



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERIA ELECTRICA

DETECCIÓN DE EVENTOS RESPIRATORIOS EN SEÑALES POLISOMNOGRÁFICAS

TESIS PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO.

JAIME JOSÉ FELLER GOUDIE

PROFESOR GUÍA:
CLAUDIO ESTÉVEZ MONTERO

MIEMBROS DE LA COMISIÓN:
SEBASTIÁN RÍOS PÉREZ
BÁRBARA POBLETE LABRA

SANTIAGO DE CHILE

2017

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE INGENIERO CIVIL ELÉCTRICO.
POR: JAIME FELLER GOUDIE.
FECHA: 2017/07/31
PROF. GUÍA: CLAUDIO ESTÉVEZ MONTERO.

Resumen Ejecutivo

Las enfermedades respiratorias son un grupo de enfermedades que afectan, como su nombre lo indica, el sistema respiratorio. En Chile, estas están catalogadas como la tercera gran causa de muerte en Chile [1]. En particular, el Síndrome de Apnea e Hipopnea Obstructiva del Sueño (SAHOS) afecta al 4% de la población adulta [2] [3], y lleva consigo riesgos derivados de que interrumpe el ciclo normal de sueño.

El siguiente trabajo plantea el modelamiento de un clasificador que sea capaz de identificar los distintos tipos de episodios respiratorios asociados al SAHOS, apneas de distinto tipo (central, mixta u obstructiva) o hipopnea. Como consecuencia directa, se pretende obtener los sensores más relevantes para construir un polisomnógrafo.

Mediante el pre-procesamiento y obtención de datos estadísticos de exámenes de polisomnografía validados por personal médico, posterior entrenamiento y validación de distintos clasificadores; reglas difusas, árboles de decisión, redes neuronales y *support vector machine*; Se construye un clasificador que agrega un subconjunto de ellos para decidir a qué clase de episodio pertenece cada evento buscando un alto cociente de posibilidades de diagnóstico (DOR).

Debido a la complejidad de procesar múltiples datos se plantea utilizar métodos de reducción de variables mediante *random forest/tree ensemble*, logrando reducir estas a un 10% con una varianza no superior al 22% en la exactitud de los clasificadores respecto al ejercicio realizado con todas las variables. Esto se traduce en proponer una reducción del número de sensores ocupados en un examen polisomnográfico desde más de 20 a un conjunto de 8 que contiene la mayor parte de la información para lo que respecta a la diferenciación de episodios. Estos sensores concuerdan con las señales que son de especial cuidado cuando se estudia el SAHOS.

Se logra proponer un clasificador que diferencia entre los distintos episodios, pero no así uno que permita diferenciar entre la presencia o ausencia de estos, debido principalmente al hecho de no contar con una base de datos adecuada, esto último se deduce al comparar el desempeño de los clasificadores al introducir un grupo de control generado con estados de sueño normal de los mismos pacientes. El clasificador construido tiene un índice de acierto de sobre el 70%, con un error del tipo I cercano al 20% y error tipo II cercano 8%, la exactitud de este es sobre el 79% y el DOR superior a 10 para cada tipo de episodio.

Debido a la dificultad de conseguir este tipo de exámenes, en parte dado por su costo, tanto monetario como en otros recursos, no se pudo trabajar con una base de datos extensa. Como trabajo futuro se recomienda realizar estos experimentos con una base de datos que contenga exámenes realizados a personas que no presentan SAHOS, para de esta manera se obtenga una base de datos limpia para caracterizar las diferencias entre un paciente sano y uno enfermo.

Dedicatoria y Agradecimientos

A mis padres por apoyarme constantemente durante este viaje que ha sido el de estudiar, por enderezar mi camino cuando tuve dudas y mantenerme con el objetivo claro.

A mis hermanos y sobrinos por estar alegrándose cada día.

A Claudio Estévez por acogerme durante tres años y estar siempre dispuesto a apoyar.

A mis amigos, tantas noches compartidas y tonterías dichas.

A la Tere por estar a mi lado.

Tabla de Contenido

Resumen Ejecutivo	i
Dedicatoria y Agradecimientos	ii
Índice de abreviaturas	vi
Índice de Tablas.....	vii
Índice de Ecuaciones	x
Índice de Figuras	x
Capítulo 2 – Introducción	1
Enfermedades en Chile	1
Enfermedades Respiratorias.....	1
Síndrome de Apnea e Hipopnea Obstructiva del Sueño	2
Diagnóstico y Exámenes.....	3
E-Health	5
Hipótesis	5
Objetivos.....	5
Objetivo General	5
Objetivos Específicos.....	5
Metodología	5
Uso de Estadísticas en Clasificación.....	6
Clasificación de Señales	6
Dominio Temporal y Densidad Espectral.....	6
Técnica Propuesta	7
Adquisición de Datos.....	7
Pre-Procesamiento	7
Procesamiento	8
Capítulo 3 – Implementación	9
Herramientas Hardware	9
Herramientas de Software.....	9
Alice Sleepware®.....	9
MATLAB®	9
KNIME® Analytics Platform	9
Implementación de la Técnica Propuesta	9
Paso 2 - EDF a JSON	12
Paso 3 - JSON a Matriz.....	12

Paso 4 - Matriz a Texto	12
Paso 5 – Grupo de Control	12
Paso 6 - Conjuntos de Entrenamiento, Prueba y Validación.....	13
Paso 7 - Línea Base	14
Paso 8 - Reducción de Características.....	14
Paso 9 - Entrenamiento de Clasificadores.....	16
Paso 10 - Prueba.....	17
Paso 11 - Resultados de las Pruebas.....	19
Paso 12 - Diseño de Clasificador y Validación.....	19
Capítulo 4 – Resultados y Discusión	21
Resultados experimento 1 – Eventos	22
Resultados Línea Base	22
Resultados con Reducción por Random Forest.....	23
Resultados con Reducción por Tree Ensemble	24
Tabla Comparación Reducción de Variables	26
Ecuación Clasificador Final	26
Matrices de Confusión	26
Resultados Experimento 2 – Eventos y Data “sin eventos”	29
Resultados Línea Base	29
Resultados con Reducción por Random Forest.....	30
Resultados con Reducción por Tree Ensemble	31
Tabla Comparación Reducción de Variables	33
Ecuación Clasificador Final	33
Matrices de Confusión	33
Resultados Experimento 3 – Eventos y Data “sin eventos”, proporcional	34
Resultados Línea Base	34
Resultados con Reducción por Random Forest.....	35
Resultados con Reducción por Tree Ensemble	36
Tabla comparación reducción de variables	38
Ecuación Clasificador Final	38
Matrices de Confusión	38
Resultados Reducción de Variables.....	39
Random Forest	39
Tree Ensemble.....	41
Discusión Resultados	43
Capítulo 5 – Conclusiones y Trabajo Futuro.....	47

Conclusiones	47
Trabajo Futuro	48
Bibliografía.....	49
Anexo 1 – Scripts MATLAB®	A
blockEdfLoad.....	A
Paso1_edf2json.m	N
Paso1_edf2json_Healthy.m	U
Paso2_json2mat.m	AA
Paso2_json2mat_Healthy.m.....	FF
Paso3_mat2txt.m.....	JJ
Paso3_mat2txt_Healthy.m	KK
csv2cell.m	LL
fourier.m.....	QQ
Anexo 2 – Configuración KNIME®	RR
Anexo 3 – Archivos Excel.....	SS
Anexo 4 – Prueba con K-Fold Cross-Validation.....	TT
Línea Base más 10-fold cross validation	TT
Reducción por random forest más 10-fold cross validation	TT
Reducción por tree ensemble más 10-fold cross validation.....	UU
Discusión de los resultados.....	UU

Índice de abreviaturas

CSA:	Apnea central de sueño (de <i>Central Sleep Apnea</i>).
CVS:	Valores separados por comas (de <i>Comma-Separated Values</i>).
DOR:	Cociente de posibilidades de diagnóstico (de <i>Diagnostic Odds Ratio</i>).
DT:	Clasificador por árbol de decisión (de <i>Decission Tree</i>).
ECG:	Electrocardiograma.
EDF:	Formato de datos europeo (de <i>European Data Format</i>).
EEG:	Electroencefalograma.
EMG:	Electromiograma.
EOG:	Electrooculograma.
FFT:	Transformada rápida de Fourier (de <i>Fast Fourier Transform</i>).
FN:	Falsos negativos (de <i>False Negative</i>).
FNR:	Razon de falsos negativos (de <i>False Negative Rate</i>).
FP:	Falsos positivos (de <i>False Positive</i>).
FPR:	Razon de falsos positivos (de <i>False Positive Rate</i>).
JSON:	Notación de objetos JavaScript (<i>JavaScript Object Notation</i>).
MLP:	Clasificador por medio de redes neuronales con multiples capas (de <i>Multi-Layer Perceptron</i>).
MSA:	Apnea mixtas del sueño (de <i>Mixed Sleep Apnea</i>).
NaN:	No numérico (de <i>Not a Number</i>).
NB:	Clasificador ingenuo de Bayes (de <i>Naive Bayes</i>).
OSA:	Apnea obstructiva del sueño (de <i>Obstructive Sleep Apnea</i>).
PNN:	Clasificador por medio de redes neuronales probabilísticas (de <i>Probabilistic Neural Network</i>).
PSG:	Polisomnografía.
RF:	Random Forest.
RUN:	Rol Único Nacional.
SAHOS:	Síndrome de Apnea e Hipopnea Obstructiva del Sueño.
SVM:	Clasificador de máquinas de vectores de soporte (de <i>Support Vector Machine</i>).
TE:	Arreglo de Arboles (de <i>Tree Ensemble</i>).
TN:	Verdaderos negativos (de <i>True Negative</i>).
TNR:	Razon de verdaderos negativos (de <i>True Negative Rate</i>).
TP:	Verdaderos positivos (de <i>True Positive</i>).
TPR:	Razon de verdaderos Positivos (de <i>True Positive Rate</i>).

Índice de Tablas

<i>Tabla 1: Algunos indicadores de mortalidad por causa. Chile año 2012. /*Tasa por 100.000 habitantes.</i>	1
<i>Tabla 2: Interpretación del DOR según intervalos</i>	19
<i>Tabla 3: Matriz de Confusión para Línea Base, Hipopnea.</i>	22
<i>Tabla 4: Matriz de Confusión para Línea Base, Apnea Obstructiva del Sueño.</i>	22
<i>Tabla 5: Matriz de Confusión para Línea Base, Apnea Central del Sueño.</i>	23
<i>Tabla 6: Matriz de Confusión para Línea Base, Apnea Mixta del Sueño.</i>	23
<i>Tabla 7: Matriz De Confusión para prueba con reducción de características por Random Forest, Hipopnea</i>	23
<i>Tabla 8: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Obstructiva del Sueño.</i>	23
<i>Tabla 9: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Central del Sueño.</i>	23
<i>Tabla 10: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Mixta del Sueño.</i>	24
<i>Tabla 11: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Hipopnea.</i>	24
<i>Tabla 12: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Apnea Obstructiva del Sueño.</i>	24
<i>Tabla 13: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Apnea Central del Sueño.</i>	24
<i>Tabla 14: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Central del Sueño.</i>	25
<i>Tabla 15: Resultados Reducción de Variables, los porcentajes representan la fracción de variables seleccionadas respecto a la línea base.</i>	26
<i>Tabla 16: Resultados Primera Etapa del Clasificador diseñado</i>	26
<i>Tabla 17: Resultados clasificadores sobre eventos sin clasificación luego de la primera etapa.</i>	27
<i>Tabla 18: Resultados Segunda Etapa del Clasificador diseñado</i>	27
<i>Tabla 19: Resultados del Clasificador diseñado (Suma Tabla 17 y Tabla 18).</i>	27
<i>Tabla 20: Matriz de Confusión para Línea Base, Hipopnea.</i>	29
<i>Tabla 21: Matriz de Confusión para Línea Base, Apnea Obstructiva del Sueño.</i>	29
<i>Tabla 22: Matriz de Confusión para Línea Base, Apnea Central del Sueño.</i>	29
<i>Tabla 23: Matriz de Confusión para Línea Base, Apnea Mixta del Sueño.</i>	30
<i>Tabla 24: Matriz de Confusión para Línea Base, Sin evento.</i>	30
<i>Tabla 25: Matriz De Confusión para prueba con reducción de características por Random Forest, Hipopnea</i>	30
<i>Tabla 26: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Obstructiva del Sueño.</i>	30
<i>Tabla 27: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Central del Sueño.</i>	31
<i>Tabla 28: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Mixta del Sueño.</i>	31
<i>Tabla 29: Matriz de Confusión para prueba con reducción de características por Random Forest, Sin evento.</i>	31

<i>Tabla 30: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Hipopnea.</i>	31
<i>Tabla 31: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Apnea Obstructiva del Sueño.</i>	32
<i>Tabla 32: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Apnea Central del Sueño.</i>	32
<i>Tabla 33: Matriz de Confusión para prueba con reducción de características por Tree Ensemble, Apnea Central del Sueño.</i>	32
<i>Tabla 34: Matriz de Confusión para prueba con reducción de características por Tree Ensemble, Sin evento.</i>	32
<i>Tabla 35: Resultados Reducción de Variables.</i>	33
<i>Tabla 36: Matriz de Confusión para Línea Base, Hipopnea.</i>	34
<i>Tabla 37: Matriz de Confusión para Línea Base, Apnea Obstructiva del Sueño.</i>	34
<i>Tabla 38: Matriz de Confusión para Línea Base, Apnea Central del Sueño.</i>	34
<i>Tabla 39: Matriz de Confusión para Línea Base, Apnea Mixta del Sueño.</i>	35
<i>Tabla 40: Matriz de Confusión para Línea Base, Sin evento.</i>	35
<i>Tabla 41: Matriz De Confusión para prueba con reducción de características por Random Forest, Hipopnea.</i>	35
<i>Tabla 42: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Obstructiva del Sueño.</i>	35
<i>Tabla 43: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Central del Sueño.</i>	36
<i>Tabla 44: Matriz de Confusión para prueba con reducción de características por Random Forest, Apnea Mixta del Sueño.</i>	36
<i>Tabla 45: Matriz de Confusión para prueba con reducción de características por Random Forest, Sin evento.</i>	36
<i>Tabla 46: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Hipopnea.</i>	36
<i>Tabla 47: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Apnea Obstructiva del Sueño.</i>	37
<i>Tabla 48: Matriz de Confusión para Prueba con reducción de características por Tree Ensemble, Apnea Central del Sueño.</i>	37
<i>Tabla 49: Matriz de Confusión para prueba con reducción de características por Tree Ensemble, Apnea Central del Sueño.</i>	37
<i>Tabla 50: Matriz de Confusión para prueba con reducción de características por Tree Ensemble, Sin evento.</i>	37
<i>Tabla 51: Resultados Reducción de Variables.</i>	38
<i>Tabla 52: Resumen Clasificador Diseñado.</i>	43
<i>Tabla 53: Otros índices, Clasificador Diseñado.</i>	43
<i>Tabla 54: Resumen desempeño algoritmos en prueba de entrenamiento con eventos validados por medico.</i>	44
<i>Tabla 55: Comparación Exactitud clasificadores hipopnea en caso base y post reducción de variables.</i>	44
<i>Tabla 56: Comparación Exactitud clasificadores apnea obstructiva en caso base y post reducción de variables.</i>	44
<i>Tabla 57: Comparación Exactitud clasificadores apnea central en caso base y post reducción de variables.</i>	45

