UN ENFOQUE BASADO EN DATOS PARA PREDECIR EVENTOS DELICTIVOS EN CIUDADES INTELIGENTES

A DATA-DRIVEN APPROACH FOR PREDICTING CRIMINAL EVENTS IN SMART CITIES

Jonathan Alfonso Mata Torres
Universidad Autónoma de Tamaulipas, México
a2093010058@alumnos.uat.edu.mx

Edgar Tello Leal
Universidad Autónoma de Tamaulipas, México
etello@uat.edu.mx

Ulises Manuel Ramírez Alcocer
Universidad Autónoma de Tamaulipas, México
a2093010066@docentes.uat.edu.mx

Gerardo Romero Galván
Universidad Autónoma de Tamaulipas, México
gromero@docentes.uat.edu.mx

Recepción: 22/octubre/2019 **Aceptación:** 23/noviembre/2019

Resumen

Actualmente, uno los retos de las instituciones gubernamentales es garantizar la seguridad de los habitantes. Este desafío también se presenta en el contexto de ciudades inteligentes, pero con la ventaja de tener sistemas de información de seguridad pública que colectan datos de los eventos delictivos en tiempo real. Por lo cual, se pueden diseñar enfoques basados en técnicas de minería de datos y aprendizaje automático que permitan predecir eventos delictivos basados en datos históricos y en el comportamiento identificados por zonas de una ciudad y en sus habitantes. En este trabajo se presenta un análisis predictivo de eventos criminales utilizando un conjunto de datos que almacena 6.4 millones de registros, colectados por un sistema de información implementado en una ciudad inteligente. El enfoque propuesto permite determinar la etiqueta de una clase de tipo binaria, la cual representa la probabilidad que un individuo sea arrestado al cometer un delito. Además, se realiza una comparación entre dos algoritmos de clasificación de datos:

algoritmo de árbol de decisión CART y algoritmo de ensamble AdaBoost, con el fin de determinar qué algoritmo obtiene un mejor rendimiento mediante la métrica de precisión y una validación cruzada. Previamente, en el conjunto de datos se aplica un método de selección de características para disminuir la dimensionalidad de los datos y el costo computacional en la ejecución de los algoritmos de clasificación.
Palabras Claves: Árbol de decisión, clasificación, ciudades inteligentes, predicción, selección de atributos.

Abstract

Nowadays, one of the challenges of government institutions is to guarantee the safety of the inhabitants. This challenge is also presented in the context of smart cities, but with the advantage of having public security information systems that collect data of criminal events in real time. Therefore, approaches based on data mining techniques and automatic learning can be designed to predict criminal events based on historical data and behavior identified by areas of a city and its inhabitants. This paper presents a predictive analysis of criminal events using a set of data that stores 6.4 million records, collected by an information system implemented in an intelligent city. The proposed approach allows determining the label of a class of binary type, which represents the probability that an individual is arrested when committing a crime. In addition, a comparison is made between two data classification algorithms: CART decision tree algorithm and AdaBoost ensemble algorithm, in order to determine which algorithm obtains better performance through precision metrics and cross-validation. Previously, a feature selection method is applied in the data set to reduce the dimensionality of the data and the computational cost in the execution of the classification algorithms.
Keywords**:** *Classification, decision tree, feature selection, prediction, smart cities.*

1. Introducción

El Internet de las Cosas (del inglés *Internet of Things*, IoT) es un paradigma donde un conjunto de objetos del mundo físico, interconectados entre sí, son vinculados a un ambiente virtual, lo cual posibilita la interacción humana en

ambientes inteligentes [Sosa,2018]. Las ciudades inteligentes son ambientes urbanos que soportan sus actividades utilizando tecnologías IoT y en conjunto con una interacción humana generan grandes y complejos volúmenes de datos heterogéneos [Pelton,2019], [Liu,2017].

