Algoritmos de aprendizaje automático para predicción de niveles de niebla usando ventanas estáticas y dinámicas.

M. Díaz-Lozano[a], D. Guijo-Rubio[a], P. A. Gutiérrez[a], C. Casanova-Mateo[b,c], S. Salcedo-Sanz[d], C. Hervás-Martínez[a]

***Resumen*—Los eventos de muy baja visibilidad producidos por niebla son un problema recurrente en ciertas zonas cercanas a ríos y grandes montañas, que afectan fuertemente a la actividad humana en diferentes aspectos. Este tipo de eventos pueden llegar a suponer costes materiales e incluso humanos muy importantes. Uno de los sectores más influenciados por las condiciones de muy baja visibilidad son los medios de transporte, fundamentalmente el transporte aéreo, cuya actividad se ve seriamente mermada, provocando retrasos, cancelaciones y, en el peor de los casos, terribles accidentes. En el aeropuerto de Valladolid son muy frecuentes las situaciones de baja visibilidad por niebla, especialmente en los meses considerados de invierno (noviembre, diciembre, enero y febrero). Esto afecta de forma directa a la manera en la que operan los vuelos de este aeropuerto. De esta forma, es muy importante conocer las posibles condiciones de niebla a corto plazo para aplicar procedimientos de seguridad y organización dentro del aeropuerto. En el presente artículo se propone el uso de diferentes modelos de ventanas dinámicas y estáticas junto con clasificadores de aprendizaje automático, para la predicción de niveles de niebla. En lugar de abordar el problema como una tarea de regresión, la variable de interés para la caracterización del nivel de visibilidad en el aeropuerto (Rango Visual de Pista, RVR) se discretiza en 3 categorías, lo que aporta mayor robustez a los modelos de clasificación obtenidos. Los resultados indican que una combinación de ventana dinámica con ventana estática, junto con modelos de clasificación basados en *Gradient Boosted Trees* es la metodología que proporciona los mejores resultados.**

***Palabras clave*—Series temporales, Eventos de baja visibilidad, modelos autorregresivos, predicción.**

I. Introducción

La niebla es un fenómeno meteorológico que consiste en la aparición de gotas de agua en suspensión en forma de gotas, lo suficientemente pequeñas como para que la gravedad terrestre no las atraiga hacia la superficie. Este fenómeno se manifiesta a nivel de suelo, pudiendo considerarse una nube a muy baja altura. Su aparición puede deberse a diferentes causas, como la evaporación de la humedad del suelo o a la expedición de vapor por parte de vegetación o de grandes masas de agua [1]. En cualquier caso, su aparición está íntimamente ligada a la disminución de las condiciones de visibilidad en la superficie. Estos eventos de baja visibilidad suponen un riesgo particular para el tráfico aéreo, marítimo y terrestre, provocando interrupciones y problemas cuyos costes humanos se han llegado a comparar a los causados por tormentas y tornados [2]. En las operaciones llevadas a cabo en los aeropuertos, el trafico de vuelos se ve fuertemente afectado en estas condiciones [3], [4], debiendo ampliar el tiempo entre aterrizajes y despegues y pudiendo provocar retrasos y cancelaciones. Por esta razón, el personal de los aeropuertos necesita conocer con cierta precisión y antelación si en un futuro cercano tendrán que trabajar con condiciones de baja visibilidad por niebla, para activar los protocolos necesarios en su caso, y tratar de mitigar este tipo de situaciones problemáticas.

La predicción de eventos futuros tiene interés en la mayoría de campos de estudio, creándose multitud de modelos destinados para este cometido [5], [6], aplicados con éxito a problemas de predicción reales. Todos estos modelos suelen estar basados en análisis de series temporales, normalmente sobre codificación real. Los eventos de niebla que provocan baja visibilidad son un problema recurrente en multitud de aeropuertos en todo el mundo, y su predicción se ha enfocado mediante diversas técnicas: en Perth, el aeropuerto más grande de la costa sudoeste de Australia, se desarrolló un modelo que aplicaba lógica difusa para conseguir una predicción precisa [7]; en el aeropuerto de Calcuta, India, se abordó el problema utilizando árboles de decisión para identificar los parámetros más importantes que influyen en la visibilidad, realizando predicciones mediante una red neuronal [8]; en el aeropuerto de Valladolid, situado al noroeste de España, se afrontó este problema haciendo uso de diversos algoritmos de aprendizaje automático para regresión como máquinas de vectores soporte, preprocesando la serie antes de aplicarlos [9].