<i>Tabla 58: Comparación Exactitud clasificadores apnea mixta en caso base y post reducción de variables</i>	45
<i>Tabla 59: Matriz de Confusión para línea base utilizando cross-validation, Hipopnea.</i>	TT
<i>Tabla 60: Matriz de confusión para prueba con reducción de características por random forest utilizando cross-validation, hipopnea</i>	TT
<i>Tabla 61: Matriz de confusión para prueba con reducción de características por tree ensemble utilizando cross-validation, hipopnea.</i>	UU

Índice de Ecuaciones

<i>Ecuación 1: Número de características</i>	12
<i>Ecuación 2: Puntuación Arboles decisión</i>	16
<i>Ecuación 3: Regla filtro de características</i>	16
<i>Ecuación 4: Exactitud (Accuracy)</i>	18
<i>Ecuación 5: Sensibilidad (Sensibility, True Positive Rate)</i>	18
<i>Ecuación 6: Especificidad (Specificity, True Negative Rate)</i>	18
<i>Ecuación 7: Tasa de Fallos (Miss Rate, False Negative Rate)</i>	18
<i>Ecuación 8: Probabilidad de Falsa Alarma (Fall-out, False Positive Rate)</i>	18
<i>Ecuación 9: Precisión (Precision, Positive Predictive Value)</i>	18
<i>Ecuación 10: Valor Predictivo Negativo (Negative Predictive Value)</i>	18
<i>Ecuación 11: Cociente de Probabilidad Positivo (Positive Likelihood Ratio)</i>	18
<i>Ecuación 12: Cociente de Probabilidad Negativo (Negative Likelihood Ratio)</i>	18
<i>Ecuación 13: Cociente de Posibilidades de Diagnostico (Diagnostic Odds Ratio)</i>	18
<i>Ecuación 14: Formula de Votación</i>	20
<i>Ecuación 15: Formula de Votación</i>	26

Índice de Figuras

<i>Figura 1: Defunciones por grandes grupos de causa. Chile año 2012.</i>	2
<i>Figura 2: Menú principal Alice Sleepware®.</i>	10
<i>Figura 3: Lista de exámenes (Alice Sleepware®).</i>	10
<i>Figura 4: Lista de exámenes (Excel).</i>	11
<i>Figura 5: Vista avanzada.</i>	11
<i>Figura 6: Lista eventos.</i>	11
<i>Figura 7: Histograma distribución de eventos</i>	13
<i>Figura 8: Muestra de la matriz de características</i>	14
<i>Figura 9: Densidad de datos en matriz de características</i>	15
<i>Figura 10: Metanodo de clasificación de Variables por Random Forest</i>	15

Capítulo 2 – Introducción

Enfermedades en Chile

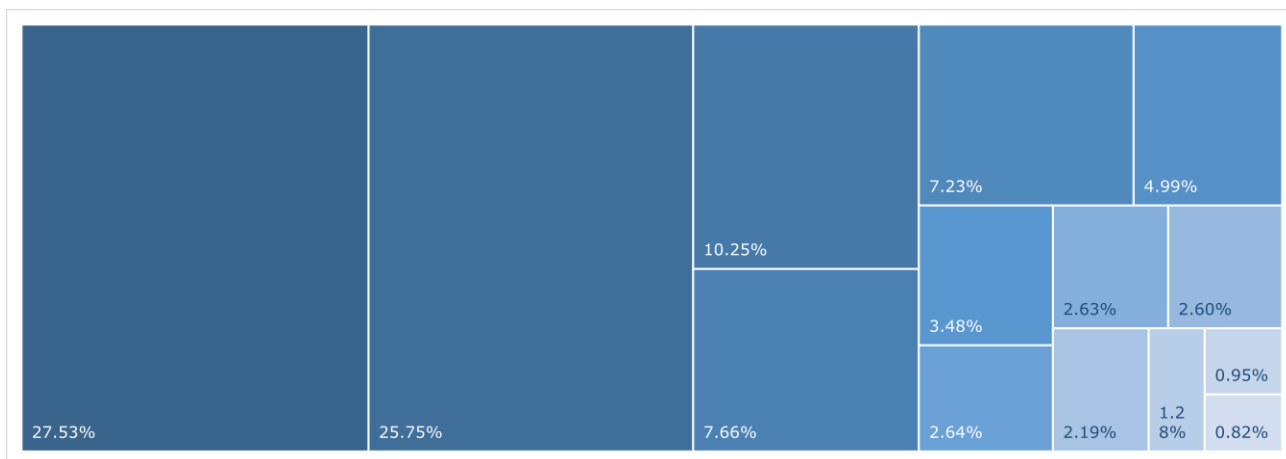
Según informa el Departamento de Estadísticas e Información de Salud (DEIS) del Ministerio de Salud (MINSAL) de Chile, en el país las principales causas de defunciones en el país son debidas a Enfermedades asociadas al sistema circulatorio, neoplasias, o sea, muertes derivadas por canceres o tumores y enfermedades del sistema respiratorio [1]. La Tabla 1 y la Figura 1 muestran el detalle de las defunciones según el MINSAL informa.

Gran Grupo de causas de muerte	Defunciones	Tasa*
Enfermedades del sistema circulatorio	27.179	156,18
Tumores (neoplasia)	25.421	146,08
Enfermedades del sistema respiratorio	10.118	58,14
Causa externa de mortalidad	7.561	43,45
Enfermedades del sistema digestivo	7.140	41,03
Enfermedades endocrinas, nutricionales y metabólicas	4.930	28,33
Total causas	98.711	567,22

TABLA 1: ALGUNOS INDICADORES DE MORTALIDAD POR CAUSA. CHILE AÑO 2012. /*TASA POR 100.000 HABITANTES

Enfermedades Respiratorias

La lista de trastornos respiratorios es extensa, resfríos, neumonías, asma, sinusitis, laringitis, distintos tipos de cáncer, entre muchas otras. Cada una de estas enfermedades conlleva distintos riesgos y deben tener cuidados específicos. Particularmente hay ciertos trastornos que disminuyen directamente el flujo de oxígeno al cuerpo, lo que trae riesgos intrínsecos y empieza a afectar el funcionamiento de otros órganos, en primera instancia aumentando la carga de sistema circulatorio que no recibe suficiente oxígeno para suministrar al resto del cuerpo. Este efecto en cadena puede terminar afectando la salud general de la persona.



- Enfermedades del sistema circulatorio
- Tumores (neoplasias)
- Enfermedades del sistema respiratorio
- Causas externas de mortalidad
- Enfermedades del sistema digestivo
- Enfermedades endocrinas, nutricionales y metabólicas
- Enfermedades del sistema nervioso
- Enfermedades del sistema genitourinario
- Síntomas, signos y hallazgos anormales clínicos y de laboratorio no clasificados en otra parte
- Trastornos mentales y del comportamiento
- Ciertas enfermedades infecciosas y parasitarias
- Resto de causas
- Malformaciones congénitas, deformidades y anomalías cromosómicas
- Ciertas afecciones originadas en el período perinatal

FIGURA 1: DEFUNCIONES POR GRANDES GRUPOS DE CAUSA. CHILE AÑO 2012.

Síndrome de Apnea e Hipopnea Obstructiva del Sueño

Dentro de los trastornos respiratorios el síndrome de apnea e hipopnea obstructiva del sueño (SAHOS) es una enfermedad que afecta al 4% de la población adulta [2] [3] y a un número similar de niños y jóvenes. Esta enfermedad se caracteriza por la suspensión de la respiración durante el sueño. Estas pausas que pueden durar, en general, más de 10 segundos, son causadas por estrechamiento de las vías aéreas superiores y pueden experimentarse desde 30 a 300 de estas pausas por noche [4] [5]. Existen distintos tipos de eventos que pueden presentarse con el SAHOS, hipopnea, apnea central, apnea obstructiva y apnea mixta.

- Las hipopneas son episodios de respiración poco profunda, en donde el flujo de aire se reduce al menos a la mitad, durante el sueño. Puede durar 10 o más segundos [6].
- Las apneas obstructivas del sueño (OSA, *Obstructive Sleep Apnea*), se caracterizan por el cese de la ventilación durante el sueño durante típicamente 20 a 40 segundos mientras aún existe esfuerzo respiratorio. Suelen estar acompañadas de ronquidos alternados con

momentos de silencio que duran entre 20 y 30 segundos y se asocian a la obstrucción de la vía respiratoria superior [7].

- Las apneas centrales del sueño (CSA, *Central Sleep Apnea*), se caracterizan por el cese repetitivo de la ventilación durante el sueño. Corresponden a una pausa de más de 10 segundos sin asociarse a ella ningún esfuerzo respiratorio. Generalmente se considera anormal presentar más de 5 de este tipo de eventos por hora. La CSA no tiene una causa particular, por lo que puede surgir de distintos síndromes [8] [9].
- Las apneas mixtas del sueño (MSA, *Mixed Sleep Apnea*), tienen componentes tanto de OSA como de CSA [6].

Un problema que se deriva directamente del SAHOS son menores índices de oxígeno en la sangre del afectado, lo que causa que el corazón deba trabajar sobre lo normal. Además, interrumpe el ciclo natural del sueño, evitando que se tenga un buen descanso [4] [9]. Una vista más profunda de los riesgos que conlleva el SAHOS nos indica que puede derivar en variados problemas cardiovasculares [9], tales como:

- Activación del sistema simpático: La activación de este sistema dado las apneas recurrentes durante el sueño afectan el periodo de vigilia. Esto implica que acciones cardiovasculares tales como aceleración del ritmo cardiaco, constricción de los vasos sanguíneos, aumento de la contracción cardiaca son propensas a ocurrir [10].
- Variabilidad cardiovascular: Se compara la actividad cardiovascular de sujetos que presentan SAHOS con sujetos de control obesos, encontrando que los pacientes con SAHOS presentan una variabilidad de ritmo cardiaco disminuida respecto a estos.
- Sustancias vaso activas: El stress causado por la interrupción del sueño aumenta la secreción de sustancias vaso activo y tóxicas que pueden causar contricción vascular que persiste por horas.
- Inflamación: Aparentemente hay una relación importante entre la inflamación sistémica y la hipoxemia o disminución de oxígeno en la sangre.
- Estrés oxidativo: La hipoxemia y re-oxigenación características en el sueño con SAHOS puede implicar que se activen mecanismos de stress oxidativo.
- Disfunción endotelial: todo lo anteriormente mencionado puede terminar en una disfunción endotelial, pese a que no es un resultado consistente.
- Resistencia a la insulina: Las patologías fisiológicas del SAOS pueden asociarse a la resistencia a la insulina, varios estudios vinculan al SAHOS con la intolerancia a la glucosa.
- Trombosis: También ha sido asociado el SAHOS con una aumentada activación de plaquetas.

Las enfermedades cardiovasculares son un común compañero del SAHOS, entre las más asociadas a este síndrome están la hipertensión (50% de los pacientes de SAHOS son hipertensos), la falla cardiaca (entre 11% y 37% dependiendo del estudio han mostrado fallas cardiacas), Paros cardiacos (Ya que solo se estudian a sobrevivientes hay un sesgo en este dato), arritmias (Hasta el 50% de los pacientes presentan arritmias), enfermedades renales de etapa final (entre 40% y 60% de los pacientes lo presentan) [9].

Diagnóstico y Exámenes

Comúnmente el diagnóstico de esta enfermedad se realiza con un examen de polisomnografía (PSG), el cual consiste en la toma de diversas señales corporales de la persona

mientras esta duerme. Es un método invasivo [5] ya que saca al paciente de su lugar habitual de sueño y le conecta diversos sensores. Entre las señales que son muestreadas están [11]:

- Flujo de aire y niveles de oxígeno en la sangre: Las señales de flujo de aire monitorean que ocurran inspiraciones y expiraciones; Junto al nivel de oxígeno en la sangre, se puede obtener una primera señal de los episodios de apnea, ya que son caracterizados por la suspensión de la respiración como se menciona anteriormente.
- Posición del cuerpo: La posición del cuerpo es monitoreada con sensores en las extremidades y tronco del paciente. Estos sensores son excitados cuando se detectan movimientos; estas señales entonces son útiles al momento de revisar ya que un movimiento brusco o constante puede ir de la mano de un déficit en la respiración, ya que el sujeto estará buscando una posición que le permita renovar el flujo de oxígeno al cuerpo.
- Electroencefalograma (EEG): permite observar la actividad neuronal del paciente, esta señal es rescatada mediante electrodos posicionados en el escalpo, este examen otorga información sobre la actividad cerebral la cual puede ser usada para detectar diversos problemas, entre ellos, desordenes del sueño.
- Electromiograma (EMG): evalúa la condición de músculos y neuronas asociadas a ellos, se obtiene mediante electrodos colocados en la piel. Un EMG relevante para las PSG es el del mentón, ya que muestra el movimiento de este y como se reacciona frente a un episodio.
- Electrooculograma (EOG): muestra el movimiento de los ojos del paciente. Se obtiene con electrodos ubicados al costado de estos; Es una herramienta útil en el discernimiento de las etapas del sueño y la comprobación que se tiene un ciclo normal de sueño.
- Electrocardiograma (ECG): muestra la actividad eléctrica del corazón. Cada latido es activado por una señal eléctrica que comienza en la parte superior y viaja hacia la inferior del corazón. El test es usado para diagnosticar enfermedades cardiacas.
- Esfuerzo respiratorio: el test de esfuerzo respiratorio muestra la actividad del tronco del individuo, sirve para complementar la información entregada por los test de flujo de aire, nivel de oxígeno y el ECG.
- Ronquidos: la escucha de ronquidos es un suplemento que ayuda a detectar apneas, ya que como el SAHOS indica en su nombre, hay episodios que son disparados por obstrucción de las vías aéreas, y esta generalmente se puede detectar a través de un ronquido.

Cabe mencionar que el uso de PSG para la detección de las etapas y enfermedades del sueño es anterior a 1970. Se tiene por ejemplo que el estándar de oro para el estudio de etapas del sueño data de 1968 [12]. Este manual indica cómo detectar visualmente los estados REM (movimiento rápido de ojos) y NREM (sin movimiento rápido de ojos) que caracterizan el ciclo normal. Este ciclo de sueño tiene una duración de 90 minutos aproximados y se recomienda completar 5 a 6 de ellos en adultos y 7 a 8 en niños. Ya que el SAHOS interrumpe el ciclo normal de sueño por lo que es importante combatirlo para un sano estado mental.

Las Apneas actualmente puede ser clasificadas por softwares especializados como Alice Sleepware®, de la línea Respiroics® de Philips®, las cuales tienden a tener una tasa alta de error del tipo falso positivo y falso negativo en niños, ya que están calibrados para su diagnóstico en adultos, además [13] muestra que los algoritmos actuales no dan resultados positivos al ser aplicados a PSG hechas en estos. Esto significa que a posteriori, un experto debe revisar y clasificar nuevamente el reporte que hace este software. Esta revisión se realiza manualmente revisando visualmente las distintas señales. Debido a que las PSG tienen una duración de entre 6 a 10 horas, este proceso de revisión y reclasificación se hace extenso, y depende fuertemente de la experiencia del médico que realiza esta labor.

