 1.0 (regularización L2).
- **n_estimators:** 200 (número máximo de árboles).
- **early_stopping_rounds:** 20 (parada temprana si no hay mejora).

Se utilizó validación durante el entrenamiento con un conjunto de evaluación que monitoreaba tanto el conjunto de entrenamiento como el de prueba, permitiendo detectar sobreajuste en tiempo real.

El código completo del proyecto, incluyendo el notebook de Jupyter con todas las etapas de preprocesamiento, entrenamiento y evaluación del modelo, está disponible públicamente en: <https://github.com/Camiev/UNAB-MSI606-202581.1980/blob/main/proyecto1.ipynb>

4) Evaluation Metrics

a: Metric Selection and Justification

Para la evaluación del modelo en este problema de clasificación multiclas desbalanceado, se seleccionaron las siguientes métricas:

- 1) **Accuracy:** Proporción de predicciones correctas sobre el total. Aunque puede ser engañosa en problemas desbalanceados, proporciona una medida general del rendimiento del modelo.
- 2) **Multi-class Logarithmic Loss (mlogloss):** Métrica principal utilizada durante el entrenamiento y evaluación. Esta métrica penaliza no solo las clasificaciones incorrectas, sino también las estimaciones de probabilidad poco confiables. Es especialmente relevante porque:
 - Evalúa la calidad de las probabilidades predichas, no solo la clase predicha.

- Penaliza más los errores en clases con mayor confianza incorrecta.
- Es apropiada para problemas multiclas con múltiples clases.
- Proporciona información sobre la calibración del modelo.

La fórmula del mlogloss para un problema multiclas es:

$$L_{\log} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^M y_{i,c} \log(p_{i,c}) \quad (1)$$

donde N es el número de muestras, M es el número de clases, $y_{i,c}$ es 1 si la muestra i pertenece a la clase c , y $p_{i,c}$ es la probabilidad predicha de que la muestra i pertenezca a la clase c .

- 3) **Classification Report:** Incluye precision, recall, y F1-score por clase, proporcionando una visión detallada del rendimiento en cada categoría GRD.
- 4) **Confusion Matrix:** Para las clases más frecuentes, permite visualizar los errores de clasificación y entender qué códigos GRD son confundidos entre sí.

Estas métricas fueron seleccionadas porque, en conjunto, proporcionan una evaluación comprehensiva del modelo: mlogloss para optimización y calibración, accuracy para interpretabilidad general, y precision/recall para análisis detallado por clase.

V. EXPERIMENTS AND RESULTS

A. DATA ANALYSIS

1) Data Quality Assessment

a: Completeness

El análisis de completitud reveló que inicialmente no se encontraron valores nulos en el dataset original, lo que indica una buena calidad de datos en términos de completitud. Sin embargo, después del procesamiento de códigos, algunas variables derivadas pueden contener valores faltantes, los cuales fueron manejados mediante estrategias específicas durante la creación de features.

b: Correctness

La verificación de correctitud incluyó:

- Validación de la consistencia de códigos GRD antes y después de la limpieza.
- Verificación de la coherencia entre diagnósticos y procedimientos.
- Identificación de posibles errores de codificación mediante análisis de frecuencias anómalas.

Se identificó que la estructura de códigos era consistente y seguía los formatos esperados.

c: Outliers

El análisis de valores atípicos se centró en la distribución de códigos GRD. Se identificaron:

- **Códigos con frecuencia extremadamente baja:** 73 códigos con frecuencia menor a 10, de los cuales 45

tenían frecuencia menor a 5, incluyendo varios códigos que aparecían solo una vez.

- **Códigos con frecuencia extremadamente alta:** 37 códigos con frecuencia mayor a 100, siendo el más frecuente el código 14610 con 1,218 ocurrencias.

Estos valores extremos no representan necesariamente errores, sino la naturaleza heterogénea de las condiciones médicas y procedimientos. Sin embargo, requieren estrategias especiales de manejo para el aprendizaje automático.

2) Descriptive Statistics

El análisis estadístico de la distribución de códigos GRD reveló:

- **Total de códigos únicos:** 210 códigos GRD distintos después de la limpieza.
- **Distribución de frecuencias:**
 - Media: 68.65 ocurrencias por código
 - Mediana: 19 ocurrencias
 - Desviación estándar: 141.05 (indicando alta variabilidad)
 - Mínimo: 1 ocurrencia
 - Máximo: 1,218 ocurrencias
- **Cuartiles:**
 - Q1 (25%): 6 ocurrencias
 - Q2 (Mediana, 50%): 19 ocurrencias
 - Q3 (75%): 71.5 ocurrencias

Estas estadísticas confirman el desbalance extremo del dataset. La mediana (19) es significativamente menor que la media (68.65), indicando una distribución altamente sesgada hacia la derecha, típica de datos con clases desbalanceadas.

a: Top 10 Códigos GRD Más Frecuentes

Los códigos GRD más frecuentes representan aproximadamente el 40% de todos los registros:

- 1) 14610: 1,218 ocurrencias (8.37%)
- 2) 14612: 925 ocurrencias (6.35%)
- 3) 14613: 741 ocurrencias (5.09%)
- 4) 07114: 501 ocurrencias (3.44%)
- 5) 13416: 458 ocurrencias (3.15%)
- 6) 11412: 357 ocurrencias (2.45%)
- 7) 04415: 341 ocurrencias (2.34%)
- 8) 06120: 332 ocurrencias (2.28%)
- 9) 06113: 326 ocurrencias (2.24%)
- 10) 04416: 319 ocurrencias (2.19%)

3) Visualizations

Se generaron visualizaciones para comprender mejor la distribución de los datos:

- **Gráfico de barras horizontales:** Muestra los top 20 códigos GRD más frecuentes, permitiendo identificar las categorías dominantes.
- **Histograma de frecuencias:** Visualiza la distribución completa de frecuencias de códigos GRD, mostrando la concentración en valores bajos y la presencia de outliers.

- **Boxplot:** Ilustra la dispersión y valores atípicos en la distribución de frecuencias.

- **Gráfico de barras por rango:** Categoriza los códigos según rangos de frecuencia (menor a 10, entre 10 y 100, mayor a 100), facilitando la comprensión del desbalance.

Estas visualizaciones confirmaron visualmente el desbalance extremo y justificaron la necesidad de estrategias especiales de balanceo.

B. PRELIMINARY MODEL RESULTS

1) Data Splitting

El dataset fue dividido en conjuntos de entrenamiento y prueba con las siguientes características:

- **Tamaño del conjunto de entrenamiento:** 11,533 muestras (80%)
- **Tamaño del conjunto de prueba:** 2,884 muestras (20%)
- **Estrategia de división:** Estratificada por clase, asegurando que la proporción de cada clase se mantenga similar en ambos conjuntos.
- **Clases en entrenamiento:** 166 clases únicas (después de agrupación)

2) Class Balancing Strategy

Se implementó una estrategia dual de balanceo:

- 1) **Agrupación de clases minoritarias:** Códigos GRD con frecuencia menor a 5 fueron agrupados en la categoría "OTROS_GRD", reduciendo las clases de 210 a 166.
- 2) **Pesos de muestra:** Se calcularon pesos usando `compute_sample_weight` con estrategia 'balanced', que asigna pesos inversamente proporcionales a la frecuencia de clase.

Ejemplos de pesos asignados a las primeras 10 clases:

- Clase 0: peso 11.58 (frecuencia: 6)
- Clase 1: peso 2.48 (frecuencia: 28)
- Clase 2: peso 0.40 (frecuencia: 173)
- Clase 4: peso 17.37 (frecuencia: 4)

Estos pesos reflejan el principio de que las clases minoritarias reciben mayor importancia durante el entrenamiento.

3) Training Process

El modelo fue entrenado con los siguientes parámetros de configuración:

- **Features utilizadas:** 205 features (100 diagnósticos + 100 procedimientos + 3 demográficas + 2 agregadas)
- **Clases a predecir:** 166 categorías GRD
- **Diagnósticos únicos en dataset:** 3,649
- **Procedimientos únicos en dataset:** 904

El proceso de entrenamiento mostró una mejora constante en la métrica mlogloss:

- **Iteración 0:** mlogloss (train) = 4.572, mlogloss (test) = 4.594

- **Iteración 50:** mlogloss (train) = 1.022, mlogloss (test) = 1.734
- **Iteración 100:** mlogloss (train) = 0.639, mlogloss (test) = 1.596
- **Mejor iteración (130):** mlogloss (train) = 0.537, mlogloss (test) = 1.588

El modelo alcanzó su mejor rendimiento en el conjunto de prueba en la iteración 130, después de lo cual se observó un ligero sobreajuste (mejora en entrenamiento pero deterioro en prueba).

4) Model Performance

El modelo final obtuvo los siguientes resultados:

- **Accuracy:** 59.22%
- **Mejor mlogloss (test):** 1.588 (iteración 130)
- **Mejora respecto a baseline:** El mlogloss inicial de 4.594 se redujo a 1.588, representando una mejora del 65.4%.

Para un problema de clasificación multiclas con 166 clases, un accuracy de 59.22% representa un rendimiento significativamente mejor que un clasificador aleatorio (que obtendría aproximadamente 0.6% de accuracy). El mlogloss de 1.588 indica que el modelo proporciona estimaciones de probabilidad razonablemente calibradas.