En la actualidad, existe un creciente interés dentro de los grupos de investigación en el área de ciencia de datos e inteligencia artificial por analizar los grandes volúmenes de datos generados por los sistemas de información implementados en ciudades inteligentes. Lo anterior, tiene como objetivo el comprender las interacciones en el mundo real con el propósito de obtener conocimiento basado en datos para solucionar problemas que afectan a los ciudadanos y mejorar la calidad de los servicios disponibles en las ciudades inteligentes. Algunos ejemplos de la automatización de procesos de negocio y de sistemas inteligentes basados en datos en el contexto de ciudades inteligentes se han desarrollado para solucionar problemas en áreas como: gestión inteligente de tráfico vehicular, reducción de la contaminación del aire, trazar rutas óptimas de transporte público, recolección inteligente de desechos o garantizar la seguridad de los ciudadanos. En este sentido, una de las áreas que han despertado gran interés son los sistemas de información que dan soporte a la seguridad de los habitantes en las ciudades inteligentes, considerándose un reto fundamental en el desarrollo de sistemas de información debido a que no es una tarea trivial para los cuerpos de seguridad [Kadar, 2018]. Mediante sistemas de información inteligentes es posible diseñar soluciones que den soporte a la toma de decisiones. Los sistemas inteligentes en el área de seguridad ciudadana están conducidos a realizar actividades de manera automática. Por ejemplo, los sistemas de foto multas, botones de pánico o registro de denuncias a través de aplicaciones móviles. Estos sistemas inteligentes no tienen la capacidad de realizar predicciones de eventos en un futuro cercano.

Mediante las técnicas de minería de datos y aprendizaje automático se posibilitan la extracción de conocimiento para soportar la toma de decisiones estratégicas basadas en datos. Los algoritmos basados en árboles de decisión son técnicas de aprendizaje supervisado basados en las variables predictivas para la clasificación de instancias, es decir, estos algoritmos tienen capacidad para analizar los atributos

de los conjuntos de datos con el fin de aprender de los comportamientos identificados y su relación con valores o etiquetas de una clase, para posteriormente clasificar o predecir la clase a la que pertenece una nueva instancia.

Los algoritmos de árboles de decisión CART [Rutkouski, 2014], ID3 [Yang, 2017], C4.5 [Mohanty,2018], C5.0 [Yu,2018] y J48 [Panigrahi, 2018] emplean un enfoque divide y vencerás, lo cual permite construir un modelo con una estructura jerárquica en forma de árbol. Las características están ordenadas siguiendo una prioridad de arriba hacia abajo de acuerdo con una métrica (por ejemplo: ganancia de información, índice Gini o ganancia de radio [Han, 2012]), basada en probabilidades que evalúan la "*pureza*" de los atributos y definen si este atributo es un criterio de separación [Aggarwal,2015], [Witten,2011], [Alst,2016].

Por otro lado, uno de los retos al implementar enfoques supervisados o no supervisados utilizando grandes volúmenes de datos es la alta dimensión de los atributos. Este problema afecta en gran medida a los algoritmos basados en árboles de decisión, debido a que desarrollan el modelo a partir del entrenamiento utilizando los atributos, seleccionando estos atributos como criterios de división para crear el modelo. Entonces, es posible que los atributos sean irrelevantes para representar la clase a predecir. Por lo tanto, tener un conjunto de datos con gran cantidad de características (alta dimensionalidad) genera complejidad en los datos, construyendo un modelo débil, con baja precisión al entrenar, y clasificar nuevas instancias, así como un alto costo computacional. En este sentido, los métodos basados en selección de características pretenden reducir la complejidad de los datos, y el volumen del conjunto de datos de manera automática o semiautomática. Los métodos de selección de características se dividen en tres tipos: filtros, cobertura y embebidos [Bolon*,2015*], [Ramirez,*2018*]. Los métodos tipo filtro evalúan la relevancia de los atributos de un conjunto de datos de alta dimensión, utilizando métricas heurísticas con un bajo costo computacional. Los métodos basados en cobertura utilizan algoritmos de clasificación con el propósito de definir la relevancia de los atributos a partir de un ranking, los cuales se caracterizan por obtener alta precisión, pero incrementando el costo computacional [*Solorio, 2019*]. Finalmente, los métodos embebidos basan su funcionamiento en la búsqueda del

subconjunto de características óptimo, construido a partir de un clasificador, capturando dependencias con un menor costo computacional [Bolon*,2015*].