E-Health

Como lo describe el *Journal of Internet Research*, *E-Health* es un concepto que surge de la intersección de la medicina con la informática. Se refiere a la entrega de servicios e información médica entregada o realizada por las tecnologías de la información. Más ampliamente el termino define el desarrollo técnico, una manera de pensar y un objetivo de pensamiento global para mejorar la salud localmente, regionalmente y a lo ancho del mundo usando tecnologías de la información y comunicación [14].

Hipótesis

Se plantea bajo estas primicias la siguiente hipótesis:

“Los estadísticos contienen información suficiente para realizar clasificación de los episodios del SAHOS.”

Un estadístico, según lo define el *McGraw-Hill Dictionary of Scientific and Technical Terms*, es una estimación o porción de datos, referente a algún parámetro, obtenida de un muestreo [15]. Así los momentos estadísticos, o sea, los valores esperados para las distribuciones de probabilidades [15], son un tipo de estadístico, particularmente se utilizan en este trabajo los momentos promedio, desviación estándar, varianza, asimetría y curtosis.

Objetivos

Objetivo General

El objetivo específico de este trabajo es diseñar un clasificador capaz de reconocer los distintos episodios de SAHOS, a través de datos que proporcionan los estadísticos elegidos.

Objetivos Específicos.

Como objetivos específicos se plantean:

- utilizar técnicas de reducción de variables para disminuir potencialmente el número de sensores utilizados en el examen de PSG.
- Probar el cociente de posibilidades de diagnóstico (Ecuación 13) como criterio de diseño de clasificadores.
- Programar un extractor de características en MATLAB® que facilite la extracción de características de los exámenes de PSG.
- Construir una base de datos que contenga la información necesaria para entrenar clasificadores para el SAHOS.
- Crear una plataforma en KNIME® que permita probar el desempeño de distintos clasificadores.

Metodología

Para lograr estos objetivos, y comprobar la hipótesis planteada se procede de la siguiente manera; Se obtendrán los datos crudos de los exámenes de PSG desde el centro médico Lira de la red UC Christus, los que serán procesados con una serie de programas para extraer tanto los episodios de SAHOS relevantes como los datos estadísticos de los mismos.

Luego de extraer estos datos, se entrenan una serie de clasificadores en un software que será capaz de entregar métricas que servirán para ver el desempeño de estos. Finalmente se construye

un algoritmo, que toma en cuenta las decisiones de los tres clasificadores con mejor desempeño y mediante votación clasifica un episodio como perteneciente a algún trastorno del SAHOS.

Uso de Estadísticas en Clasificación

La utilización de estadísticos, como variable de entrenamiento para clasificadores en general, es común en estudios científicos dado que la información proveniente de señales puede tener mucha varianza entre dos muestras que representen un mismo evento [16]. La manera en que los estadísticos condensan información relevante a una señal se utiliza para caracterizar una señal, tanto en el dominio del tiempo como en el de las frecuencias, reduciendo está a solo unos cuantos elementos característicos, en nuestro caso promedio, desviación estándar, asimetría, curtosis y correlación.

Clasificación de Señales

Las señales en general pueden ser tratadas por procesos matemáticos para extraer información que no es posible de reconocer fácilmente, ya sean sus momentos estadísticos (desviación estándar, asimetría curtosis, correlación) su comportamiento (tasa de cambio, energía acumulada) sus componentes de frecuencia, entre otros. Estos procesos aportan información valiosa la cual puede ser usada para una mejor clasificación de varios tipos de eventos, varios trabajos centran sus esfuerzos en poder obtener información relevante de señales EEC, EEG o EMG para el diagnóstico de enfermedades como epilepsia o arritmias [17] [18] [19]. Cabe notar que la clasificación de señales no es una técnica exclusiva para medicina, sirve para afrontar problemas en diversas áreas, como biometría [20], ingeniería [21] [22], geología [23], entre otras.

Los momentos estadísticos de alto orden, como la asimetría y curtosis, han probado su utilidad en variados campos; sonar, radar, sismología, reconstrucción de imágenes, estimación de atrasos temporales, filtros adaptativos, entre otros [24].

De acuerdo a Mendel [25] momentos son útiles especialmente en problemas donde el ruido no-gaussiano, sin fase mínima, coloreado o no lineal son importantes y deben ser considerados, debido a que son ciegos a cualquier tipo de proceso gaussiano, mientras que la correlación y los métodos basados en ella, no lo son. Entonces permite aumentar el radio señal a ruido en procesos contaminados por ruido coloreado o blanco. Esta propiedad permite trabajar procesos que no son lineales y tienen presencia de ruido no-gaussiano, y los procesos del mundo real pertenecen ciertamente a este grupo.

Dominio Temporal y Densidad Espectral

Normalmente un sensor trabaja obteniendo datos durante cierto periodo temporal, es natural entonces almacenar estos datos como una magnitud contrapuesta a una escala temporal. Esto es lo que se conoce como una señal en el dominio temporal. La información que se puede obtener por simple observación en una señal de este estilo es limitada sin ayuda de herramientas computacionales, y gracias a las ventajas del procesamiento de datos, se puede obtener información más completa para caracterizarla.

El espacio de las frecuencias tiene la ventaja de entregar información que a simple vista es invisible al ojo no entrenado. Ha sido ampliamente utilizado para obtener información de señales en diversos campos. Entre los métodos para obtener esta información están el uso de la transformada de Fourier o la transformada Wavelet. Cada una tiene ventajas comparativas y

entregan información que puede ser relevante. Podemos ver ejemplos de su uso en ECG [26] y EEG [27] con resultados favorables.

Para el estudio de señales en el espacio de las frecuencias suele utilizarse la transformada rápida de Fourier (FFT) que consiste en una técnica que, con ciertas limitaciones, logra con baja exigencia computacional, entregar la densidad espectral de una señal, esto es, la potencia contenida en cada una de las frecuencias que la señal contiene.

Técnica Propuesta

Se empleará una técnica que no se ha encontrado en la literatura revisada para procesar y analizar los datos. En gran parte seguirá una metodología de diseño estándar para una clasificación a partir de características, con el objetivo de mantener la tasa de falsos negativos menor al 10%.

Se pueden diferenciar distintas etapas para la realización del estudio. En primer lugar, se extraen las características relevantes de los exámenes de PSG. Segundo, se entrena una serie de clasificadores con los datos de episodios ya clasificados por un experto. Finalmente se validan estos modelos y se diseña un clasificador compuesto por aquellos que logren mejores resultados según los criterios que más adelante se detallan.

Adquisición de Datos

Los datos son recolectados en el centro médico San Jorge, parte de la red de salud UC Christus de la Pontificia Universidad Católica de Chile, donde un técnico examina al paciente con ayuda de una máquina, que, junto a un software especializado, es capaz de recolectar datos de múltiples sensores. Este equipo es conocido como un polisomnógrafo. Los datos son recolectados durante un examen de PSG, el cual tiene una duración igual a una noche de sueño del paciente, entre 6 y 10 horas. Cabe mencionar que este examen (Código FONASA¹ 1101045) tiene un costo entre los 300.000 y 700.000 pesos chilenos [28] [29] [30].

Los datos de la PSG se encuentran en forma de archivos generados por una máquina propietaria de Philips® y almacenados en un formato privado. Existe la capacidad de importar estos datos a un formato libre, *European Data Format* (EDF), que se define como “un formato simple y flexible para el intercambio y almacenamiento multicanal de señales físicas y biológicas” [31]. EDF no es capaz de almacenar las observaciones hechas por el médico ni tampoco las realizadas por el software propietario, solo presenta las señales recolectadas.

Las PSG recolectadas corresponden a los exámenes de 90 pacientes en diversos rangos etarios y con distintos episodios de SAHOS registrados. Debido a las dificultades que implica el estudio de SAHOS en niños, se seleccionan aquellos mayores de 18 años, 70 personas.

Mediante Alice Sleepware ® se pueden exportar los datos que el técnico recolecta a un formato no propietario, *Comma-Separated Values* (CSV), el cual entrega los datos como una serie de listas ordenadas. Este formato es compatible con MATLAB®, en donde se obtendrán los datos estadísticos de las señales.

Pre-Procesamiento

Para el pre-procesamiento de los datos se utiliza el software MATLAB® junto a un script (blockEdfLoad.m) que permite leer de manera simple los archivos EDF, que sumado a la capacidad

¹ Código de identificación para prestaciones médicas en Chile.

nativa del programa para trabajar con CVS, permite una expedita lectura de los archivos. Con la ayuda del toolbox JSONlab [32], se desarrolla entonces un programa que vincula los datos del CVS con las señales del EDF, rescatando medidas estadísticas de los episodios en archivos en formato JSON (*JavaScript Object Notation*), archivos de texto de bajo peso que ofrecen facilidades en su lectura, tanto a humanos como máquinas [33].

A continuación, se escriben los datos estadísticos en una tabla que puede ser leída por el software KNIME®, el cual permite aplicar distintos métodos de clasificación los cuales vienen ya implementados.

Procesamiento

Con los datos ya cargados en KNIME® se entrenan distintos clasificadores para hacer un estudio comparativo entre ellos y así poder elegir el más apto para detectar cada uno de los distintos tipos de apneas (hipopnea, CSA, OSA y MSA). En paralelo se realizan técnicas de reducción de variables para disminuir el tiempo de computación, tanto en el entrenamiento de los clasificadores como en la clasificación de los episodios [34].

Capítulo 3 – Implementación

Herramientas Hardware

Se utiliza como herramienta un equipo proporcionado por el *Optical and Wireless Laboratory* (OWL) de la Universidad de Chile para el desarrollo de este proyecto. El equipo es una máquina de 64-bits corriendo *Windows 7 Professional, Service Pack 1*. Sus características de Hardware son:

- Procesador: Intel Core I7 4770 (Haswell, cuarta generación) @ 3.40GHz de cuatro núcleos y ocho hilos
- Memoria: 16GB DDR3
- Video: NVIDIA Quadro K600 + Intel HD Graphics 4600
- Placa Madre: Hewlett-Packard 1998

Herramientas de Software

Alice Sleepware®

Software desarrollado por Philips® para su línea Respironics®, especializado en el diagnóstico de enfermedades respiratorias. Incluye interfaz que permite la revisión de exámenes y la detección automática de eventos respiratorios. Actualmente es utilizado por los centros médicos de la red UC Christus para facilitar la labor de diagnóstico de SAHOS en pacientes [35].

MATLAB®

Plataforma matemática, propietaria de MathWorks, que ofrece un entorno de desarrollo integrado (IDE) ajustado para trabajos iterativos científicos e ingenieriles. Posee herramientas que permiten la exploración de data y su procesamiento, además de una extensa librería de códigos dispuestos por usuarios para usuarios a través de una comunidad en línea (*MathWorks File Exchange*).

El lenguaje propio de MATLAB®, “lenguaje M”, permite una rápida y comprensiva programación. Facilita además la programación orientada a objetos, el trabajo de matrices y vectores, creación de gráficos en 3D y 2D, uso de estructuras complejas de datos, entre varias otras funciones [36]

KNIME® Analytics Platform

Plataforma de código abierto de análisis de data, reporte e integración [37]. Permite la creación de flujos de trabajo de manera expedita con la ayuda de módulos especializados en labores específicas. Al igual que un lenguaje de programación, los módulos ofrecen la capacidad de realizar tareas, tanto en paralelo como en serie sobre una base de datos. Los resultados pueden observarse en el mismo programa en forma de tablas o gráficos generados a petición del usuario.

Implementación de la Técnica Propuesta

Como se menciona anteriormente, de los exámenes, en el programa Alice Sleepware®, son rescatadas tanto las señales de los distintos sensores como la información que ha incluido el software y corroborado el medico sobre la naturaleza de los episodios encontrados. Posterior a este proceso se implementan tres programas de MATLAB®; El primero separa los datos exportados en

EDF en distintos archivos, rescata información estadística y metadatos de ellos, el segundo los acumula en una matriz y el tercero escribe esta matriz en un archivo manejable por KNIME®.

Paso 1 - Extracción de datos de Alice Sleepware®

Con los exámenes pre-cargados en la maquina se inicializa Alice Sleepware®, en pantalla se desplegara el menú principal (Figura 2) Figura 2: Menú principal Alice Sleepware®, en el cual se elige la opción *View and Validate* (Ver y validar) con lo que aparecerá en pantalla la lista de exámenes con metadatos del paciente examinado (Figura 3).

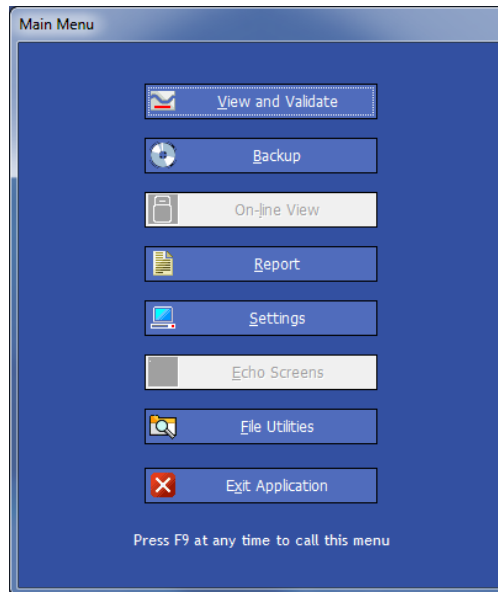


FIGURA 2: MENÚ PRINCIPAL ALICE SLEEPWARE®.

Last name	First name	Gender	Birth date	Acq Date/Start time	Patient type	Acq Nb	Acq D...	Acq Type	Patient Id	Comment
		F		29-04-2016 22:24:11	Adult	92	526	Alice 5		SAHOS.
		M		25-04-2016 21:12:42	Adult	255	560	Alice 5		EN ESTUDIO
		F		30-04-2016 21:56:15	Adult	259	489	Alice 5		EN ESTUDIO
		M		05-05-2016 22:20:42	Adult	260	490	Alice 5		ENFERMEDAD DE STEINERT.-
		M		08-08-2013 22:09:57	Adult	423	547	Alice 5		EN ESTUDIO.-
		M		09-12-2015 21:46:30	Adult	735	595	Alice 5		TRASTORNO DEL SUEÑO.-
		M		11-12-2015 22:03:54	Adult	743	503	Alice 5		HIPERSOMNIA DIURNA
		F		16-12-2015 23:10:08	Adult	756	443	Alice 5		TRAST. DEL SUEÑO DE LARGA DATA.-
		M		20-12-2015 21:31:05	Adult	770	591	Alice 5		OBS. APNEA OBS. DEL SUEÑO - HTA -
		M		27-12-2015 21:53:26	Adult	782	548	Alice 5		ALTERACION DEL SUEÑO
		M		05-01-2016 21:56:27	Adult	810	552	Alice 5		OBS. SAHOS
		M		14-01-2016 22:23:32	Adult	825	511	Alice 5		TRASTORNO DEL SUEÑO, HIPERSOMNIA DIURNA
		M		12-02-2016 22:04:13	Adult	869	145	Alice 5		SD. DOWN-OBESIDAD-OBS.ADS

FIGURA 3: LISTA DE EXÁMENES (ALICE SLEEPWARE®).