El objetivo de este artículo es proponer un modelo de

[a]: Dpto. de Informática y Análisis Numérico, Universidad de Córdoba, Córdoba, España. E-mail: {i42dilom, dguijo, pagutierrez, chervas}@uco.es

[b]: LATUV: Laboratorio de Teledetección, Universidad de Valladolid, Valladolid, España.

[c]: Dpto. de Ingeniería Civil: Construcción, Infraestructura y Transporte, Universidad Politécnica de Madrid, Madrid, España.

[d]: Dpto. de Teoría de la Señal y Comunicaciones, Universidad de Alcalá, Alcalá de Henares, Madrid, España. E-mail: sancho.salcedo@uah.es

Este trabajo ha sido desarrollado con la financiación de los proyectos TIN2017-85887-C2-1-P, TIN2017-85887-C2-2-P y TIN2017-90567-REDT del Ministerio de Economía y Competitividad de España (MINECO) y fondos FEDER. La investigación de David Guijo Rubio ha sido subvencionada por el proyecto PI15/01570 de la Fundación de Investigación Biomédica (FIBICO) y por el Programa Predoctoral FPU (Ministerio de Educación y Ciencia de España), referencia de beca FPU16/02128.

predicción horario basado en el preprocesamiento de series temporales categóricas, donde los eventos temporales son tres diferentes condiciones atmosféricas relacionadas con la aparición de eventos de baja visibilidad por niebla: no-niebla, neblina y niebla. Este preprocesamiento se lleva a cabo mediante el uso de diversos tipos de ventanas de valores pasados de las series temporales consideradas, de forma que la información obtenida a partir de cada serie pueda ser usada en el entrenamiento de cualquier modelo de aprendizaje automático.

En la siguiente sección (sección II), se presenta la base de datos considerada. En la sección III, se detallan los modelos de ventana propuestos para preprocesar los datos y para realizar las predicciones. En la sección IV, se detalla la configuración de la experimentación y los resultados obtenidos. Por último, la sección V incluye las conclusiones obtenidas a partir de los resultados conseguidos.

II. Base de datos

II-A. Origen de los datos

Los datos utilizados en este artículo han sido recopilados mediante un sistema localizado en el aeropuerto de Valladolid y perteneciente a la Agencia Estatal de Meteorología Española (AEMET). Dicho sistema provee información relevante sobre las condiciones meteorológicas al personal de los aeropuertos (pilotos, controladores y personal de tierra). De forma horaria, el sistema recoge información sobre diferentes factores meteorológicos, tales como la temperatura o la humedad. Dichos datos son recogidos por sensores y, por tanto, no existen datos perdidos, pudiéndose considerar cada variable meteorológica obtenida como una serie temporal. Las variables recogidas por este sistema son las siguientes: Temperatura (grados Celsius), Humedad relativa (%), Velocidad del viento (m/s), Dirección del viento (grados), Presión reducida al nivel del mar (QNH, hPa) y Rango Visual de Pista (*Runaway Visual Range, RVR*), que es la variable objetivo a predecir y se mide en metros.

El RVR se obtiene a partir de la media ponderada de tres sensores de visibilidad (visibilímetros), colocados a diferentes alturas de la pista (zona de toma de tierra, media pista y zona de parada). Además, las condiciones de niebla pueden modelarse mediante la combinación del resto de variables meteorológicas medidas, por estar ambas intrínsecamente relacionadas. Los experimentos llevados a cabo en este artículo se han realizado con datos horarios de 8 años completos, concretamente desde el 1 de noviembre de 2009 hasta el 31 de diciembre de 2016. De la totalidad de datos, el 70 % inicial ha sido usado para la fase de entrenamiento y el último 30 % para la fase de generalización.