En este trabajo de investigación se presenta una comparación entre dos algoritmos de clasificación de datos. El primer algoritmo se basa técnicas de árboles de decisión (conocido como CART), y el segundo algoritmo de tipo ensamble (conocido como AdaBoost), estos algoritmos son elegidos con el propósito de comparar el rendimiento y la efectividad de los algoritmos basados en árboles de decisión implementados sobre grandes volúmenes de datos generados por sistemas inteligentes en ambientes reales. Para realizar esta implementación, en la etapa de entrenamiento son utilizadas un 80% de las instancias del conjunto de datos y en la etapa de evaluación con 20% de los registros del conjunto de datos, seleccionados aleatoriamente en forma automática. Para la validación de la propuesta se utiliza un conjunto de datos con un tamaño de 6.4 millones de registros, generados por un sistema de información implementado en una ciudad inteligente, en el ámbito de seguridad pública. Los datos son colectados en forma automática mediante dispositivos, sensores e interacción humano-máquina. La etiqueta de la clase a predecir por los algoritmos es de tipo binaria. Además, se presenta la implementación de un método de selección de características de tipo cobertura (eliminación recursiva de características), con el fin de reducir la dimensionalidad del conjunto de datos, así como disminuir el costo computacional y el tiempo de procesamiento de los algoritmos. La evaluación de los algoritmos de clasificación se realiza mediante la métrica de precisión, así como validación cruzada, en donde el algoritmo CART obtiene una precisión de 0.8266, en promedio. Finalmente, el enfoque propuesto es implementado mediante un sistema de software, que incluye funcionalidades de preprocesamiento de los datos de entrenamiento de los dos algoritmos de clasificación y predicción de la etiqueta de la clase a partir de una nueva instancia.

2. Métodos

Eliminación Recursiva de Características

El método de eliminación recursiva de características (del inglés *Recursive Feature Elimination*, RFE), funciona mediante un proceso iterativo donde el

algoritmo inicialmente selecciona el conjunto de datos completo, el cual va evaluando en cada iteración y removiendo las características una a una, de acuerdo con la función objetivo definida en la ecuación 1.

$$DJ(i) = \left(\frac{1}{2}\right)\frac{\partial^2 J}{\partial\, w_i^2}(Dw_i)^2 \qquad (1)$$

Donde $DJ(i)$ es la función de costo, J es una función cuadrática de w_i por lo cual estas dos funciones son equivalentes, $Dw_i = w_i$ es el cambio de los pesos que corresponde a remover la característica i del conjunto de datos. Entonces, inicialmente se entrena el clasificador, optimizando los pesos w_i respecto a J. A continuación, se calcula un criterio de *ranking* para todas las características, utilizando la función objetivo $DJ(i)$. Finalmente, se remueven los criterios que alcanzaron una posición menor dentro del ranking. Este proceso se ejecuta hasta encontrar los atributos más representativos del conjunto de datos.

El método RFE ha demostrado su eficiencia a través de su implementación utilizando diferentes algoritmos de clasificación, tales como máquinas de soporte vectorial [You, 2014], bayesiano ingenuo [Youn, 2009] bosques aleatorios [Zhou, 2014]. Debido al eficiente desempeño presentado en los trabajos mencionados utilizando múltiples clasificadores, en este trabajo se presenta la selección de atributos utilizando el método selección recursiva de características, con el propósito de elegir de manera automática los mejores atributos para entrenar los algoritmos de clasificación y mejorar su rendimiento.

Algoritmo de Clasificación CART

El algoritmo de árboles de regresión y clasificación permite clasificar instancias utilizando datos categóricos y datos continuos en el tiempo. Los datos son considerados variables aleatorias independientes $(\mathrm{X}^1, \ldots . \mathrm{X}^p, \mathrm{Y})$, donde X^k son las variables explicativas, Y es considerada la variable categórica a ser explicada. Entonces, un árbol de decisión es considerado como una estructura compuesta por nodos y ramas. Donde un *nodo terminal* es llamado *hoja* y cada *nodo no terminal* se define como un *atributo de división*, en donde estas divisiones son denominadas *ramas*, que conectan con otros atributos hasta encontrar una decisión final (hoja).

El algoritmo CART [Rutkouski, 2014] es un método basado en reglas que genera un árbol binario a través de particiones binarias recursivas, dividiendo los datos en subconjuntos de acuerdo con un criterio de división previamente seleccionado. Cada división se basa en una sola variable, algunas variables pueden ser usadas varias veces mientras que otras pueden ser ignoradas. El algoritmo CART se define formalmente como [Bel, 2009]: sea T el mapeo asignado de las hojas t para cada ejemplo $(\mathrm{X}_{\mathrm{i}}^{1}, \ldots . \mathrm{X}^{p})$, donde i es un índice para los ejemplos. T puede ser visto como un mapeo para asignar el valor $\tilde{\mathrm{Y}}_{i} = (\mathrm{X}_{\mathrm{i}}^{1}, \ldots . \mathrm{X}_{\mathrm{i}}^{\mathrm{p}})$ para cada muestra. Sea $p(j|t)$ la proporción de la clase j in la hoja t.