Al hacer clic derecho sobre esta pantalla se despliega la opción *Save/Export* (guardar/exportar) que permite guardar la lista como un archivo CSV. Paso siguiente, se anonimiza la lista abriendo el archivo CSV en Excel y eliminando las columnas *Last Name*, *First Name* y *Patient Id*, que contienen los apellidos, el nombre y el RUN de cada individuo. Se crea una columna *EDF Name* que contiene el nombre del archivo EDF con el que se debe vincular esta metadata y se reemplaza la columna *Acq. Date/Start time* (fecha adquisición/hora de inicio) por una columna que contiene la fecha del examen, una que contiene la hora del examen y tres que separan la hora

en componentes de hora, minutos y segundos y es guardada como archivo de Excel (Figura 4) bajo el identificador “ListaExámenes.xlsx”.

EDF Name	Gender	Birth date	Acq Date	St Hour	St Min	St Sec	Start Time	Age	Patient type	Acq Nb	Acq Dur [min]	Acq Type	Comment
A0009817	M		15-04-2016	21	8	42	21:08:42	67	Adult	9817	600	Alice 5	OBS. APNEAS DEL SUEÑO
A0017342	M		22-04-2016	22	30	47	22:30:47	44	Adult	17342	480	Alice 5	TRAST. RESP. DEL SUEÑO.-
A0003148	M		15-02-2016	22	26	49	22:26:49	27	Adult	3148	503	Alice 5	OBS SAHOS
A0000879	M		20-02-2016	21	52	24	21:52:24	46	Adult	879	549	Alice 5	SAHOS EN ESTUDIO OBESIDAD
A0000092	F		29-04-2016	22	24	11	22:24:11	42	Adult	92	526	Alice 5	SAHOS.
A0017315	F		16-02-2016	21	31	33	21:31:33	61	Adult	17315	582	Alice 5	RONCOPATÍA - OBS. SAHOS - INSOMNIO 2 DA
A0000878	F		19-02-2016	21	41	5	21:41:05	56	Adult	878	531	Alice 5	RONCOPATÍA
A0004494	F		13-08-2013	22	13	54	22:13:54	70	Adult	4494	525	Alice 5	SAHOS- RONCOPATIA
A0017311	M		10-02-2016	21	33	58	21:33:58	51	Adult	17311	1440	Alice 5	SAHOS.-
A0002371	M		04-04-2016	23	44	10	23:44:10	19	Adult	2371	492	Alice 5	TITULACIÓN DE BIPAP DE NOCHE COMPLETA.-
A0001002	M		28-04-2016	22	36	26	22:36:26	28	Adult	1002	494	Alice 5	OBS. SAHOS.
A0002389	F		06-05-2016	22	41	4	22:41:04	29	Adult	2389	500	Alice 5	BRUXISMO - INSOMNIO CRÓNICO

FIGURA 4: LISTA DE EXÁMENES (EXCEL).

Paralelamente se deben exportar los exámenes a archivos EDF, para esto se hace doble clic sobre una de las entradas de la lista que presenta Alice Sleepware® (Figura 3). Esto despliega una vista en avanzada del examen (Figura 5), en donde se pueden explorar detalles, observar las distintas señales y eventos que se encontraron durante la toma de este. Haciendo clic en *File* en esta pantalla se despliega un menú, se debe elegir *Export* y luego *Export to EDF Format* para exportar las señales de los sensores al formato EDF con el que se trabajara.

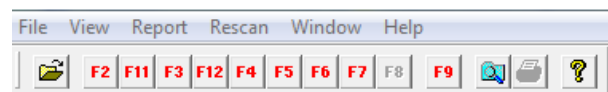


FIGURA 5: VISTA AVANZADA.

Para ver la lista de eventos que detectó el software y aquellos validados por el médico, se hace clic en el botón F2 desplegado en la pantalla (Figura 5), y al hacer clic derecho en esta lista (Figura 6) podemos elegir la opción *Save/Export...* para exportar esta lista a formato CSV.

Event type	Stage	Time	Epoch	Date	Duration	Validation
Central apnea	N2	22:21:50	139	25-04-2016	12 (11)	-
Central apnea	N2	0:49:49	435	26-04-2016	13.5 (12)	-
Central apnea	N2	1:08:49	473	26-04-2016	17 (14.5)	-
Central apnea	N2	3:18:31	732	26-04-2016	10.5 (10)	-
Central apnea	N2	3:19:20	734	26-04-2016	11 (11)	*
Central apnea	N2	3:34:21	764	26-04-2016	17	@
Central apnea	N2	3:34:23	764	26-04-2016	10.5 (9.5)	-
Central apnea	N2	4:01:56	819	26-04-2016	14.5 (13)	-
Central apnea	N2	4:21:23	858	26-04-2016	10 (9.5)	-
Central apnea	N2	4:52:45	921	26-04-2016	17.5 (14)	-
Central apnea	N2	4:58:07	931	26-04-2016	16.5 (16.5)	-
Central apnea	N2	5:21:42	979	26-04-2016	12 (11)	-
Obstructive apnea	N3	22:45:35	186	25-04-2016	26.5	-

FIGURA 6: LISTA EVENTOS.

Para cada examen se debe exportar el archivo EDF y la lista de eventos en CSV, y luego mover todos los archivos CSV y “ListaExámenes.xlsx” en la carpeta *DataHarvest* y los EDF en la carpeta *edf* las cuales se deben encontrar en el mismo directorio que los scripts de MATLAB®.

Paso 2 - EDF a JSON

Con los archivos ya en los lugares indicados se inicia MATLAB®, y se ubica el *workspace* en el directorio donde se trabaja y se ejecuta “*paso1_edf2json.m*”. Este script carga a memoria la lista de los exámenes almacenada en “*ListaExámenes.xlsx*”, y se cruza la información que contiene la lista con las señales en el archivo EDF y la data de los episodios guardada en el CSV. De esta manera, se crea un archivo JSON para cada episodio.

El script, luego de cargar los datos del episodio a memoria, procede a tomar cada señal en el dominio temporal, calcular su integral y derivada. Después calcula su FFT y vuelve a calcular integral y derivada en el dominio de las frecuencias.

Sobre estas seis señales se aplican los estadísticos: promedio, desviación estándar, varianza, asimetría y curtosis. De esta manera se obtienen cinco datos de cada señal, y, ya que cada examen de PSG está definido por los datos que entregan entre 20 y 30 sensores, se obtienen entre 600 y 900 características de cada episodio.

$$N^{\circ} \text{ Sensores} \times N^{\circ} \text{ Señales} \times N^{\circ} \text{ Estadísticos} = N^{\circ} \text{ Características}$$

ECUACIÓN 1: NÚMERO DE CARACTERÍSTICAS

Todas estas características, sumadas al segmento de las señales, metadatos del episodio y paciente (duración del episodio, edad y género del individuo, que tipo de evento es y si el evento fue clasificado por software o fue clasificado por un experto), son almacenados en un archivo JSON.

Paso 3 - JSON a Matriz

El siguiente script “*Paso2_json2mat.m*” se encarga de recoger los JSON creados por el programa anterior, y transcribir cada una de las características a una tabla, la cual es guardada para que la siguiente tarea trabaje.

Paso 4 - Matriz a Texto

El tercer script, “*Paso3_mat2txt.m*” recupera los datos del segundo y los escribe en un archivo de texto plano llamado “*data.txt*”, este archivo puede ser procesado por KNIME®.

Paso 5 – Grupo de Control

Para probar a los clasificadores es necesario tener un grupo de control. Debido a que no se cuentan con exámenes realizados a personas sanas, se propone obtener los datos de sueño normal en los exámenes que se tienen disponibles.

Para realizar esto se observan los momentos en los cuales los pacientes no presentan eventos respiratorios por más de 30 segundos, y se utiliza un programa similar al presentado en “Paso 2 - EDF a JSON” para obtener las señales sin eventos. Estas se rigen por las reglas:

- Están 15 segundos antes y 15 segundos después de cualquier evento marcado por Alice Sleepware®.
- Tienen duración de entre 10 y 40 segundos, elegida al azar.

Luego, siguiendo la lógica detrás de los algoritmos descritos en Paso 3 - JSON a Matriz y Paso 4 - Matriz a Texto se rescata la data sin eventos y se deja en un formato legible por KNIME®. Los programas son llamados “Paso1_edf2json_Healthy.m”, “Paso2_json2mat_Healthy.m” y “Paso3_mat2txt_Healthy.m” respectivamente.

Paso 6 - Conjuntos de Entrenamiento, Prueba y Validación

Para entrenar, probar y validar los clasificadores se utilizan las PSG de 90 pacientes, se seleccionan a los mayores de 18 años (70 individuos) y se obtienen de ellos 10.198 eventos. Por otro lado, el grupo de control se construye con 7.431 tomas de datos clasificadas como “sin evento”. Estas se recogen de los mismos 70 individuos usando el método explicado en la sección anterior.

Se realizan 3 pruebas con KNIME®, la primera sobre el conjunto de eventos (10.198 datos), la segunda sobre el grupo de eventos más datos “sin evento” (17.629 datos) y la última sobre los eventos y una fracción de los datos clasificados como “sin evento”, tal que el número de estos sea similar al tipo de evento con mayor representación, apneas obstructivas (Alrededor de 14.800 datos).

Estos eventos se distribuirán de manera aleatoria en tres conjuntos, los cuales corresponden a: de entrenamiento (60% de los datos), de prueba (20%) y de validación (20%).

Notas sobre los eventos

Cabe destacar la distribución de los datos que se manejan no están equilibrados. Estos presentan dos clases sub-representadas (Figura 7), lo que significará que los clasificadores que se entrenen para estas no sean caracterizados con la misma precisión que las otras.

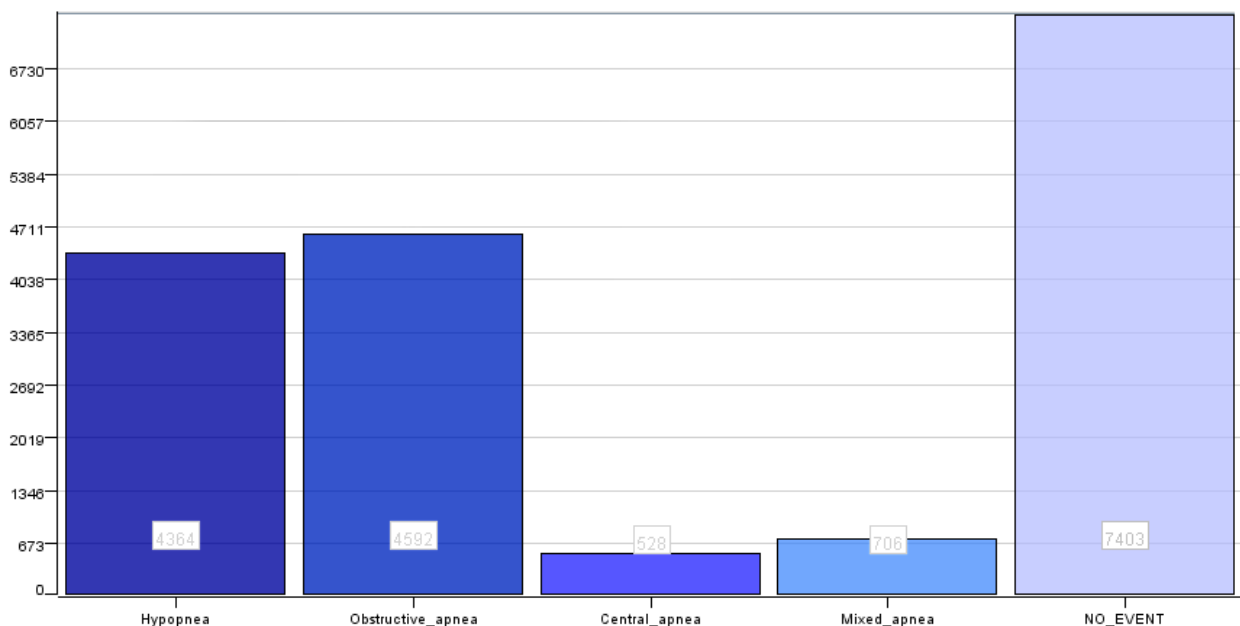


FIGURA 7: HISTOGRAMA DISTRIBUCIÓN DE EVENTOS

Paso 7 - Línea Base

En este trabajo se define como línea base una prueba inicial que se realiza utilizando varios clasificadores sobre la totalidad de las características de los episodios, para luego comparar la eficiencia de estos frente a la clasificación con reducción de variables ofrecidas por los algoritmos descritos en la siguiente sección.

El objetivo de este ejercicio es encontrar un conjunto reducido de características que permitan una clasificación de igual o mejor calidad que la que se obtiene con la totalidad de las características. Así se logra determinar cuáles son los sensores que son necesarios para construir un polisomnógrafo capaz de entregar un reporte confiable sobre los eventos que definen al SOHAS.

Paso 8 - Reducción de Características

Cada examen de PSG usa distintos sensores para la captura de datos, el conjunto de estos no es estándar por lo que, en el total de los 70 exámenes, se extraen 1.684 características y 5 metadatos (paciente, hora del examen, hora del evento, naturaleza del evento y tipo de validación). Estos datos se pueden representar como una matriz, en donde cada columna representaría una característica y cada fila un episodio, una muestra de esta matriz puede observarse en la Figura 8.