II-B. Umbralización del RVR

Los visibilímetros utilizados en el aeropuerto de Valladolid solo obtienen medida hasta 2000m, considerando situaciones de visibilidad óptima por encima de esta medida (es decir marcan 2000m incluso cuando el valor real de visibilidad es más alto). Esto es debido a que los valores de visibilidad por encima de 2000m son considerados óptimos, y por tanto no son relevantes para la gestión de situaciones de baja visibilidad en el aeropuerto. En la experimentación diseñada en este artículo se propone un enfoque de predicción categórico mediante 3 clases, en el que a cada hora se le asigna una de las 3 posibles clases (alta, media o baja en relación al valor de RVR, asociado a las condiciones de niebla). Los umbrales utilizados para la discretización de los valores de RVR son los siguientes:

$$\text{Clase} = \begin{cases} \text{niebla}, & \text{si } RVR < 1000, \\ \text{neblina}, & \text{si } 1000 <= RVR < 1990, \\ \text{no niebla}, & \text{si } RVR >= 1990. \end{cases}$$

De esta forma, se obtiene un problema de clasificación con 3 clases, altamente desequilibrado, debido a que las condiciones de baja visibilidad son, afortunadamente, muy minoritarias respecto a situaciones de visibilidad óptima. La Tabla I muestra la proporción de las diferentes clases en los conjuntos de entrenamiento y generalización considerados.

Tabla I
Proporción de clases

Clase	Entrenamiento	Generalización
niebla	856 (6 %)	772 (12 %)
neblina	912 (6 %)	514 (8 %)
no niebla	12294 (88 %)	4740 (80 %)
Total	14062 (70 %)	6026 (30 %)

III. Modelos utilizados

Una serie temporal se define como un conjunto de datos cronológicamente ordenados y muestreados con una frecuencia constante. Formalmente, una serie temporal unidimensional se define como:

$$Y = \{y_0, y_1, y_2, ..., y_N\},$$

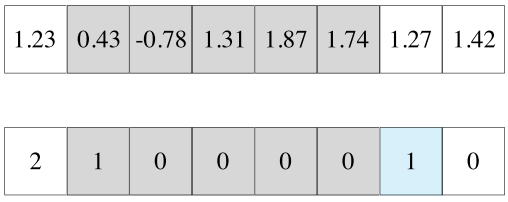
donde N es la longitud de la serie temporal.

La experimentación desarrollada en este artículo se ha llevado a cabo mediante el análisis de las series temporales descritas en la sección II-A, utilizando diferentes ventanas de valores pasados. Inicialmente, se crea un conjunto de patrones en los que la variable dependiente es el valor de la serie objetivo en el instante de tiempo a predecir y las variables independientes están formadas por la información extraída a partir de las ventanas. Con este conjunto de patrones, entrenamos cualquier modelo de aprendizaje automático, por lo que podríamos considerar que las ventanas actúan como un método de preprocesamiento.

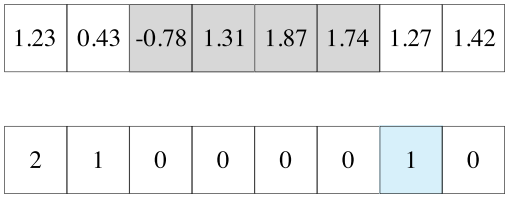
III-A. Extracción de características basada en ventanas

El análisis de series temporales mediante métodos autorregresivos permite modelar una variable en función de valores pasados, tanto de ella misma como de otras variables independientes relacionadas con el problema. En este artículo se propone el uso de 3 métodos, cada uno de los cuales limita de distinta forma la ventana de valores pasados utilizada para predecir el siguiente. Estos métodos pueden ser utilizados de forma individual, aplicando un único tipo de análisis,

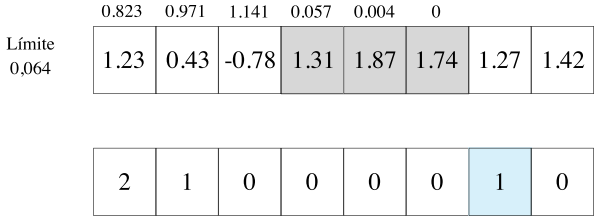
o combinada, de forma que se pueda aunar la información obtenida mediante varios tipos de ventanas. La Figura 1 muestra gráficamente el funcionamiento de estos 3 métodos en un problema sintético: una serie temporal categórica asociada a una variable dependiente a predecir y una serie temporal de valores reales asociada a una variable independiente, correlada con la primera. En dicha figura, el valor de la serie dependiente a predecir se encuentra sombreado en azul, mientras que las ventanas utilizadas para ello se encuentran sombreadas en gris.