Algoritmo de Clasificación AdaBoost

El algoritmo Adaboost (del inglés *Adaptative Boosting*) [Chu, 2018] se considera un algoritmo de ensamble, ya que se puede conformar por múltiples algoritmos de clasificación base. En un algoritmo de *boosting* se asignan pesos a cada ejemplo de entrenamiento, y con ello una serie de k clasificadores son iterativamente entrenados. Después de que M_{i} es entrenado, los pesos son actualizados permitiendo al clasificador subsecuente $\mathrm{M}_{\mathrm{i+1}}$ enfocarse en las instancias que fueron clasificadas erróneamente por M_{i} [Han, 2012]. Una descripción formal del algoritmo AdaBoost se puede definir como: sea *D*, un conjunto de datos con *d* muestras etiquetadas por una clase, $(\mathrm{X}_{1}, \mathrm{y}_{1})(\mathrm{X}_{2}, 2), \ldots . , (\mathrm{X}_{\mathrm{d}}, \mathrm{y}_{\mathrm{d}})$, donde y_{i} es la clase de la muestra X_{i}. Inicialmente AdaBoost asigna a cada muestra de entrenamiento un peso igual a $\frac{1}{\mathrm{d}}$. Generando k clasificadores para el ensamble requiere k rondas a través del resto del algoritmo. En la ronda i, las instancias de D son muestreadas para formar el conjunto de entrenamiento D_{i}, de tamaño d, para realizar esta tarea se utiliza muestreo con remplazo. Por lo tanto, una muestra puede ser utilizada más de una vez en los entrenamientos de los clasificadores [Han, 2012].

3. Resultados

El enfoque propuesto se evaluará utilizando un conjunto de datos de incidentes de un delito o crimen, los cuales fueron colectados en la ciudad de Chicago, USA.

Las instituciones gubernamentales de Chicago han implementado diferentes estrategias basadas en las tecnologías de información y comunicaciones para automatizar servicios en beneficios de los ciudadanos, la mayoría son servicios soportados por sistemas de inteligentes y con la disponibilidad de datos abiertos contenidos en repositorios públicos en línea. En este sentido, el conjunto de datos contiene 6.4 millones de instancias y 22 atributos. En la tabla 1 se muestra la descripción de los atributos, los cuales se conforman por atributos con distintos tipos de datos: carácter, numérico, booleano, o espaciotemporales. Entre los atributos del conjunto de datos se pueden mencionar el distrito, descripción del delito, coordenadas, fecha, código del delito, entre otros. Cabe mencionar que los atributos *ID*, *Case Number*, y *Date* son descartados en el experimento al representar datos de identificados de la instancia (Tabla 1).

Tabla 1 Atributos del conjunto de datos.

Atributo	Descripción	Número de casos
ID	Identificador único para el registro.	6,457,411
Case Number	Identificador único para el incidente asignado por la policía de Chicago.	6,457,411
Date	Fecha cuando el incidente ocurrió.	2,740,512
Block	Extracto de la dirección donde el incidente ocurrió.	60,144
UICR	Código usado para clasificar incidentes criminales	350
Primary Type	Descripción primaria para el código UICR	35
Description	Descripción secundaria para el código IUCR.	380
Location Description	Descripción de la ubicación donde el incidente ocurrió.	180
Arrest	Variable binaria que indica si el criminal fue arrestado.	2
Domestic	Indica si el incidente está relacionado a violencia doméstica.	2
Beat	Indica la zona donde el incidente ocurrió. Un Beat es una pequeña área geográfica policiaca.	304
District	Indica el distrito policiaco donde el incidente ocurrió.	25
Ward	Ward (distrito del condado en la ciudad) donde el incidente ocurrió.	50
Latitude	La latitud de la ubicación donde el incidente tomó lugar.	861,599
Longitude	Longitud donde el incidente sucedió.	861,046
Location	Este atributo está conformado por los atributos latitud y longitud.	862,781
Community Area	Indica el área donde el incidente ocurrió.	77
FBI Code	Indica la clasificación del crimen basado en el sistema del FBI	26
X coordinate	La coordenada X donde el incidente ocurrió en el estado de Illinois.	78,582

Las instancias del conjunto de datos se colectan en forma automática mediante diferentes sistemas de información interconectados. Cada instancia contiene las características de la ocurrencia de un evento (delito), y la clase es determinada por el atributo *Arrest* (tabla 1).

El conjunto de datos se generó en tiempo real por lo cual puede contener inconsistencias, tales como valores nulos, instancias corruptas o erróneas. Por tal motivo, se realiza un pre-procesamiento en los datos que consiste en normalizar los atributos de tipo de dato “carácter”, remover valores no alfanuméricos de las instancias, y evaluar el tratamiento de valores nulos. A continuación, se realiza una conversión numérica del conjunto de datos, debido a que una codificación por un número entero permite reducir el volumen de los datos sin afectar el valor contenido en los datos.