S Patient	i Age	S Gender	S Exam_time	S ▲ Event...	S Event_time	i Duration	S ▼ Stage	S Validation	D EEG_F3...	D EEG_F3...	D EEG_F3...	D EEG_F3...	D EEG_F3...	D EEG_F3...
A0003148	27	M	15-Feb-2016 22:26:49	Central_apnea	16-Feb-2016 06:44:51	29	W	*	?	?	?	?	?	?
A0017353	72	F	04-May-2016 21:12:37	Central_apnea	05-May-2016 01:17:13	21	W	*	?	?	?	?	?	?
A0017367	54	M	19-May-2016 22:30:19	Central_apnea	20-May-2016 04:03:51	11	W	*	?	?	?	?	?	?
A0000770	57	M	20-Dec-2015 21:31:05	Hypopnea	20-Dec-2015 23:17:08	16	W	*	-1,048.344	2,276.499	5,182,446.808	0.718	2.69	-2.245
A0000770	57	M	20-Dec-2015 21:31:05	Hypopnea	20-Dec-2015 23:31:49	11	W	*	-1,823.41	3,701.543	13,701,420.26	0.492	1.982	-1.949
A0000770	57	M	20-Dec-2015 21:31:05	Hypopnea	21-Dec-2015 00:17:42	12	W	*	-1,767.749	807.257	651,664.123	0	2.475	-3.14
A0000975	30	M	14-Apr-2016 22:45:21	Hypopnea	14-Apr-2016 23:55:57	10	W	*	-2,381.301	2,477.07	6,135,876.636	-1.45	3.842	0.674
A0001664	48	M	08-Aug-2013 21:45:10	Hypopnea	09-Aug-2013 00:58:52	37	W	*	?	?	?	?	?	?
A0001664	48	M	08-Aug-2013 21:45:10	Hypopnea	09-Aug-2013 02:14:23	23	W	*	?	?	?	?	?	?
A0003230	82	M	07-May-2016 22:13:52	Hypopnea	07-May-2016 22:50:00	14	W	*	?	?	?	?	?	?
A0003253	78	F	23-May-2016 22:39:35	Hypopnea	24-May-2016 01:38:49	22	W	*	?	?	?	?	?	?
A0009761	38	M	26-Feb-2016 22:21:21	Hypopnea	27-Feb-2016 02:47:58	12	W	*	-87.376	3,409.469	11,624,480.92	-1.935	6.058	-1.306
A0009778	49	M	13-Mar-2016 22:15:53	Hypopnea	14-Mar-2016 02:56:28	10	W	*	-3,777.61	7,873.952	61,999,124.65	-1.236	3.15	1.642
A0017314	20	M	15-Feb-2016 21:45:07	Hypopnea	16-Feb-2016 04:21:09	22	W	*	?	?	?	?	?	?
A0017330	52	M	29-Mar-2016 21:41:56	Hypopnea	30-Mar-2016 02:57:33	14	W	*	?	?	?	?	?	?
A0017330	52	M	29-Mar-2016 21:41:56	Hypopnea	30-Mar-2016 03:35:01	11	W	*	?	?	?	?	?	?
A0003230	82	M	07-May-2016 22:13:52	Mixed_apnea	08-May-2016 00:09:32	27	W	*	?	?	?	?	?	?
A0009778	49	M	13-Mar-2016 22:15:53	Mixed_apnea	14-Mar-2016 00:46:25	10	W	*	991.639	496.778	246,788.691	-0.904	2.31	0.951
A0000879	46	M	20-Feb-2016 21:52:24	Obstructive_...	20-Feb-2016 22:45:36	22	W	@	7,718.63	5,043.594	25,437,843.59	0.08	1.815	6.988
A0003144	61	M	11-Feb-2016 22:07:42	Obstructive_...	12-Feb-2016 00:16:24	12	W	@	-3,295.339	5,715.237	32,663,938.41	-2.699	9.085	0.418

FIGURA 8: MUESTRA DE LA MATRIZ DE CARACTERÍSTICAS

Un gran volumen de datos influye directamente en el desempeño de los programas, así como también hay datos que no aportan significativamente en lo que se desea hacer [37] [38], El equipo de KNIME® explora distintos métodos para mejorar el rendimiento de distintos clasificadores mostrados por por KNIME® [37] de los cuales se rescatan los que mejores resultados en cuanto al cociente entre la inversión tiempo y mejora de exactitud según su estudio.

Además, al aplicar clasificadores, estos generan problemas con los valores vacíos o de valor “No Numérico” (*NaN*), es por esto que se debe aplicar un método para tratarlos. En el caso de este trabajo, se aplica un reemplazo por valor cero a los valores *NaN* o no encontrados.

Dado a que el conjunto de sensores no es estándar, existe una cierta cantidad de características sin data. Particularmente, el 49,89% de los datos trabajados son clasificados como “no es un número” o vacío. La Figura 9 muestra la densidad de datos en los exámenes que se manejan en este proyecto. En color azul se ve donde no hay data y en celeste donde si hay. El eje horizontal representa los eventos y el vertical las características.



FIGURA 9: DENSIDAD DE DATOS EN MATRIZ DE CARACTERÍSTICAS

Random Forest / Tree Ensemble

Random Forest (RF) o *Tree Ensemble* (TE) son distintos nombres que se le asignan a las técnicas de clasificación que consisten en la generación de múltiples árboles de decisión binarios. Su fin es descubrir cuáles son las características que generan una mejor división de las clases con las que se trabaja [39] y luego, utilizar información estadística para seleccionar el conjunto de características que entregan mayor información [40] [41].

Utilizando el algoritmo propuesto por KNIME® en [37] como base, se define el sistema descrito por los nodos en la Figura 10. Este sirve para la selección de características que mejor definen las diferencias entre clases de episodios para el conjunto de datos que se manejan.

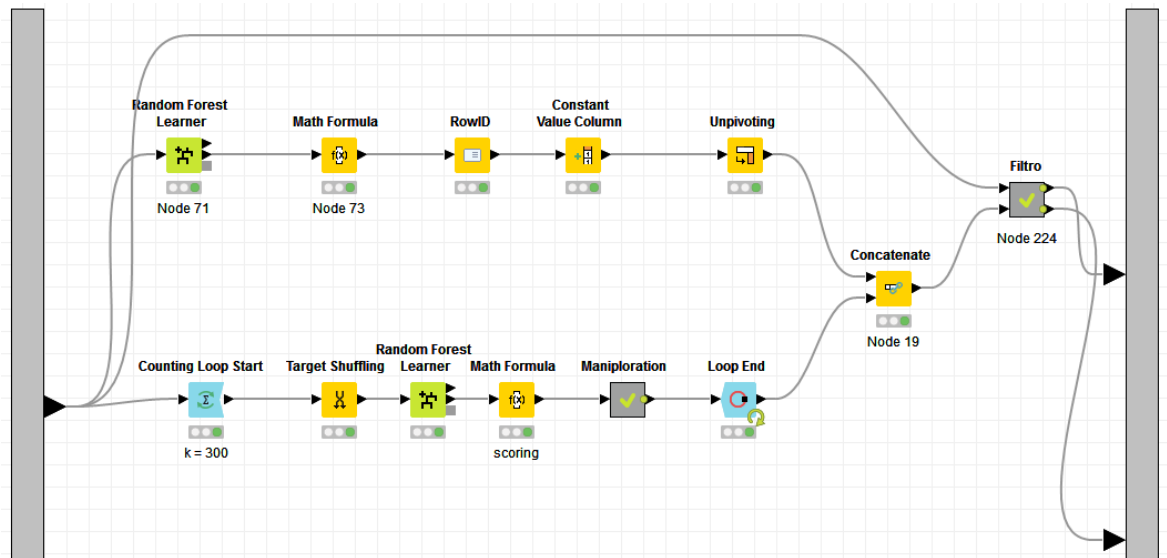


FIGURA 10: METANODO DE CLASIFICACIÓN DE VARIABLES POR RANDOM FOREST

Específicamente se genera un bosque de árboles. Se entrena de manera individual a 5.000 árboles, de dos niveles de profundidad, en 3 características.

Una vez generado el bosque, se puntúan las características según cuantas veces han sido seleccionadas en las bifurcaciones y en qué nivel (entre más bajo su nivel, es mejor la diferenciación que produce). Luego se compara con el número total de candidatos para ese nivel en el arreglo. Se puede observar la formula en la Ecuación 2, que indica que tan predictiva es la característica relativa a las demás.

$$Puntaje = \frac{N^{\circ} Apariciones nivel 0}{N^{\circ} Candidatos nivel 0} + \frac{N^{\circ} Apariciones nivel 1}{N^{\circ} Candidatos nivel 1}$$

ECUACIÓN 2: PUNTUACIÓN ARBOLES DECISIÓN

Este proceso se ejecuta iterativamente en 300 instancias. Las características son alimentadas de manera aleatoria a los árboles y los puntajes son almacenados en memoria. Se obtiene el promedio y la desviación estándar de los puntajes de las características y se filtran según la regla descrita en la Ecuación 3. De esta manera, las que cumplan la condición impuesta se ignoran.

$$puntaje\ base < (Promedio\ puntaje + 1.5 \times desviacion\ estandar\ puntaje)$$

ECUACIÓN 3: REGLA FILTRO DE CARACTERÍSTICAS

Cabe mencionar que KNIME® presenta dos implementaciones del algoritmo, *Random Forest* y *Tree Ensemble*. TE presenta una interfaz más simple que RF, que tiene una mayor variedad de opciones para ajustar el modelo. Para cada prueba los algoritmos reducen en distinta proporción las variables.

Paso 9 - Entrenamiento de Clasificadores

Para entrenar los clasificadores se alimentan con el 60% de los datos recopilados para cada experimento. Los clasificadores utilizados son los siguientes:

Árbol de decisión

Clasificación por árboles de decisiones (DT) es un método que utiliza un árbol de decisión como modelo predictivo, a partir de las características de un objeto crea un mapa para llegar a la clase que pertenece. Cada nodo representa una decisión que se hace en base a la característica definida, y cada rama representa una decisión. El nodo final de una serie de ramificaciones identifica la clase clasificada [40] [42].

Naive Bayes

Naive Bayes (NB) es una familia de clasificadores que utiliza probabilidades y el teorema de Bayes [43], que de manera ingenua (de aquí que sea *Naive*) asume independencia entre las variables. Este clasificador asigna la clase más probable a la que pertenece un vector de características [44].

Reglas Difusas

Un clasificador por Reglas Difusas (FR, *Fuzzy Rules*) es un modelo lingüístico que utiliza reglas difusas. Este modelo consta de tres partes, una serie de reglas “**IF this THEN that**” (Si esto pasa **entonces** hago esto otro), una base de datos y un motor de inferencia que permite interpretar las distintas reglas para llegar a la clase final de clasificación [45].

Support Vector Machine

Support Vector Machine (SVM) es un algoritmo que busca un hiperplano que logre realizar una división binaria sobre los datos. En el contexto de este trabajo, se considera a cada evento como un vector en el cual sus características representan distintas dimensiones. SVM busca crear un hiperplano que logre maximizar la distancia entre el mismo y los puntos de data más cercanos [42].

Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) es una red neuronal compuesta de una capa de ingreso, varias capas ocultas y una capa de salida. Todas formadas por múltiples neuronas computacionales. La señal a clasificar se propaga a través de la red en una dirección capa por capa. Las neuronas son activadas según las reglas aprendidas durante el entrenamiento de la red e intentan mapear cada señal de entrada a una clase, representada por alguna de las neuronas en la capa de salida. Las capas ocultas son las encargadas de hacer que la red aprenda tareas al extraer las características más significativas para la clasificación [46].

Probabilistic Neural Network

Probabilistic Neural Network (PNN) es un modelo de red neuronal que en lugar de usar heurística para generar los patrones que sigue su clasificación, utiliza funciones derivadas estadísticamente para generar su red interna. Esta función se aproxima a la superficie óptima que genera un NB [47].

Paso 10 - Prueba

Luego de entrenar los clasificadores se realiza una prueba de su validación con del 20% de los episodios. En KNIME® esto significa cargar los datos generados por el nodo de modelo al nodo de predictor y recoger las estadísticas de clasificación.

Ya que se entrenaron separadamente los clasificadores para cada tipo de episodio, cada clasificador podrá decidir si es que se presenta el evento para el cual fue entrenado o si no se presenta. Esta clasificación entrega una matriz de confusión en la que podemos identificar la eficiencia del clasificador a partir de la cuenta de:

- Verdaderos positivos (TP): cantidad de veces que el clasificador diagnosticó el episodio que se buscaba correctamente.
- Verdaderos negativos (TN): cantidad de veces que el clasificador identificó que no existía el episodio para el cual fue entrenado.
- Falsos positivos (FP): cantidad de veces que el clasificador diagnosticó un evento cuando este no pertenecía a la clase entrenada.
- Falsos negativos (FN): cantidad de veces que el clasificador no identificó la naturaleza del episodio.

De esta información básica se desprenden múltiples indicadores útiles para determinar el rendimiento del clasificador, las que utilizaremos están definidas como:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

ECUACIÓN 4: EXACTITUD (ACCURACY)

$$TPR = \frac{TP}{TP + FN}$$

ECUACIÓN 5: SENSIBILIDAD (SENSIBILITY, TRUE POSITIVE RATE)

$$TNR = \frac{TN}{TN + FP}$$

ECUACIÓN 6: ESPECIFICIDAD (SPECIFICITY, TRUE NEGATIVE RATE)

$$FNR = \frac{FN}{TP + FN}$$

ECUACIÓN 7: TASA DE FALLOS (MISS RATE, FALSE NEGATIVE RATE)

$$FPR = \frac{FP}{TN + FP}$$

ECUACIÓN 8: PROBABILIDAD DE FALSA ALARMA (FALL-OUT, FALSE POSITIVE RATE)

$$PPV = \frac{TP}{TP + FP}$$

ECUACIÓN 9: PRECISIÓN (PRECISION, POSITIVE PREDICTIVE VALUE)

$$NPV = \frac{TN}{TN + FN}$$

ECUACIÓN 10: VALOR PREDICTIVO NEGATIVO (NEGATIVE PREDICTIVE VALUE)

$$LR_+ = \frac{TPR}{FPR}$$

ECUACIÓN 11: COCIENTE DE PROBABILIDAD POSITIVO (POSITIVE LIKELIHOOD RATIO)

$$LR_- = \frac{FNR}{TNR}$$

ECUACIÓN 12: COCIENTE DE PROBABILIDAD NEGATIVO (NEGATIVE LIKELIHOOD RATIO)

$$DOR = \frac{LR_+}{LR_-} = \frac{\frac{TP}{FP}}{\frac{FN}{TN}}$$

ECUACIÓN 13: COCIENTE DE POSIBILIDADES DE DIAGNOSTICO (DIAGNOSTIC ODDS RATIO)

Para este estudio es particularmente interesante el cociente de posibilidades de diagnóstico (DOR), que relaciona directamente que tan fuerte es la asociación entre el diagnóstico del clasificador y la presencia de la enfermedad [48].

El DOR puede leerse como la ratio entre la posibilidad de tener resultado positivo cuando se presenta la enfermedad y la posibilidad de tener resultado positivo cuando no hay enfermedad. Esto se representa en la Ecuación 13. El valor del DOR va desde cero hasta infinito. La Tabla 2 presenta la interpretación del valor.

[0,1)	1	(1, ∞)
Mala interpretación de los resultados	No discrimina	Discrimina correctamente

TABLA 2: INTERPRETACIÓN DEL DOR SEGÚN INTERVALOS

Entre más alto el DOR mejor discriminación logra el clasificador entre quienes presentan la condición y quienes no la presentan [48]. O sea, DOR representa cuantas veces más posibilidad hay de detectar enfermedad en quien la padece, que encontrarla en alguien que no la padece. El DOR presenta un problema, puede quedar indefinido si alguno de los valores de la matriz de confusión es cero. Añadir 0,5 a cada uno de estos valores es un método común para calcular una aproximación del DOR [48].

Paso 11 - Resultados de las Pruebas

Luego de clasificar los datos de prueba se tabulan las matrices de confusión obtenidas y se calculan los indicadores. Después se rescatan aquellas pruebas que presentan las siguientes condiciones:

- DOR mayor o igual a 2: Esto nos indica que hay una buena probabilidad de que el test identifique un episodio en el examen realizado a una persona que presenta la enfermedad.
- Valor de TP o TN distinto a cero: Esto nos permite identificar los clasificadores que lograron al menos detectar, de manera correcta, una instancia en donde se presenta un episodio y alguna instancia donde no se presenta.

Paso 12 - Diseño de Clasificador y Validación.