(a) Ventana fija con *p=5*



(b) Ventana dinámica basada en cambio de clase



(c) Ventana dinámica basada en varianza con *r=0.1*

Figura 1. Distintos tipos de ventana propuestos.

III-A1. Modelo de ventana fija (VF): utiliza un número de instantes pasados constante para la predicción de cada valor de la serie temporal. Esta ventana es la que emplean los modelos autorregresivos (AR) clásicos, comúnmente utilizados en Estadística. El número de instantes es un parámetro del modelo, p (o orden del modelo AR), y los resultados obtenidos mediante este preprocesamiento son muy sensibles al valor de p. El modelo AR de orden p puede definirse como:

$$X_t = c + \sum_{i=1}^{p} \alpha_i X_{t-i} + \varepsilon_t \quad (1)$$

donde c representa una constante, α_i el coeficiente correspondiente al valor de la serie temporal en el instante $t-i$ y ε_t ruido blanco, esto es, una variable aleatoria Normal de media 0 y de varianza a determinar. En la Figura 1(a) se muestra un ejemplo de este tipo de preprocesamiento haciendo uso de una ventana fija de tamaño 5. Como se puede ver en dicha figura, para prededecir el valor sombreado en azul hace uso de los 5 valores anteriores de todas las series involucradas en el problema, tanto la dependiente como las independientes.

III-A2. Modelo de ventana dinámica basada en cambio de clase (VDCC): en series temporales categóricas, este modelo crea ventanas de tamaño variable limitándolas en base a los cambios de clase. De esta forma, para predecir un determinado valor, se identifica la clase inmediatamente anterior y se añaden valores anteriores mientras la clase no cambie, creándose ventanas más grandes cuando la clase es estable (existe una racha). En este tipo de preprocesamiento, los valores pasados de la serie dependiente son utilizados de forma indirecta dado que, aunque no se utilizan sus valores, son estos los que determinan el tamaño de las ventanas que se crean. En la Figura 1(b) se representa gráficamente la ventana creada para un determinado valor. En este ejemplo, la clase inmediatamente anterior al valor a analizar es 0, por lo que la ventana se extenderá hacia detrás, hasta que se encuentre una categoría distinta de 0, creando en este caso una ventana de 4 elementos. Tras ello, la información de las muestras de cada serie independiente que caigan dentro de la ventana se resume mediante las métricas detalladas en la sección IV-A.

III-A3. Modelo de ventana dinámica basada en varianza (VDV): en series temporales de valores reales, proponemos este modelo que crea ventanas de tamaño variable en función de la dinámica de la serie. Se analiza, de forma individual para cada serie temporal, la varianza de los valores incluidos en la ventana. Se añaden valores previos a la ventana hasta que se alcanza un determinado límite de varianza, que se establece como un porcentaje de la varianza total de la serie. Dicho porcentaje es un parámetro del modelo, r. La idea subyacente es que, cuando la varianza es demasiado alta, puede no tener sentido intentar resumir la información incluida en la misma. Además, dado que cada serie temporal se analiza de forma independiente, se pueden obtener ventanas de distinto tamaño para cada variable de entrada, cuando haya muchas variables independientes (como sucede en el problema considerado). La Figura 1(c) muestra un ejemplo de este procesamiento con un porcentaje de varianza total del 10 %. Este procesamiento no puede aplicarse sobre la serie categórica. En la serie de valores reales, se muestra, sobre cada valor, la varianza parcial, de forma tal que la ventana utilizada para predecir el valor sombreado en azul crecerá mientras que la varianza parcial sea menor que el límite (10 % de la total). Las ventanas serán resumidas mediante las métricas de la sección IV-A.