Posterior a la etapa de pre-procesamiento de los datos, se ejecuta el método de selección de características RFE, con el fin de seleccionar en forma automática los atributos más representativos del conjunto de datos. El método RFE recibe como parámetros el número de características a seleccionar y un algoritmo de clasificación (en nuestra experimentación se implementa CART). De forma iterativa se evaluó el número de atributos en el que el algoritmo CART obtiene la mejor métrica de precisión.

En la tabla 2 se muestra un extracto de las características seleccionadas por el método RFE, en los que se alcanza una mejor precisión. La columna “No. Atributos” representa el valor de *K* (de 2 hasta 17 atributos), la mejor precisión (0.805) se obtiene con 7 atributos (Tabla 2) (Block, IUCR, Location Description, Beat, Ward, X coordinate y Location). De acuerdo con el método RFE los 7 atributos seleccionados permitirán predecir la etiqueta de la clase, posibilitando una reducción de la dimensionalidad del conjunto de datos.

Entonces, del conjunto de datos original se construye un sub-conjunto de datos que contiene exclusivamente los 7 atributos seleccionados y el total de registros del conjunto de datos original.

Posteriormente, se generan dos sub-conjuntos de datos para la etapa de entrenamiento y validación, utilizando el 80% de los registros para el entrenamiento

del algoritmo de clasificación y el 20% de los registros para la validación del modelo. Los registros o instancias son seleccionados aleatoriamente y en forma automática.

Tabla 2 Extracto de los atributos propuestos a seleccionar por el método RFE.

No. Atributos	Atributos seleccionados	Precisión
11	Block, IUCR, Location Description, Beat, Ward, Community Area, X Coord. Y Coord, Update On, Lat, Lon.	0.775
10	Block, IUCR, Prim. Type, Beat, Ward, Update On, Lat, Lon, Location	0.800
9	Block, IUCR, Primary Type, Beat, Ward, Community Area, Update On, Lon.	0.795
8	IUCR, Description, Location Description, Beat, Ward, X Coord, Y Coord, Location.	0.800
7	Block, IUCR, Location Description, Beat, Ward, X Coord, Location.	0.805
6	Block, IUCR, Beat, Ward, Lat, Lon.	0.770

En la figura 1 se puede observar que con un número de atributos igual a 6, la precisión disminuye considerablemente, comparada con un valor de *K*=7.

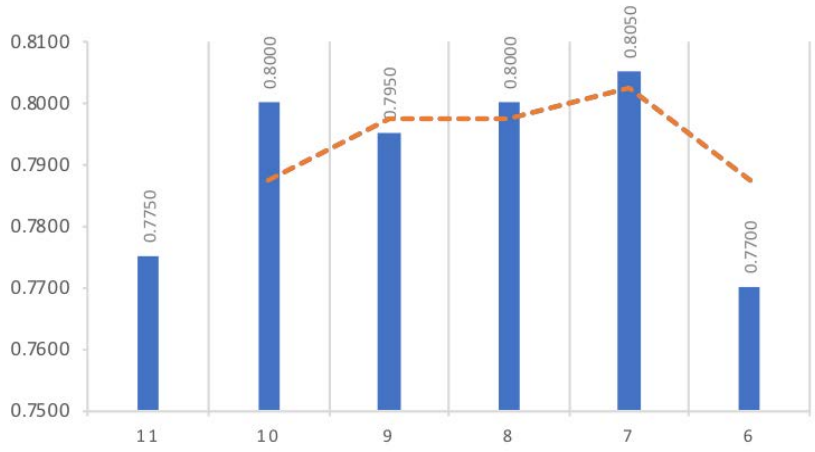

Figura 1 Precisión de acuerdo con el número de atributos seleccionados por método RFE.

En conjunto de datos se considera el atributo *Arrest* como la clase que permitirá la clasificación de los datos. Esta clase es de tipo binario y posibilita predecir en base a conjunto de atributos sí un individuo (humano) será arrestado al cometer un delito. Los algoritmos para clasificación de datos que se utilizan en nuestro experimento

son CART y AdaBoost, utilizando el mismo conjunto de datos de entrenamiento y validación para los dos clasificadores.