Se propone un sistema tal que se considere el resultado de los tres clasificadores con mejor DOR para decidir finalmente la clase a la que pertenecen los episodios. Este sistema considerara, en una primera etapa, si todos los clasificadores coinciden en el tipo de evento que se presenta. En aquel caso se tiene una clasificación exitosa, de lo contrario, se vota por el mejor candidato. Esta votación se guía por la Ecuación 14:

$$Voto_{clasificación} = \operatorname{sgn} \left(\begin{array}{l} \text{Clasificación}_1 \begin{cases} TPR_1 - FPR_1 \rightarrow \text{Clasificación}_1 = 1 \\ TNR_1 - FNR_1 \rightarrow \text{Clasificación}_1 = 0 \end{cases} + \dots \\ \dots + \text{Clasificación}_2 \begin{cases} TPR_2 - FPR_2 \rightarrow \text{Clasificación}_2 = 1 \\ TNR_2 - FNR_2 \rightarrow \text{Clasificación}_2 = 0 \end{cases} + \dots \\ \dots + \text{Clasificación}_3 \begin{cases} TPR_3 - FPR_3 \rightarrow \text{Clasificación}_3 = 1 \\ TNR_3 - FNR_3 \rightarrow \text{Clasificación}_3 = 0 \end{cases} \end{array} \right)$$

ECUACIÓN 14: FORMULA DE VOTACIÓN.

En donde $Clasificación_{xx}$ tomará el valor de 1 si es que se detecta que el episodio fue detectado como presente y -1 si es que no fue detectado. TPR está definido en Ecuación 5, FPR en Ecuación 8, TNR por Ecuación 6 y FNR en Ecuación 7.

Capítulo 4 – Resultados y Discusión

Los resultados presentados a continuación corresponden a aquellos obtenidos al hacer el experimento con solamente los datos que representan los eventos. Se destacan en naranja aquellas pruebas que presentan las condiciones definidas en Paso 11 - Resultados de las Pruebas

Resultados experimento 1 – Eventos

El experimento 1 se lleva a cabo sobre una base de datos que contiene eventos clasificados ya sea por software o un especialista previamente. No existe una base de datos que contenga sujetos sanos o datos de fracciones de datos que representen un estado normal de ventilación pulmonar. Esta base de datos se construye con la totalidad de los eventos rescatados según lo expuesto en la sección Implementación de la Técnica Propuesta y el análisis posterior se realiza sobre todos los pacientes a la vez, ya que no se diferencia si el episodio proviene de algún individuo particular.

Se observa en los resultados tabulados (Tabla 3 - Tabla 14) que los clasificadores FR, DT y NB consistentemente obtienen las condiciones propuestas como deseables; DOR mayor que 2 y numero de TP o TN distinto a cero. Esto sumado a tener exactitud cercana o superior al 70% lo que significa que son confiables y pueden ser utilizados para el diseño posterior. Cabe destacar que luego de aplicar reducción de variables, estos resultados se mantienen en más o menos el mismo nivel de exactitud. En naranja se destacan aquellos clasificadores que presentan las condiciones definidas en Paso 11 - Resultados de las Pruebas.

Resultados Línea Base

Hipopnea	TP	FP	TN	FN	Total	Exactitud ²	DOR ³
FR	647	105	1.009	154	1.915	81,26%	40,3725
DT	688	126	1.062	162	2.038	85,87%	35,7954
SVM	0	0	1.188	850	2.038	58,29%	1,3974
MLP	0	0	1.188	850	2.038	58,29%	1,3974
PNN	384	332	856	466	2.038	60,84%	2,1246
NB	383	205	983	467	2.038	67,03%	3,9326

TABLA 3: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, HIPOPNEA.

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	290	49	482	40	861	37,88%	71,3163
DT	708	238	867	225	2.038	77,28%	11,4629
SVM	0	0	1.105	933	2.038	54,22%	1,1843
MLP	840	970	135	93	2.038	47,84%	1,2571
PNN	470	348	757	463	2.038	60,21%	2,2082
NB	302	95	1.010	631	2.038	64,38%	5,0883

TABLA 4: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	6	0	1.828	42	1.876	89,99%	559,3059
DT	44	49	1.882	63	2.038	94,50%	26,8247
SVM	0	0	1.931	107	2.038	94,75%	17,9674
MLP	0	0	1.931	107	2.038	94,75%	17,9674
PNN	0	0	1.931	107	2.038	94,75%	17,9674
NB	82	531	1.400	25	2.038	72,72%	8,6478

² Ecuación 4

³ Ecuación 13

TABLA 5: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	1	1	1.657	89	1.748	81,35%	18,6180
DT	41	94	1.796	107	2.038	90,14%	7,3211
SVM	0	0	1.890	148	2.038	92,74%	12,7306
MLP	72	934	956	76	2.038	50,44%	0,9697
PNN	0	0	1.890	148	2.038	92,74%	12,7306
NB	8	45	1.845	140	2.038	90,92%	2,3429

TABLA 6: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA MIXTA DEL SUEÑO.

Resultados con Reducción por Random Forest

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	563	80	881	144	1.668	70,85%	43,0558
DT	701	138	1.050	149	2.038	85,92%	35,7966
SVM	0	0	1.188	850	2.038	58,29%	1,3974
MLP	0	0	1.188	850	2.038	58,29%	1,3974
PNN	414	386	802	436	2.038	59,67%	1,9729
NB	591	439	749	259	2.038	65,75%	3,8932

TABLA 7: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, HIPOPNEA

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	471	98	764	178	1.511	60,60%	20,6285
DT	721	243	862	211	2.037	77,67%	12,1214
SVM	0	0	1.105	933	2.038	54,22%	1,1843
MLP	0	0	1.105	933	2.038	54,22%	1,1843
PNN	488	390	715	445	2.038	59,03%	2,0105
NB	544	329	776	389	2.038	64,77%	3,2985

TABLA 8: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	16	13	1.853	49	1.931	91,71%	46,5432
DT	39	44	1.887	68	2.038	94,50%	24,5966
SVM	0	0	1.931	107	2.038	94,75%	17,9674
MLP	87	1273	658	20	2.038	36,56%	2,2485
PNN	0	0	1.931	107	2.038	94,75%	17,9674
NB	63	307	1.624	44	2.038	82,78%	7,5742

TABLA 9: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	4	3	1.750	116	1.873	86,06%	20,1149
DT	29	89	1.801	119	2.038	89,79%	4,9315
SVM	0	0	1.890	148	2.038	92,74%	12,7306
MLP	0	10	1.880	148	2.038	92,25%	0,6030
PNN	0	0	1.890	148	2.038	92,74%	12,7306
NB	33	216	1.674	115	2.038	83,76%	2,2239

TABLA 10: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA MIXTA DEL SUEÑO.

Resultados con Reducción por Tree Ensemble

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	547	84	857	148	1.636	68,89%	37,7074
DT	705	151	1.037	145	2.038	85,48%	33,3905
SVM	0	0	1.188	850	2.038	58,29%	1,3974
MLP	0	0	1.188	850	2.038	58,29%	1,3974
PNN	414	386	802	436	2.038	59,67%	1,9729
NB	581	426	762	269	2.038	65,90%	3,8634

TABLA 11: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, HIPOPNEA.

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	462	95	757	174	1.488	59,81%	21,1575
DT	704	219	886	229	2.038	78,02%	12,4373
SVM	0	0	1.105	933	2.038	54,22%	1,1843
MLP	933	1.105	0	0	2.038	45,78%	0,8444
PNN	488	390	715	445	2.038	59,03%	2,0105
NB	583	356	749	350	2.038	65,36%	3,5046

TABLA 12: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	11	12	1.854	53	1.930	91,51%	32,0660
DT	38	48	1.883	69	2.038	94,26%	21,6045
SVM	0	0	1.931	107	2.038	94,75%	17,9674
MLP	3	119	1.812	104	2.038	89,06%	0,4392
PNN	0	0	1.931	107	2.038	94,75%	17,9674
NB	62	307	1.624	45	2.038	82,73%	7,2883

TABLA 13: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	5	6	1.754	112	1.877	86,31%	13,0506
DT	33	93	1.797	115	2.038	89,79%	5,5447
SVM	0	0	1.890	148	2.038	92,74%	12,7306
MLP	6	31	1.859	142	2.038	91,51%	2,5338
PNN	0	0	1.890	148	2.038	92,74%	12,7306
NB	33	182	1.708	115	2.038	85,43%	2,6930

TABLA 14: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA CENTRAL DEL SUEÑO.

Tabla Comparación Reducción de Variables

Como muestra la siguiente tabla, se logra reducir el número de variables a valores cercanos al 10%, y sumado a lo observado anteriormente, la exactitud de los clasificadores se mantiene en niveles de confianza, cercanos o superiores al 70%.

	Línea Base (N° Variables)	Random Forest (N° Variables)		Tree Ensemble (N° Variables)	
Hipopnea	771	71	9,21%	70	9,08%
OSA	771	79	10,25%	77	9,99%
CSA	771	72	9,34%	72	9,34%
MSA	771	67	8,69%	71	9,21%

TABLA 15: RESULTADOS REDUCCIÓN DE VARIABLES, LOS PORCENTAJES REPRESENTAN LA FRACCIÓN DE VARIABLES SELECCIONADAS RESPECTO A LA LÍNEA BASE.

Ecuación Clasificador Final

Con los resultados obtenidos y observados en las tablas, los clasificadores FR, DT y NB consistentemente cumplen las condiciones fijadas anteriormente

$$\begin{aligned}
 & Voto_{clasificación} \\
 & = \operatorname{sgn} \left(\begin{aligned}
 & \text{Clasificación}_{FR} \times \begin{cases} TPR_{FR} - FPR_{FR} \rightarrow \text{Clasificación}_{FR} = 1 \\ TNR_{FR} - FNR_{FR} \rightarrow \text{Clasificación}_{FR} = 0 \end{cases} + \dots \\
 & \dots + \text{Clasificación}_{DT} \times \begin{cases} TPR_{DT} - FPR_{DT} \rightarrow \text{Clasificación}_{DT} = 1 \\ TNR_{DT} - FNR_{DT} \rightarrow \text{Clasificación}_{DT} = 0 \end{cases} + \dots \\
 & \dots + \text{Clasificación}_{NB} \times \begin{cases} TPR_{NB} - FPR_{NB} \rightarrow \text{Clasificación}_{NB} = 1 \\ TNR_{NB} - FNR_{NB} \rightarrow \text{Clasificación}_{NB} = 0 \end{cases}
 \end{aligned} \right)
 \end{aligned}$$

ECUACIÓN 15: FORMULA DE VOTACIÓN.

$Clasificación_{FR}$ corresponde al resultado de clasificar con el clasificador de FR, $Clasificación_{DT}$ al clasificador DT y, $Clasificación_{NB}$ al NB. TPR está definido en Ecuación 5, FPR en Ecuación 8, TNR por Ecuación 6 y FNR en Ecuación 7.

Matrices de Confusión

	Hipopnea	OSA	CSA	MSA	Sin clase	Total casos
Hipopnea	660	97	9	16	0	782
OSA	33	466	21	46	0	566
CSA	3	6	21	4	0	34
MSA	5	24	10	19	0	58
No clasificados	191	297	46	64	0	598
Total predicción	892	890	107	149	0	2.038

TABLA 16: RESULTADOS PRIMERA ETAPA DEL CLASIFICADOR DISEÑADO

En la Tabla 16 se puede observar en color verde los verdaderos positivos para cada una de las clases de evento, con un 29,34% de los episodios sin clasificar. Se aplica un segundo clasificador por votación que considera clasificadores para Hipopnea, OSA y CSA, ya que ninguno de los clasificadores para MSA cumple el requisito impuesto para tener confiabilidad.

Además, se estudian por separado los clasificadores que presentan mayor DOR para los distintos episodios, DT y NB, sobre los 598 episodios que no lograron ser identificados. Se eligen sobre estos clasificadores aquellos que entregan un mayor índice TP. Por lo tanto, se determina trabajar con DT para hipopnea, OSA y MSA, y con NB para CSA.

Ya con las clasificaciones realizadas se diseña el clasificador como una cascada. Se utiliza primero aquel que logra mejor DOR, según se observa en Tabla 17. Esto significa que primero se aplicara el test de hipopnea, segundo el de MSA, en tercera instancia el de CSA y finalmente OSA. En la Tabla 18 se puede observar la matriz de confusión para los 598 episodios sin clasificar y en la Tabla 19 el resultado agregado luego de ambas etapas.

	TP	FP	TN	FN	Total	Exactitud	DOR
DT (hip)	108	82	367	40	597	0,2330	12,0841
NB (hip)	156	38	251	153	598	0,1997	6,7347
DT (OSA)	181	116	213	88	598	0,1933	3,7767
NB (OSA)	10	287	275	26	598	0,1398	0,3685
DT (CSA)	5	41	536	16	598	0,2654	4,0853
NB (CSA)	9	37	501	51	598	0,2502	2,3895
DT (MSA)	0	64	534	0	598	0,2620	8,2868
NB (MSA)	0	64	532	2	598	0,2610	0

TABLA 17: RESULTADOS CLASIFICADORES SOBRE EVENTOS SIN CLASIFICACIÓN LUEGO DE LA PRIMERA ETAPA.

	Hipopnea	OSA	CSA	MSA	Sin clase	Total casos
Hipopnea	108	31	5	4	0	108
OSA	19	158	17	29	0	223
CSA	9	26	9	4	0	48
MSA	0	0	0	0	0	0
No Clasificados	55	82	15	27	0	179
Total predicción	191	297	64	46	0	598

TABLA 18: RESULTADOS SEGUNDA ETAPA DEL CLASIFICADOR DISEÑADO

	Hipopnea	OSA	CSA	MSA	Sin Clase	Total casos
Hipopnea	768	128	21	13	0	930
OSA	52	624	63	50	0	789
CSA	14	50	28	14	0	106
MSA	3	6	4	21	0	34
No clasificados	55	82	27	15	0	179
Total predicción	892	890	107	149	0	2.038

TABLA 19: RESULTADOS DEL CLASIFICADOR DISEÑADO (SUMA TABLA 17 Y TABLA 18)

Así, con 179 eventos sin clasificar, o un 8,78% del total de eventos, se da por concluido el diseño del clasificador. Podemos observar que, de los 2.038 eventos, 1.441 (70,70%) fueron clasificados correctamente, 418 (20,51%) fueron asignados como otro tipo de evento (error de tipo I) y 179 (8,78%) como negativos a alguna de las condiciones (error de tipo II).

Resultados Experimento 2 – Eventos y Data “sin eventos”

Este experimento 2 trabaja sobre la misma base de datos del primero, pero agrega una base de datos construida con información tomada de los mismos pacientes, en espacios de tiempos que ni el médico, ni el software detectaron eventos propios del SAHOS. Cabe destacar que los datos considerados como sin eventos son recopilados de personas enfermas de SAHOS, o que presentan algún tipo de apnea durante el sueño.

En los resultados no se observan prácticamente clasificadores que logren cumplir los criterios propuestos. Pese a esto, se observa un alto porcentaje de exactitud, sobre el 70% para la mayoría de los casos. En naranja se destacan aquellos clasificadores que presentan las condiciones definidas en Paso 11 - Resultados de las Pruebas.

Resultados Línea Base

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	92	282	2.113	713	3.200	62,66%	0,9668
DT	247	694	1.932	642	3.515	61,92%	1,0710
SVM	0	0	2.629	890	3.519	74,71%	2,9528
MLP	250	675	1.954	640	3.519	62,63%	1,1308
PNN	66	191	2.438	824	3.519	71,16%	1,0224
NB	286	849	1.780	604	3.519	58,71%	0,9928

TABLA 20: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, HIPOPNEA.