III-B. Clasificadores

La información obtenida tras el uso de las diferentes combinaciones de ventanas será utilizada para entrenar clasificadores que realicen la predicción de la categoría. En una primera aproximación, hemos considerado algunos de los modelos más clásicos de los incluidos en *scikit-learn* [15], una librería de código abierto implementada en *Python*. De esta forma, los clasificadores considerados son: Regresión logística (RL) [11],

Árboles de decisión (AD) [12] y conjuntos de clasificadores (concretamente *RandomForest (RF)* [13] y *GradientBoosting-Classifier (GB)* [14]).

IV. Experimentación y resultados

IV-A. Configuración

Como ya se especificó en la sección II-A, un 70 % de los datos se ha empleado como conjunto de entrenamiento, mientras que el 30 % formó el conjunto de *test*.

Para ambas ventanas dinámicas (tanto la basada en varianza como en el cambio de clase), se ha elegido resumir la información de las ventanas creadas utilizando dos estadísticos:

- La media aritmética ($\overline{W_s}$) de la ventana, definida como:

$$\overline{W_s} = \frac{1}{s} \sum_{y \in W_s} y$$

  donde s es el número de valores de la ventana y W_s la ventana creada.
- La varianza ($S^2_{W_s}$), utilizada para obtener una medida de dispersión de los valores contenidos en la ventana y definida como:

$$S^2_{W_s} = \frac{1}{s-1} \sum_{y \in W_s} (y - \overline{W_s})^2.$$

Los parámetros de los métodos de preprocesamiento y de los clasificadores han sido optimizados mediante una validación cruzada anidada de tipo 5-*fold*. A continuación se indican los valores explorados. Para el análisis de ventana fija, se ha utilizado un numero de muestras previas $p \in \{1, 2, 3, \ldots, 6\}$. Para el análisis de ventana dinámica basado en varianza, se ha optado por optimizar la proporción de varianza total utilizada para limitar las ventanas según $r \in \{0,1; 0{,}2; 0{,}3; \ldots; 0{,}6\}$.

Los parámetros de los modelos de clasificación también han sido optimizados mediante validación cruzada. Para RL, se ha optimizado la intensidad de la regularización mediante el parámetro $C \in \{10^{-3}, 10^{-2}, \ldots, 10^2, 10^3\}$. Para los AD se ha optimizado la función para medir la calidad de una división, utilizando el coeficiente *Gini* y la entropía. En el caso de RF, se ha optimizado el número de árboles en cada bosque $n \in \{10, 50, 100\}$, y, para GB, el número de árboles utilizado en el algoritmo ha sido $n \in \{50, 100, 150, 200\}$ y el ratio de aprendizaje $l \in \{0{,}05; 0{,}1; 0{,}2; 0{,}3\}$.

IV-B. Métricas de rendimiento

Existen multitud de métricas para evaluar la calidad de una clasificación obtenida. Una de las más comunes es el porcentaje de patrones bien clasificados (*Correct Classified Ratio, CCR*) con la que se obtiene una medida de rendimiento global. No obstante, el problema a tratar en este artículo es altamente desequilibrado, haciendo que cualquier clasificador trivial que clasifique todos los patrones en la clase mayoritaria obtenga buena puntuación en *CCR*. Por ello, el rendimiento ha de medirse mediante otras métricas. Hemos elegido la Media Geométrica de Sensibilidades (*Geometric Mean of the Sensitivities, GMS*), una medida de rendimiento que tiene en cuenta la precisión de la clasificación en todas las clases. El GMS se define como:

$$GMS = \sqrt[n]{\prod_{i=1}^{n} S_i}$$

donde n es el número de clases y S_i la sensibilidad de la clase i-ésima, es decir:

$$S_i = \frac{n_i}{N_i}$$

siendo n_i el número de instancias de la clase *i* correctamente clasificadas y N_i el número total de instancias de la clase *i*. Es una medida a maximizar.