Con el fin de evaluar los clasificadores CART y AdaBoost se utilizó la técnica de validación cruzada con un valor de *k* igual a 10. Esta métrica evalúa 10 segmentos del conjunto de datos elegidos de manera aleatoria, para cada evaluación entrega la precisión obtenida para cada subconjunto. En la tabla 3 se muestra el promedio para la evaluación de los subconjuntos. Además, se evalúa la precisión para cada uno de los modelos obtenidos mediante tres entrenamientos, estas tres ejecuciones se realizaron con el propósito de calcular un resultado estimado que nos permita reducir el sesgo de la precisión obtenida con cada uno de los entrenamientos de los modelos utilizando sub-conjuntos aleatorios de datos. Finalmente, se presenta el promedio de la precisión obtenida en las tres ejecuciones.

Tabla 3 Precisión alcanzada por los clasificadores CART y AdaBoost.

Ejecución	Precisión-AB	AB 10-CF	Precisión-CART	CART 10-CF
1	0.8000	0.7949	0.8000	0.7971
2	0.8000	0.7812	0.8200	0.7736
3	0.8050	0.7799	0.8600	0.7775
Promedio:	*0.8010*	*0.7853*	*0.8266*	*0.7827*

Adicionalmente, se desarrolló un sistema de software que implementa el enfoque de clasificación propuesto, de manera gráfica e intuitiva. En la figura 2 se muestra el procedimiento para cargar el conjunto de datos (en formato csv) y seleccionar el algoritmo de clasificación.

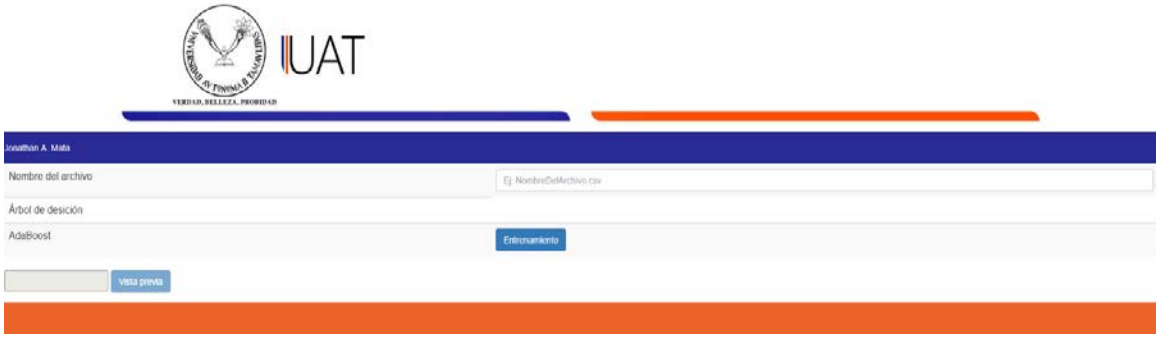

Figura 2 Aplicación web para la clasificación datos.

El sistema de software tiene funcionalidades que permiten ejecutar el pre-procesamiento de los datos, entrenamiento de los clasificadores. Además, el sistema de software desarrollado calcula y despliega métricas validación cruzada y precisión, tal como se muestra en la figura 3. Finalmente, en la interfaz presentada en la figura 3 se habilita la captura de los atributos de una nueva instancia, con el fin que los algoritmos de los clasificadores realicen la predicción de la etiqueta de la clase a la que pertenece.

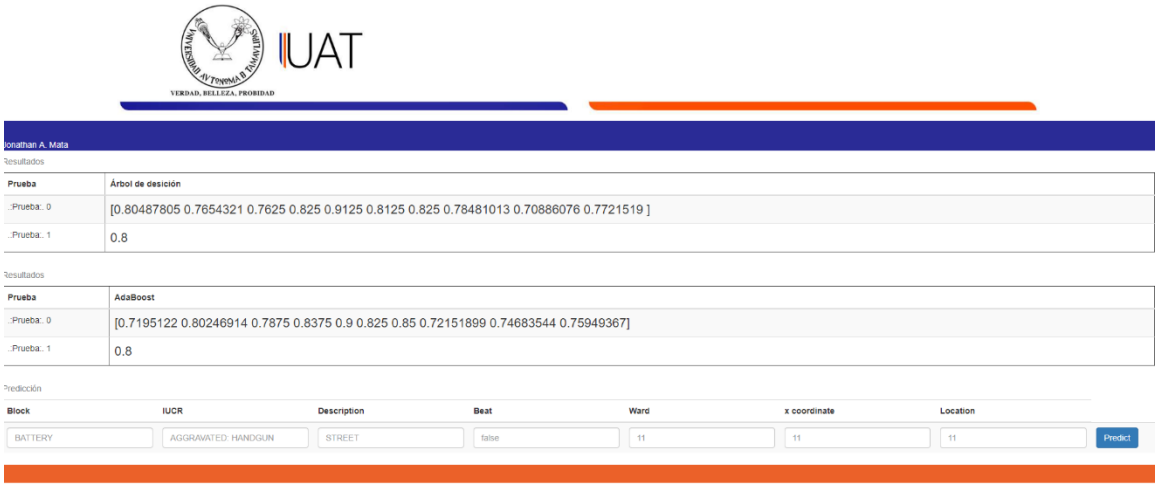

Figura 3 Despliegue de métricas obtenidas por los clasificadores en el entrenamiento.