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	21	49	621	241	932	18,24%	1,1043
DT	206	602	2.010	701	3.519	62,97%	0,9812
SVM	0	0	2.612	907	3.519	74,23%	2,8788
MLP	707	2.025	587	200	3.519	36,77%	1,0247
PNN	72	183	2.429	835	3.519	71,07%	1,1445
NB	40	109	2.503	867	3.519	72,26%	1,0594

TABLA 21: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	0	3.036	96	3.132	86,27%	31,4663
DT	3	101	3.306	109	3.519	94,03%	0,9009
SVM	0	0	3.407	112	3.519	96,82%	30,2889
MLP	0	0	3.407	112	3.519	96,82%	30,2889
PNN	0	0	3.407	112	3.519	96,82%	30,2889
NB	19	523	2.884	93	3.519	82,50%	1,1266

TABLA 22: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	2	2.852	121	2.975	81,05%	4,6954
DT	3	103	3.278	135	3.519	93,24%	0,7072
SVM	0	0	3.381	138	3.519	96,08%	24,4152
MLP	0	0	3.381	138	3.519	96,08%	24,4152
PNN	0	0	3.381	138	3.519	96,08%	24,4152
NB	11	135	3.246	127	3.519	92,55%	2,0826

TABLA 23: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA MIXTA DEL SUEÑO.

Sin Evento	TP	FP	TN	FN	Total	Exactitud	DOR
FR	439	589	1.243	869	3.140	47,80%	1,0661
DT	539	792	1.254	929	3.514	50,95%	0,9186
SVM	0	0	2.047	1.472	3.519	58,17%	1,3905
MLP	74	125	1.922	1.398	3.519	56,72%	0,8139
PNN	526	754	1.293	946	3.519	51,69%	0,9535
NB	93	158	1.889	1.379	3.519	56,32%	0,8063

TABLA 24: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, SIN EVENTO.

Resultados con Reducción por Random Forest

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	37	149	1.894	612	2.692	54,87%	0,7685
DT	240	684	1.945	650	3.519	62,09%	1,0499
SVM	0	0	2.629	890	3.519	74,71%	2,9528
MLP	881	2.611	18	9	3.519	25,55%	0,6748
PNN	108	347	2.282	782	3.519	67,92%	0,9082
NB	587	1.677	952	303	3.519	43,73%	1,0998

TABLA 25: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, HIPOPNEA

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	115	323	1.461	516	2.415	44,79%	1,0081
DT	257	721	1.891	650	3.519	61,04%	1,0370
SVM	0	0	2.612	907	3.519	74,23%	2,8788
MLP	69	203	2.409	838	3.519	70,42%	0,9771
PNN	72	213	2.399	835	3.519	70,22%	0,9712
NB	245	683	1.929	662	3.519	61,78%	1,0452

TABLA 26: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	1	10	3.224	113	3.348	91,65%	2,8531
DT	7	113	3.268	131	3.519	93,07%	1,5454
SVM	0	0	3.381	138	3.519	96,08%	24,4152
MLP	35	920	2.487	77	3.519	71,67%	1,2288
PNN	0	1	3.380	138	3.519	96,05%	8,0662
NB	35	1.044	2.337	103	3.519	67,41%	0,7607

TABLA 27: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	2	2.852	121	2.975	81,05%	4,8979
DT	3	103	3.278	135	3.519	93,24%	0,7072
SVM	0	0	3.381	138	3.519	96,08%	24,4152
MLP	1	15	3.366	137	3.519	95,68%	1,6380
PNN	0	0	3.381	138	3.519	96,08%	24,4152
NB	11	135	3.246	127	3.519	92,55%	2,0826

TABLA 28: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA MIXTA DEL SUEÑO.

Sin Evento	TP	FP	TN	FN	Total	Exactitud	DOR
FR	314	405	1.057	738	2.514	38,96%	1,1104
DT	640	857	1.190	832	3.519	52,00%	1,0681
SVM	0	0	2.047	1.472	3.519	58,17%	1,3905
MLP	1.472	2.047	0	0	3.519	41,83%	0,7192
PNN	518	713	1.334	954	3.519	52,63%	1,0159
NB	522	403	1.069	1.525	3.519	45,21%	0,9080

TABLA 29: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, SIN EVENTO.

Resultados con Reducción por Tree Ensemble

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	44	121	1.865	610	2.640	54,25%	1,1118
DT	237	704	1.925	653	3.519	61,44%	0,9924
SVM	0	0	2.629	890	3.519	74,71%	2,9528
MLP	771	2.297	332	119	3.519	31,34%	0,9364
PNN	31	90	2.539	859	3.519	73,03%	1,0181
NB	530	1.484	1.145	360	3.519	47,60%	1,1359

TABLA 30: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, HIPOPNEA.

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	99	273	1.406	491	2.269	42,77%	1,0384
DT	254	734	1.878	653	3.519	60,59%	0,9952
SVM	0	0	2.612	907	3.519	74,23%	2,8788
MLP	153	409	2.203	754	3.519	66,95%	1,0930
PNN	50	138	2.474	857	3.519	71,72%	1,0459
NB	251	741	1.871	656	3.519	60,30%	0,9661

TABLA 31: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	1	3.300	108	3.409	93,78%	10,1397
DT	1	74	3.333	111	3.519	94,74%	0,4058
SVM	0	0	3.407	112	3.519	96,82%	30,2889
MLP	0	4	3.403	112	3.519	96,70%	3,3614
PNN	0	0	3.407	112	3.519	96,82%	30,2889
NB	41	1.307	2.100	71	3.519	60,84%	0,9278

TABLA 32: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	4	3.279	134	3.417	93,18%	2,7092
DT	3	134	3.247	135	3.519	92,36%	0,5385
SVM	0	0	3.381	138	3.519	96,08%	24,4152
MLP	0	5	3.376	138	3.519	95,94%	2,2163
PNN	0	1	3.380	138	3.519	96,05%	8,1360
NB	43	1.077	2.304	95	3.519	66,70%	0,9683

TABLA 33: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, APNEA CENTRAL DEL SUEÑO.

Sin Evento	TP	FP	TN	FN	Total	Exactitud	DOR
FR	319	392	1.012	728	2.451	37,82%	1,1312
DT	592	906	1.141	880	3.519	49,25%	0,8472
SVM	0	0	2.047	1472	3.519	58,17%	1,3905
MLP	644	864	1.183	828	3.519	51,92%	1,0649
PNN	559	807	1.240	913	3.519	51,12%	0,9408
NB	933	1.345	702	539	3.519	46,46%	0,9035

TABLA 34: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, SIN EVENTO.

Tabla Comparación Reducción de Variables

Se logra una reducción de variables considerable, hasta llegar a resultados de una fracción menor al 9% de los datos. Se mantienen a groso modo los índices DOA y exactitud, mas, los clasificadores no son utilizables para los fines de este trabajo.

	Línea Base	Random Forest		Tree Ensemble	
Hipopnea	702	45	6,41%	40	5,70%
OSA	702	48	6,84%	52	7,41%
CSA	702	59	8,40%	63	8,97%
MSA	702	45	6,41%	44	6,27%
Sin Evento	702	34	4,84%	35	4,99%

TABLA 35: RESULTADOS REDUCCIÓN DE VARIABLES.

Ecuación Clasificador Final

Debido a que no se obtuvieron resultados satisfactorios para los criterios definidos, no se propone una combinación de clasificadores.

Matrices de Confusión

Ya que no se propone un clasificador, no se obtiene matriz de confusión para este experimento.

Resultados Experimento 3 – Eventos y Data “sin eventos”, proporcional

Este experimento utiliza la misma base de datos combinada del experimento 2. Esta base de datos es manipulada de tal manera que los datos del evento con mayor representación y de aquellos con la etiqueta de “sin evento” tengan una cantidad similar de representantes. La idea tras este experimento es descartar que el experimento 2 sufriera de un sobreajuste debido a la proporción de los eventos (presentada en la Figura 7).

En los resultados se observa que los clasificadores no lograron los criterios mínimos propuestos para considerarlos en el diseño de un clasificador compuesto. En naranja se destacan aquellos clasificadores que presentan las condiciones definidas en Paso 11 - Resultados de las Pruebas.

Resultados Línea Base

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	141	324	1.615	604	2.684	59,36%	1,1636
DT	259	647	1.480	571	2.957	58,79%	1,0376
SVM	0	0	2.128	830	2.958	71,94%	2,5629
MLP	139	376	1.752	691	2.958	63,93%	0,9373
PNN	134	344	1.784	696	2.958	64,84%	0,9985
NB	107	270	1.858	723	2.958	66,43%	1,0184

TABLA 36: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, HIPOPNEA.

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	21	69	361	184	635	12,91%	0,5971
DT	289	612	1.408	648	2.957	57,37%	1,0261
SVM	0	0	2.021	937	2.958	68,32%	2,1563
MLP	1	1	2.020	936	2.958	68,32%	2,1581
PNN	124	288	1.733	813	2.958	62,78%	0,9178
NB	266	562	1.459	671	2.958	58,32%	1,0291

TABLA 37: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	0	2.452	92	2.544	82,89%	26,5135
DT	5	138	2.713	102	2.958	91,89%	0,9637
SVM	0	0	2.851	107	2.958	96,38%	26,5256
MLP	0	0	2.851	107	2.958	96,38%	26,5256
PNN	0	0	2.851	107	2.958	96,38%	26,5256
NB	4	133	2.718	103	2.958	92,02%	0,7936

TABLA 38: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	1	2.250	119	2.370	76,06%	6,2775
DT	2	106	2.703	147	2.958	91,45%	0,3469
SVM	0	0	2.809	149	2.958	94,96%	18,7926
MLP	1	101	2.708	148	2.958	91,58%	0,1812
PNN	0	0	2.809	149	2.958	94,96%	18,7926
NB	8	96	2.713	141	2.958	91,99%	1,6034

TABLA 39: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, APNEA MIXTA DEL SUEÑO.

No Evento	TP	FP	TN	FN	Total	Exactitud	DOR
FR	172	341	1.477	699	2.689	55,75%	1,0658
DT	302	618	1.405	631	2.956	57,71%	1,0881
SVM	0	0	2.023	935	2.958	68,39%	2,1630
MLP	197	421	1.602	738	2.958	60,82%	1,0158
PNN	142	313	1.710	793	2.958	62,61%	0,9783
NB	241	478	1.545	694	2.958	60,38%	1,1224

TABLA 40: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE, SIN EVENTO.

Resultados con Reducción por Random Forest

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	68	187	1.339	523	2.117	47,57%	0,9310
DT	258	672	1.456	572	2.958	57,94%	0,9773
SVM	0	0	2.128	830	2.958	71,94%	2,5629
MLP	295	687	1.441	535	2.958	58,69%	1,1566
PNN	105	254	1.874	725	2.958	66,90%	1,0685
NB	640	1.666	462	190	2.958	37,25%	0,9341

TABLA 41: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, HIPOPNEA

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	55	113	1.125	516	1.809	39,89%	1,0612
DT	293	666	1.355	644	2.958	55,71%	0,9257
SVM	0	0	2.021	937	2.958	68,32%	2,1563
MLP	0	0	937	2.021	2.958	31,68%	0,4638
PNN	193	447	1.574	744	2.958	59,74%	0,9134
NB	600	1.267	754	337	2.958	45,77%	1,0595

TABLA 42: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	1	8	2.702	96	2.807	91,38%	3,5182
DT	4	119	2.732	103	2.958	92,49%	0,8916
SVM	0	0	2.851	107	2.958	96,38%	26,5256
MLP	16	657	2.194	91	2.958	74,71%	0,5872
PNN	0	0	2.851	107	2.958	96,38%	26,5256
NB	48	1.182	1.669	59	2.958	58,05%	1,1488

TABLA 43: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	3	2.659	141	2.803	89,89%	2,6850
DT	5	95	2.714	144	2.958	91,92%	0,9920
SVM	0	0	2.809	149	2.958	94,96%	18,7926
MLP	8	178	2.631	141	2.958	89,22%	0,8386
PNN	0	0	2.809	149	2.958	94,96%	18,7926
NB	47	762	2.047	102	2.958	70,79%	1,2378

TABLA 44: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, APNEA MIXTA DEL SUEÑO.

No Evento	TP	FP	TN	FN	Total	Exactitud	DOR
FR	63	128	1.115	522	1.828	39,82%	1,0513
DT	323	684	1.339	612	2.958	56,19%	1,0332
SVM	0	0	2.023	935	2.958	68,39%	2,1630
MLP	853	1.875	148	82	2.958	33,84%	0,8211
PNN	138	275	1.748	797	2.958	63,76%	1,1006
NB	632	1.368	655	303	2.958	43,51%	0,9987

TABLA 45: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST, SIN EVENTO.

Resultados con Reducción por Tree Ensemble

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	74	186	1.258	485	2.003	45,03%	1,0319
DT	242	647	1.481	588	2.958	58,25%	0,9421
SVM	0	0	2.128	830	2.958	71,94%	2,5629
MLP	394	1.002	1.126	436	2.958	51,39%	1,0155
PNN	62	152	1.976	768	2.958	68,90%	1,0495
NB	661	1.666	462	169	2.958	37,96%	1,0846

TABLA 46: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, HIPOPNEA.

OSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	77	155	1.108	522	1.862	40,06%	1,0545
DT	284	653	1.368	653	2.958	55,85%	0,9111
SVM	0	0	2.021	937	2.958	68,32%	2,1563
MLP	0	2	2.019	937	2.958	68,26%	0,4308
PNN	194	449	1.572	743	2.958	59,70%	0,9142
NB	682	1.447	574	255	2.958	42,46%	1,0609

TABLA 47: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, APNEA OBSTRUCTIVA DEL SUEÑO.

CSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	2	11	2.724	98	2.835	92,16%	5,0538
DT	4	110	2.741	103	2.958	92,80%	0,9677
SVM	0	0	2.851	107	2.958	96,38%	26,5256
MLP	0	4	2.847	107	2.958	96,25%	2,9431
PNN	0	0	2.851	107	2.958	96,38%	26,5256
NB	49	1.198	1.653	58	2.958	57,54%	1,1657

TABLA 48: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	0	4	2.654	139	2.797	89,72%	2,1143
DT	7	95	2.714	142	2.958	91,99%	1,4083
SVM	0	0	2.809	149	2.958	94,96%	18,7926
MLP	14	321	2.488	135	2.958	84,58%	0,8038
PNN	0	0	2.809	149	2.958	94,96%	18,7926
NB	46	700	2.109	103	2.958	72,85%	1,3455

TABLA 49: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, APNEA CENTRAL DEL SUEÑO.

MSA	TP	FP	TN	FN	Total	Exactitud	DOR
FR	73	154	1.133	535	1.895	40,77%	1,0039
DT	283	621	1.402	652	2.958	56,96%	0,9799
SVM	0	0	2.023	935	2.958	68,39%	2,1630
MLP	743	1.647	376	192	2.958	37,83%	0,8834
PNN	146	301	1.722	789	2.958	63,15%	1,0586
NB	625	1.320	703	310	2.958	44,90%	1,0737

TABLA 50: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE, SIN EVENTO.

Tabla comparación reducción de variables

Se logra una reducción de variables considerable, hasta llegar a obtener resultados de una fracción menor al 7% de los datos. Se mantienen a groso modo los índices DOA y exactitud, mas, los clasificadores no son utilizables para los fines de este trabajo.