Por otro lado, la naturaleza ordinal de nuestro problema (las categorías están organizadas en una escala ordinal) hace que sea necesario obtener una medida de error que penalice más algunos tipos de errores (por ejemplo, la confusión de "no niebla" con "niebla alta"), para lo que se utilizará la Media de Errores Absolutos Medios (*Average Mean Absolute Error, AMAE*). El $AMAE$ se define como:

$$AMAE = \frac{1}{n} \sum_{i=1}^{n} MAE_i$$

donde n es el número de clases y MAE_i es la desviación media producida con respecto a la clase real considerando solo la clase *i*. Se define como:

$$MAE_i = \frac{1}{N_i} \sum_{i=1}^{N_i} |C(y_i) - C(\hat{y_i})|$$

siendo N_i el número de instancias de la clase *i*, $C(y_i)$ la clase real de la instancia i y $C(\hat{y_i})$ la clase predicha para la instancia *i*, representadas mediante valores numéricos (es decir, $C(y_i) = 0$ para no niebla, $C(y_i) = 1$ para no neblina y $C(y_i) = 2$ para niebla). Al tratarse de una medida de error, es una métrica a minimizar.

Tabla II
Resultados de *test* en GMS

Tipos de ventanas	Clasificadores			
	GB	RF	RL	AD
VF	*0.624*	*0.453*	*0.567*	*0.518*
VDCC	0.286	0.375	0.297	0.405
VDV	0.0	0.072	0.278	0.286
VF+VDCC	0.617	**0.492**	0.547	**0.540**
VF+VDV	**0.656**	0.355	0.533	0.467
VDCC+VDV	0.475	0.368	0.442	0.425
VF+VDCC+VDV	0.620	0.447	**0.591**	0.469

IV-C. Resultados

Las Tablas II y III incluyen todos los resultados obtenidos, para GMS y $AMAE$, respectivamente. Por filas, se muestran las combinaciones de ventanas (ventana fija, VF, ventana dinámica basada en cambio de clase, VCCC, y ventana dinámica basada en varianza, VDV, más todas las combinaciones posibles, VF+VDCC, VF+VDV, VDCC+VDV y VF+VDCC+VDV) y por columnas cada uno de los 4

Tabla III
Resultados de *test* en $AMAE$

Tipos de ventanas	Clasificadores			
	GB	RF	RL	AD
VF	0.317	**0.373**	*0.314*	*0.437*
VDCC	0.538	0.592	0.419	0.660
VDV	*0.312*	0.406	0.333	0.442
VF+VDCC	0.314	0.385	0.318	**0.426**
VF+VDV	**0.304**	0.396	0.323	0.457
VDCC+VDV	0.349	0.412	**0.306**	0.474
VF+VDCC+VDV	*0.312*	*0.384*	0.316	0.461

clasificadores considerados. Los mejores resultados por cada clasificador se muestran en negrita, los segundos mejores en cursiva. El mejor resultado global se encuentra doblemente subrayado, mientras que el segundo mejor está marcado con subrayado simple.

Atendiendo a dichas tablas, se pueden obtener diversas conclusiones:

- En términos de *GMS*, los mejores resultados para cada modelo se obtienen siempre mediante un preprocesamiento en el que intervienen dos o más ventanas. En términos de *AMAE*, a excepción del modelo *RandomForest*, ocurre lo mismo, lo que demuestra que el uso combinado de distintas ventanas mejora ampliamente los resultados obtenidos, en comparación con los obtenidos mediante el uso individual de los mismos.
- Al optimizar el *AMAE*, conviene utilizar la combinación de ventanas que produce el segundo mejor resultado, debido a que el tamaño del patrón que genera el uso de VDCC+VDV será mucho más pequeño que usar VF+VDV. Esto se debe a que el uso de ventanas dinámicas resume las muestras de las ventanas en dos métricas (media y varianza), haciendo que como máximo los patrones cuenten con 20 características. Atendiendo al *grid* de parámetros utilizados para ventana fija, los patrones generados con este método junto con la ventana dinámica pueden ser de hasta 40 características, ralentizando bastante el proceso de entrenamiento.
- La combinación VF+VDV consigue los mejores resultados, tanto en *GMS* como *AMAE*, mediante el uso del clasificador *GradientBoosting*. La ventana basada en varianza provee de una capacidad dinámica en el análisis de las series independientes, adaptándose a cada serie de forma individual, lo que combinado con una ventana fija que utilice muestras previas devuelva el mejor resultado.
- La VF siempre obtiene los segundos mejores resultados en *GMS*. Este aspecto era esperable, dado que existe una alta persistencia en la serie categórica dependiente a predecir. Así, en la optimización del tamaño de ventana fija, se incluye la posibilidad de crear ventanas de tamaño 1, haciendo que el modelo tienda a predecir la salida únicamente en función de un instante pasado. Aunque esto devuelva unos resultados aceptables, no es deseable, dado que este tipo de modelos nunca serán capaces de detectar un cambio de clase en la serie.
- El uso de un preprocesamiento VDV aislado devuelve unos resultados pobres. Esto es debido a que el uso de este tipo de ventanas está orientado a series reales, y al tratar con una serie dependiente de naturaleza categórica, la serie a predecir se modela únicamente en función de series independientes, perdiendo una parte importante de la información.
- Los resultados de *GMS* obtenidos con el uso de VDCC aislado, a pesar de utilizar solamente los valores de las series independientes, son mejores que los de VDV. La razón de este resultado es que, a pesar de que VDCC no usa los valores de la serie dependiente de forma directa, son éstos los que determinan el tamaño de ventana para cada muestra, por lo que de forma indirecta se está utilizando una información inherente a la serie dependiente.