4. Discusión

En la métrica validación cruzada (ver tabla 5) se observa que el promedio de los dos clasificadores es muy cercano, pero es menor a la precisión alcanzada por los clasificadores CART y AdaBoost al utilizar validación cruzada. Este resultado en la precisión se debe a que los subconjuntos de datos seleccionados por validación cruzada tienden a tener una alta variabilidad ocasionada por un desbalance de clases en el conjunto de datos. Por otro lado, el algoritmo CART obtiene una precisión promedio de 0.8266, superando al algoritmo AdaBoost. Al iniciar el estudio de investigación se esperaba que AdaBoost, identificado como un algoritmo de ensamble, obtuviera resultados superiores a su algoritmo base (CART).

Finalmente, al comparar los resultados obtenidos con la propuesta presentada en [Kummar, 2018], en la cual se ejecuta un análisis predictivo de delitos mediante un

clasificador bayesiano, reportan una precisión máxima de 0.7446 para clase en particular. En nuestro experimento se obtiene una métrica de precisión superior. Desde nuestro punto de vista, la selección de características basada en el método RFE posibilita alcanzar una tasa de precisión superior, al entrenar el modelo con atributos que permiten clasificar correctamente una cantidad mayor de instancias.

5. Conclusiones

El desarrollo de sistemas inteligentes basados en datos que posibiliten la toma de decisiones de manera automática es fundamental para trazar el futuro de las ciudades inteligentes.

En el enfoque propuesto se comprueba que los métodos de selección de características mejoran considerablemente la clasificación de conjuntos de datos, así como la predicción de etiquetas de una clase. La comparación realizada de los algoritmos de clasificación CART y AdaBoost permite validar el desempeño del algoritmo con subconjuntos de datos generados a partir de la selección de características recomendadas por el método RFE. Los algoritmos de clasificación de datos se evaluaron mediante las métricas de precisión y validación cruzada, en los que se observa que el algoritmo CART supera la precisión alcanzada por el algoritmo de ensamble.

Además, se presentó el desarrollo de un sistema de software que implementa el enfoque propuesto y permite ejecutar el pre-procesamiento de los datos, entrenamiento y clasificación utilizando los algoritmos mencionados.

En un trabajo futuro de la propuesta de investigación se plantea una combinación de diversos conjuntos de datos (por ejemplo, perfiles de usuario y datos de redes sociales), con el fin de determinar patrones de comportamiento de los ciudadanos en zonas identificados con índice de criminalidad dentro de ciudades inteligentes.

Agradecimientos

Este trabajo ha sido financiado por el Consejo Nacional de Ciencia y Tecnología (CONACYT) dentro del Programa Nacional de Posgrados de Calidad (PNPC).

6. Bibliografía y Referencias

[1] Aalst V., W.M.,Process Mining: Data Science in Action. Springer-Verlag Berlin Heidelberg, 2 edn., ISBN 978-3-662-49851-4, 2016.

[2] Aggarwal C., Data mining, 1st ed. New Delhi: Springer, ISBN 978-3-319-14142-8, 2015.

[3] Bel L., et. Al., CART algorithm for spatial data: Application to environmental and ecological data, Computational Statistics and Data Analysis. 3082-3093, 2009, https://doi.org/10.1016/j.csda.2008.09.012, 2009.

[4] Bolón-Canedo V., Sanchez-Moroño N., Alonso-Betanzos A., Feature Selection for High-Dimensional Data, Springer International Publishing, ISBN 978-3-319-21858-8, 2015.

[5] Chu J., Lee T., Ullah A., Component-wise AdaBoost algorithms for high-dimensional binary classification and class probability prediction, Handbook of Statistics, https://doi.org/10.1016/bs.host.2018.10.003, 2018.

[6] GUYON I., WESTON J., BARNHILL S., Gene Selection for Cancer Classification using Support Vector Machines, Machine Learning,46,389-422, https://doi.org/10.1023/A:1012487302797, 2002.