5) Analysis of Results

Los resultados demuestran que:

- 1) **Viabilidad del enfoque:** Es posible predecir códigos GRD con una precisión razonable utilizando técnicas de aprendizaje automático.
- 2) **Efectividad del balanceo:** La estrategia de agrupación combinada con pesos de muestra permitió manejar el desbalance extremo sin perder información significativa.
- 3) **Selección de features:** El enfoque de usar los top 100 diagnósticos y procedimientos más frecuentes capturó la información más relevante mientras mantenía la dimensionalidad manejable.
- 4) **Espacio de mejora:** El gap entre mlogloss de entrenamiento (0.537) y prueba (1.588) sugiere que existe sobreajuste moderado, lo que indica oportunidades para mejorar la regularización.

VI. DISCUSSION

El modelo desarrollado demuestra que la predicción automática de códigos GRD es viable utilizando técnicas de aprendizaje automático modernas. El accuracy del 59.22% es prometedor considerando la complejidad del problema (166 clases) y el desbalance extremo de los datos.

La estrategia de agrupación de clases minoritarias resultó efectiva, preservando la información de clases raras sin introducir sesgos artificiales que podrían surgir de técnicas de oversampling o undersampling extremas.

El uso de XGBoost con pesos de muestra permitió que el modelo aprendiera de todas las clases, incluyendo las

minoritarias, lo cual es crucial para un sistema de predicción médica que debe ser útil para todas las condiciones, no solo las más comunes.

Sin embargo, se identificaron áreas de mejora:

- **Regularización:** Ajustar hiperparámetros de regularización podría reducir el sobreajuste y mejorar la generalización.
- **Feature engineering:** Explorar interacciones entre features o características temporales podría mejorar el rendimiento.
- **Ensemble methods:** Combinar múltiples modelos podría aumentar la robustez y precisión.

VII. CONCLUSION

Este estudio presenta un modelo de aprendizaje automático para la predicción de códigos GRD que:

- Maneja efectivamente el desbalance extremo mediante agrupación y pesos de muestra.
- Logra un accuracy del 59.22% en un problema de clasificación multiclas con 166 clases.
- Utiliza 205 features derivadas de diagnósticos, procedimientos y datos demográficos.
- Demuestra la viabilidad de automatizar la asignación de códigos GRD.

Los resultados son prometedores y sugieren que con refinamientos adicionales, este enfoque podría convertirse en una herramienta valiosa para la gestión hospitalaria. Futuros trabajos deberían explorar optimización de hiperparámetros, técnicas avanzadas de feature engineering, y validación en datasets adicionales.

CODE AVAILABILITY

El código fuente completo de este proyecto, incluyendo el notebook de Jupyter con todas las etapas de preprocesamiento, ingeniería de features, entrenamiento del modelo XGBoost y evaluación, está disponible públicamente en GitHub: <https://github.com/Camiev/UNAB-MSI606-202581.1980/blob/main/proyecto1.ipynb>

REFERENCES

- [1] R. B. Fetter, Y. Shin, J. L. Freeman, R. F. Averill, and J. D. Thompson, "Case mix definition by diagnosis-related groups," *Med. Care*, vol. 18, no. 2, pp. 1–53, 1980.
- [2] J. P. Weiner, I. Dobson, S. L. Maxwell, K. Coleman, B. Starfield, and G. Anderson, "Risk-adjusted Medicare capitation rates using ambulatory and inpatient diagnoses," *Health Care Financ. Rev.*, vol. 18, no. 3, pp. 77–99, 1996.
- [3] K. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, P. Sundberg, H. Yee, K. Zhang, Y. Zhang, G. Flores, G. E. Dahl, M. Furst, S. A. Kuhlmann, J. Hughes, J. B. Patil, W. A. Chou, K. de Fauw, J. R. Ledsam, O. Ronneberger, "Scalable and accurate deep learning with electronic health records," *NPJ Digit. Med.*, vol. 1, no. 1, pp. 1–10, 2018, doi: 10.1038/s41746-018-0029-1.
- [4] Z. Obermeyer and E. J. Emanuel, "Predicting the future—big data, machine learning, and clinical medicine," *N. Engl. J. Med.*, vol. 375, no. 13, pp. 1216–1219, 2016, doi: 10.1056/NEJMmp1606181.
- [5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

-
- [6] S. Liu, B. Liu, H. Elhajj, W. Fang, X. Liu, and H. Yu, "Multiclass classification of mechanical ventilated ICU patients by ICU outcome using machine learning," in Proc. IEEE Int. Conf. Bioinformatics Biomed. (BIBM), Madrid, Spain, 2018, pp. 570–575, doi: 10.1109/BIBM.2018.8621175.
 - [7] J. Heaton, N. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
 - [8] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
 - [9] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Ann. Statist., vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.
 - [10] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi, "Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance," Neural Netw., vol. 21, no. 2-3, pp. 427–436, 2008, doi: 10.1016/j.neunet.2007.12.031.
 - [11] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: a review," Int. J. Pattern Recognit. Artif. Intell., vol. 23, no. 4, pp. 687–719, 2009, doi: 10.1142/S0218001409007326.
 - [12] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," J. Mach. Learn. Technol., vol. 2, no. 1, pp. 37–63, 2011.

• • •