Por último, la Figura 2 muestra una comparación de un fragmento de 96 horas de la serie real y la serie predicha por *GradientBoosting* con la mejor combinación de ventanas. Como puede observarse, el modelo funciona de forma aceptable, prediciendo correctamente los cambios producidos en las etiquetas reales. Tan solo en la última parte del gráfico se observan algunos artificios introducidos por el método.

V. Conclusiones

Este artículo evalúa el uso de diferentes clasificadores de aprendizaje automático para predecir eventos de baja visibilidad por niebla en el aeropuerto de Valladolid. Se realiza una predicción categórica basada en tres posibles tipos de situaciones (no niebla, neblina y niebla), utilizando como entrada valores pasados de un conjunto de variables meteorológicas medidas en el aeropuerto. Proponemos considerar distintos tipos de ventanas a la hora de analizar los valores pasados: ventanas fijas junto con dos tipos de ventanas dinámicas (una basada en cambios de la categoría de los días pasados analizados y otra basada en varianza de la variable independiente examinada). La ventaja fundamental de estos métodos dinámicos es que evitan prefijar el tamaño de la ventana, pudiéndose adaptar de forma dinámica a la serie temporal considerada.

Los resultados obtenidos indican que la combinación de ventanas fijas y dinámicas (especialmente aquella basada en varianza), junto con el conjunto de clasificadores *GradientBoosting* obtiene los mejores resultados, con valores de $AMAE$ cercanos a $0{,}3$ (es decir, la predicción difiere en $0{,}3$ categorías, en media, con respecto al valor real) y un GMS mayor que el $65\,\%$. Teniendo en cuenta estos resultados, se puede concluir que los modelos obtenidos pueden mejorar la seguridad y la eficiencia de las operaciones aeronáuticas que se llevan a cabo en aeropuertos bajo condiciones de baja visibilidad por niebla. Como trabajo futuro, planteamos el uso de clasificadores ordinales, dada la naturaleza ordinal de la clases consideradas.