[7] Mohanty M., Sahoo S., Biswal Pradyut., Sabut S., Efficient classification of ventricular arrhythmias using feature selection and C4.5 classifier, Biomedical Signal Processing and Control, 44,200-208, https://doi.org/10.1016/j.bspc.2018.04.005,2018.

[8] Kadar C., Pletikosa I., Mining large-scale human mobility data for long-term crime prediction, EPJ Data Science, https://doi.org/10.1140/epjds/s13688-018-0150-z, 2018.

[9] Kai-Quan S., Chong-Jin O., Xiao-Ping L., Zhen H., Wilder-Smith E. P.V., A Feature Selection Method for Multilevel Mental Fatigue EGG Classification, IEEE Transaction on Biomedical Engineering, 54, 1231-1237, 10.1109/TBME.2007.890733, 2007.

[10] Kuman R., Nagpal B., Analysis and prediction of crime patters using big data, International Journal of Information Technology, https://doi.org/10.1007/s41870-018-0260-7, 2018.

[11] Liu, D., Huang, R. and Wosinski, M., Smart learning in smart cities. 1st ed. Springer Singapore, pp.18-19, https://doi.org/10.1007/978-981-10-4343-7, 2017.

[12] Pelton, J. and Singh, I., Smart cities of today and tomorrow. 1st ed. Springer International Publishing AG, part of Springer Nature, https://doi.org/10.1007/978-3-319-95822-4, 2019.

[13] Panigrahi R., Borah S., Rank Allocation to J48 of Decision Tree Classifiers using Binary and Multiclass Intrusion Detection Datasets, International Conference on Computational Intelligence and Data Science, 132, 323-332, https://doi.org/10.1016/j.procs.2018.05.186,2018.

[14] Solorio-Fernandez S., Carrasco-Ochoa J. A., Martínez-Trinidad J. F., A review of unsupervised feature selection methods, Artificial Intelligence Review, https://doi.org/10.1007/s10462-019-09682-y, 2019.

[15] Sosa C., Tello E., Lara D., Mata J., A Metodology Based on Model-Driven Enginnering for IoT Application Development, The Twelfth International Conference on Digital Society and Governments, ISBN: 978-1-61208-615-6, 2018.

[16] Ramírez-Gallego S., Mouriño-Talín H., Martínez-Rego D., Bolón-Canedo V., Manuel Benítez J., Alonso-Betanzos A., Herrera F., An Information Theory-Based Feature Selection Framework for Big Data Under Apache Spark, IEEE Transaction on Systems Man and Cybernetics Systems,48, 10.1109/TSMC.2017.2670926,2018.

[17] Rutkowski L., Jaworski M., Pietruczuk L., Duda P., The CART decision tree for mining data streams, Information Sciences,266,1-15, https://doi.org/10.1016 /j.ins.2013.12.060,2014.

[18] Vomfell L., Härdle W. and Lessmann S., improving crime count forecasts using Twitter and taxi data, Decision Support Systems, vol. 113, pp. 73-85, https://doi.org/10.1016/j.dss.2018.07.003, 2018.

[19] Yang S., Gou, J., Jin J., An improved Id3 algorithm for medical data classification, Computers and Electrical Engineering, 1-14, https://doi.org/10.1016/j.compeleceng.2017.08.005, 2017.

[20] Witten I., Frank E., Hall M., Data mining, 3rd ed. Burlington, Mass.: Morgan Kaufmann Publishers, https://doi.org/10.1016/C2009-0-19715-5,2011.

[21] Yu F., Li G., Chen H., Guo Y., Yuan Y., Coulton B., A VRF Charge Diagnosis Method based on Expert Modification C5.0 Decision Tree, International Journal of Refrigeration, https://doi.org/10.1016/j.ijrefrig.2018.05.034, 2018.

[22] You W., Yang Z., Ji G., PLS-based recursive feature elimination for high-dimensional small sample, Knowledge-Based Systems, 55,15-28, https://doi.org/10.1016/j.knosys.2013.10.004,2014.

[23] Youn E., Jeong M., Class dependent feature scaling method using naïve Bayes classifier for text datamining, Pattern Recognition Letters,30,477-485, https://doi.org/10.1016/j.patrec.2008.11.013, 2009.

[24] Zhou Q., Zhou H., Zhou Q., Yang f., Lou L., Structure damage detection based on random forest recursive feature elimination, Mechanical Systems and Signal Processing, 46,82-90, https://doi.org/10.1016/j.patrec .2008.11.013, 2014.