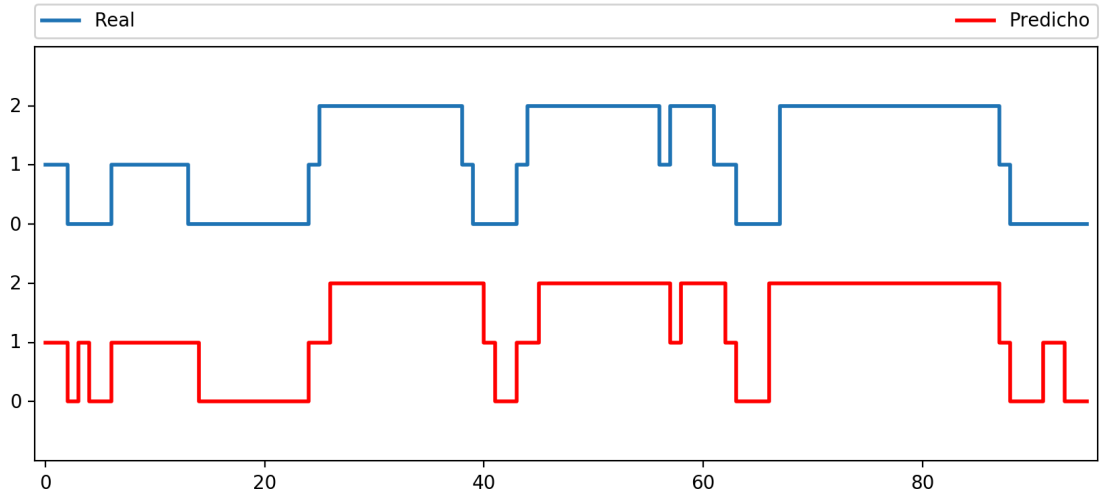


Figura 2. Comparación de etiquetas reales con las predichas por *GradientBoosting* con VF +VDV durante 96 horas. La clase 1 corresponde a no niebla, la clase 2 a neblina y la clase 3 a niebla.

Referencias

[1] D. Koracin, C. E. Dorman, J. M. Lewis, J. G. Hudson, E. M. Wilcox and A. Torregrosa, "Marine fog: A review," Atmospheric Research, vol. 143, pp. 142-175, 2014.

[2] I. Gultepe, et al., "Fog Research: A Review of Past Achievements and Future Perspectives," Pure and Applied Geophysics, vol. 164, pp. 1121-1159, 2007.

[3] H. Huang and C. Chen, "Climatological aspects of dense fog at Urumqi Diwopu International Airport and its impacts on flight on-time performance," Natural Hazards, vol. 81, pp. 1091-1106, 2016.

[4] N. Fedorova et al. "Fog Events at Maceio Airport on the Northern Coast of Brazil During 2002–2005 and 2007," Pure and Applied Geophysics, vol. 172, pp. 2727-2749, 2015.

[5] L. Zhenling et al., "Novel forecasting model based on improved wavelet transform, informative feature selection, and hybrid support vector machine on wind power forecasting," Journal of Ambient Intelligence and Humanized Computing, pp. 1-13, 2018.

[6] G. Noradin, A. Adel, S. Hossein and A. Oveis, "A new prediction model based on multi-block forecast engine in smart grid," Journal of Ambient Intelligence and Humanized Computing, pp. 1-16, 2017.

[7] Miao, Y. and Potts, R. and Huang, X., and Elliott, G. and Rivett, R., "A Fuzzy Logic Fog Forecasting Model for Perth Airport," Pure and Applied Geophysics, vol. 169, pp. 1107-1119, 2012.

[8] D. Duta and S. Chaudhuri, "Nowcasting visibility during wintertime fog over the airport of a metropolis of India: decision tree algorithm and artificial neural network approach," Natural Hazards, vol. 75, pp. 1349-1368, 2015.

[9] L. Cornejo-Bueno, C. Casanova-Mateo, J. Sanz-Justo, E. Cerro-Prada and S. Salcedo-Sanz, "Efficient Prediction of Low-Visibility Events at Airports Using Machine-Learning Regression," Boundary-Layer Meteorology, vol. 165, pp. 349-370, 2017.

[10] J. Yan-jie, G. Liang-peng, C. Xiao-shi and G. Wei-hong, "Strategies for multi-step-ahead available parking spaces forecasting based on wavelet transform," Journal of Central South University, vol. 24, pp. 1503-1512, 2017.

[11] J. P. Chao-Ying, L. L. Kuk and M. I. Gary, "An Introduction to Logistic Regression Analysis and Reporting," The Journal of Educational Research, vol. 96, pp. 3–14, 2002.

[12] L. Wei-Yin, "Fifty Years of Classification and Regression Trees," International Statistical Review, pp. 329-348, 2002.

[13] L. Breiman, "Random forests," Machine Learning, vol. 45(1), pp. 5-32, 2001.

[14] J. F. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," The Annals of Statistics, vol. 29, pp. 1189-1232, 2001.

[15] F. Pedregosa, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.