	Línea Base	Random Forest		Tree Ensemble	
Hipopnea	702	47	6,7%	54	7,7%
OSA	702	48	6,8%	48	6,8%
CSA	702	57	8,1%	50	7,1%
MSA	702	50	7,1%	52	7,4%
Sin Evento	702	46	6,6%	48	6,8%

TABLA 51: RESULTADOS REDUCCIÓN DE VARIABLES.

Ecuación Clasificador Final

Debido a que no se obtuvieron resultados satisfactorios para los criterios definidos, no se propone una combinación de clasificadores.

Matrices de Confusión

Ya que no se propone un clasificador, no se obtiene matriz de confusión para este experimento.

Resultados Reducción de Variables

Para cada algoritmo de reducción de variables se detallan las señales que fueron seleccionadas para cada uno de los tipos de episodios. Se presentan los resultados relacionados al primer experimento.

Random Forest

Hipopnea

- Duration
- Stage
- EEG_C3_A2
- EEG_P3_A2
- EEG_F4_A1
- EEG_P4_A1
- EEG_O1_A2
- EEG_O2_A1
- EEG_Fp1_A2
- EEG_Fp2_A1
- EEG_A1_A2
- EOG_Left
- EMG_Chin
- Flow_Patient
- Flow_Patient_2
- Effort_Tho
- Effort_Abd
- Snore
- ECG_I
- RR
- Leg_PIERNA1
- Leg_PIERNA2
- SpO2
- Imp

OSA

- Duration
- Stage
- EEG_Fp1_A2
- EEG_A1_A2
- EOG_Right
- EOG_Left
- Flow_Patient
- Flow_Patient_2
- Flow_Patient_3
- Effort_Tho
- Effort_Abd
- Snore
- RR
- Leg_PIERNA1
- Leg_PIERNA2
- xPAP_IPAP
- SpO2
- Imp

CSA

- Age
- Stage
- EEG_A1_A2
- EOG_Right
- EOG_Left
- EMG_Chin
- Flow_Patient
- Flow_Patient_2
- Effort_Tho
- Effort_Abd
- Snore
- ECG_I
- RR
- Body
- Leg_PIERNA1
- Leg_PIERNA2
- xPAP_EPAP
- xPAP_IPAP
- Imp

MSA

- Age
- EEG_O1_A2
- EEG_Fp1_A2
- EEG_A1_A2
- EOG_Right
- EOG_Left
- EMG_Chin
- Flow_Patient
- Flow_Patient_2
- Effort_Tho
- Effort_Abd
- Snore
- ECG_I
- RR
- Body
- Leg_PIERNA2
- xPAP_IPAP
- SpO2
- Imp

Tree Ensemble

Hipopnea

- Duration
- Stage
- EEG_C3_A2
- EEG_F4_A1
- EEG_P4_A1
- EEG_O1_A2
- EEG_Fp1_A2
- EEG_Fp2_A1
- EEG_A1_A2
- EOG_Left
- EMG_Chin
- Flow_Patient
- Flow_Patient_2
- Effort_Tho
- Effort_Abd
- Snore
- RR
- Body
- Leg_PIERNA1
- Leg_PIERNA2
- xPAP_EPAP
- SpO2
- Imp

OSA

- Duration
- Stage
- EEG_Fp1_A2
- EEG_A1_A2
- EOG_Left
- Flow_Patient
- Flow_Patient_2
- Flow_Patient_3
- Effort_Tho
- Effort_Abd
- Snore
- RR
- Leg_PIERNA1
- Leg_PIERNA2
- xPAP_IPAP
- SpO2
- Imp

CSA

- Age
- Stage
- EEG_A1_A2
- EOG_Right
- EOG_Left
- EMG_Chin
- Flow_Patient
- Flow_Patient_2
- Effort_Tho
- Effort_Abd
- Snore
- ECG_I
- RR
- Body
- Leg_PIERNA1
- Leg_PIERNA2
- xPAP_EPAP
- xPAP_IPAP
- Imp

MSA

- Age
- Gender
- EEG_O1_A2
- EEG_Fp1_A2
- EEG_A1_A2
- EOG_Right
- EOG_Left
- EMG_Chin
- Flow_Patient
- Flow_Patient_2
- Flow_Patient_3
- Effort_Tho
- Effort_Abd
- Snore
- ECG_I
- RR
- Body
- Leg_PIERNA2
- xPAP_EPAP
- xPAP_IPAP
- SpO2
- Imp

Discusión Resultados

Los resultados presentados por la segunda y tercera prueba no nos permiten realizar una satisfactoria clasificación, esto puede deberse a distintos factores.

El primero de ellos se puede deber a la posibilidad que la muestra tomada como control, al ser derivada de los mismos exámenes de pacientes con al menos una de las patologías del SAHOS, se encuentre de alguna forma contaminada con rasgos propios de alguno de los eventos estudiados.

Esto es coherente con que la primera prueba obtuviera resultados satisfactorios, ya que el algoritmo es capaz de reconocer los rasgos distintivos de cada evento en ese experimento, como se puede observar en la Tabla 19, y se reescribe en la siguiente tabla:

Eventos	TP	FP	TN	FN	Total	Exactitud	DOR
Hipopnea	768	124	984	162	2.038	85,97%	37,6201
OSA	636	254	977	171	2.038	79,15%	14,3061
CSA	19	88	1.894	37	2.038	93,87%	11,0522
MSA	27	122	1.850	39	2.038	92,10%	10,4981

TABLA 52: RESUMEN CLASIFICADOR DISEÑADO

Con un DOR superior a 10 en cada evento, podemos validar lo propuesto en el párrafo anterior, y si se observan los otros índices, presentados en la Tabla 53, se puede ver que, en general, se tiene una alta confiabilidad en los resultados, con TPR mayor al 75% en el caso de hipopnea y apnea obstructiva y TNR mayor al 79% en todos los casos.

Eventos	TPR	TNR	FNR	FPR	PPV	NPV
Hipopnea	82,58%	88,81%	17,42%	11,19%	86,10%	85,86%
OSA	78,81%	79,37%	21,19%	20,63%	71,46%	85,10%
CSA	33,93%	95,56%	66,07%	4,44%	17,76%	98,08%
MSA	40,91%	93,81%	59,09%	6,19%	18,12%	97,94%

TABLA 53: OTROS ÍNDICES, CLASIFICADOR DISEÑADO

Un segundo factor se puede deber al uso tanto episodios clasificados por una máquina, como aquellos validados por un médico indistintamente. Se puede clarificar que tan cierta es esta hipótesis al realizar un cuarto experimento, en el cual se entrenen los clasificadores solamente con episodios que sean validados por un médico. Esto significa que se trabajaría con 2.789 eventos para el entrenamiento, divididos entre las cuatro clases.

Adicionalmente, se realiza un experimento donde se entrenan a los clasificadores usando solamente los episodios validados por un médico más parte de los episodios marcados como “sin evento”. La validación se realiza con una fracción de estos mismos episodios, y se propone un diseño de clasificador a partir de estos resultados para luego probarlo con los datos que encontró Alice Sleepware® y que no valido un médico.

El objetivo de esto es contraponer los eventos clasificados por Alice Sleepware® con los datos del grupo de control o datos marcados como “sin evento”.

Se presenta una tabla que resume los resultados del entrenamiento y validación de este experimento, solo se muestran aquellos resultados que presentan un DOR mayor a 2 y no presentan ceros en los índices TP y TN.

Eventos	Caso	Clasificador	TP	FP	TN	FN	Total	Exactitud	DOR
CSA	Base	DT	4	25	637	32	698	91,83%	3,1850
CSA	Base	NB	3	8	654	33	698	94,13%	7,4318
CSA	RF	MLP	7	51	611	29	698	88,54%	2,8918
MSA	RF	PNN	1	2	640	55	698	91,83%	5,8182
MSA	TE	PNN	1	2	640	55	698	91,83%	5,8182

TABLA 54: RESUMEN DESEMPEÑO ALGORITMOS EN PRUEBA DE ENTRENAMIENTO CON EVENTOS VALIDADOS POR MEDICO

Es necesario notar que se trabajó con una base de datos rescatada de 70 personas, y al solo seleccionar los eventos validados, los clasificadores pueden quedar sub-ajustados. Esto explica que no se logran indicadores satisfactorios para esta prueba.

Otro resultado interesante es observar cómo se relacionan el número de características frente a la exactitud de los clasificadores. Esto es relevante en el contexto de lograr un clasificador que pueda identificar los episodios con la menor cantidad posible de señales involucradas. Se puede resumir esta información para el primer experimento en las siguientes tablas.

Eventos	Exactitud Base	Diferencia Exactitud RF	Ratio de Reducción RF	Diferencia Exactitud TE	Ratio de Reducción TE
FR	81,26%	-10,41%	90,79%	-12,37%	90,92%
DT	85,87%	0,05%		-0,39%	
SVM	58,29%	0,00%		0,00%	
MLP	58,29%	0,00%		0,00%	
PNN	60,84%	-1,17%		-1,17%	
NB	67,03%	-1,28%		-1,13%	

TABLA 55: COMPARACIÓN EXACTITUD CLASIFICADORES HIPOPNEA EN CASO BASE Y POST REDUCCIÓN DE VARIABLES

Eventos	Exactitud Base	Diferencia Exactitud RF	Ratio de Reducción RF	Diferencia Exactitud TE	Ratio de Reducción TE
FR	37,88%	22,72%	89,75%	21,93%	90,01%
DT	77,28%	0,39%		0,74%	
SVM	54,22%	0,00%		0,00%	
MLP	47,84%	6,38%		-2,06%	
PNN	60,21%	-1,18%		-1,18%	
NB	64,38%	0,39%		0,98%	

TABLA 56: COMPARACIÓN EXACTITUD CLASIFICADORES APNEA OBSTRUCTIVA EN CASO BASE Y POST REDUCCIÓN DE VARIABLES

Eventos	Exactitud Base	Diferencia Exactitud RF	Ratio de Reducción RF	Diferencia Exactitud TE	Ratio de Reducción TE
FR	89,99%	1,72%	90,66%	1,52%	90,66%
DT	94,50%	0,00%		-0,24%	
SVM	94,75%	0,00%		0,00%	
MLP	94,75%	-58,19%		-5,69%	
PNN	94,75%	0,00%		0,00%	
NB	72,72%	10,06%		10,01%	

TABLA 57: COMPARACIÓN EXACTITUD CLASIFICADORES APNEA CENTRAL EN CASO BASE Y POST REDUCCIÓN DE VARIABLES

Eventos	Exactitud Base	Diferencia Exactitud RF	Ratio de Reducción RF	Diferencia Exactitud TE	Ratio de Reducción TE
FR	81,35%	4,71%	91,31%	4,96%	90,79%
DT	90,14%	-0,35%		-0,35%	
SVM	92,74%	0,00%		0,00%	
MLP	50,44%	41,81%		41,07%	
PNN	92,74%	0,00%		0,00%	
NB	90,92%	-7,16%		-5,49%	

TABLA 58: COMPARACIÓN EXACTITUD CLASIFICADORES APNEA MIXTA EN CASO BASE Y POST REDUCCIÓN DE VARIABLES

En general se logra una reducción al 10% de las variables con variaciones de entre -12,37% y +22,72% para los casos que se utilizaron en el diseño final del clasificador (FR, DT y NB). Finalmente podemos hacer listas de cuáles son las variables que fueron seleccionadas durante la reducción de estas bajo el subtítulo “

Tabla Comparación Reducción de **Variables**”, cabe mencionar que pese a que la señal “Imp” corresponde a un reloj del sistema aparece como señal relevante en todos los experimentos.

Para la hipopnea y el OSA es relevante la duración del episodio, en CSA y MSA la edad es un aspecto importante. Para todos los episodios son relevantes las señales “Flow_Patient”, “Flow_Patient_2”, y en la mayoría de los casos la “SpO2”, estas tres señales están relacionadas con el flujo de oxígeno y el nivel de oxigenación de la sangre. En OSA, CSA y MSA la señal de “Snore” es seleccionada, esta muestra los ronquidos del paciente. “Effort_Tho” y “Effort_Abd” aparece seleccionada para todos los episodios, esto muestra cómo se mueve la garganta (*Throat* en inglés) y el Abdomen, movimientos fuertemente relacionados con la respiración. Los EOG y “Stage” son seleccionados también en varios casos, estas señales se relacionan al movimiento ocular y a la etapa del sueño respectivamente. Cabe destacar que hay etapas del sueño que se relacionan fuertemente con el movimiento ocular [12].

Capítulo 5 – Conclusiones y Trabajo Futuro

Conclusiones

Se construye satisfactoriamente una base de datos que contiene estadísticos ampliamente utilizados en la literatura, tanto para la señal en el dominio temporal como en el dominio de las frecuencias. Esto se logra gracias a los programas que se escriben para este fin.

Se logra construir un clasificador, a partir de estadísticos, de episodios del SAHOS capaz de identificar exitosamente el 70,70% de los episodios que se alimentan. Existe error de tipo I con el 20,51% de los casos y un error tipo II con el 8,78% de ellos. La exactitud que alcanza el clasificador es superior al 79% para cada uno de los episodios (Tabla 52). Se logra también una reducción de las características de clasificación a un 10%, en casos incluso se alcanza a mejorar la exactitud de los modelos de clasificación (Tabla 55 a Tabla 58). Esto ayuda a corroborar los resultados presentados por KNIME® en [34].

Se observa que las características que han sido seleccionadas repetidamente mediante los diferentes algoritmos de reducción concuerdan en gran medida con lo que se debería observar con detenimiento en un paciente con SOHAS, por ejemplo: Flujo de aire, oxigenación de la sangre, ronquidos, esfuerzos respiratorios y etapas del sueño. Esto indica que la mayor parte de la información relevante se encuentra en estos sensores y debiesen ser los esenciales a la hora de construir un polisomnógrafo.

Producto de esto se reduce desde más de 20 señales a 8, lo cual es suficiente para poder diferenciar entre distintos episodios; EOG derecho e izquierdo, sensor de ronquidos, Esfuerzos abdominal y de garganta, ambos sensores de flujo, derivado de la temperatura y del flujo de la respiración, y un sensor de nivel de oxígeno en la sangre (Resultados Reducción de Variables).

El DOR es un estadístico válido para diseñar clasificadores de alta confianza. En base al criterio propuesto: un DOR mayor a 2, se logra diseñar un clasificador para el experimento 1 que de manera satisfactoria logra identificar los distintos episodios del SAHOS.

Pese a estos resultados positivos, el estudio no es suficiente para asegurar que se ha construido un clasificador que pueda compararse a la herramienta profesional ofrecida por Phillips®. Esto se debe a que no se ha obtenido una base de datos lo suficientemente extensa, ni un grupo de control adecuado para lograr un buen entrenamiento y comparación para los experimentos 2 y 3. La plataforma que se crea en KNIME® ayuda a una fácil repetición de estas experiencias.

Trabajo Futuro

Para continuar este trabajo se recomienda obtener una base de datos más completa, sobre la cual realizar estos mismos experimentos y poder construir de esta manera un clasificador confiable a partir de la presencia de pacientes sanos dentro de la muestra. Estos nuevos datos podrán validar en los experimentos 2 y 3 los resultados obtenidos en el experimento 1.

Se pueden implementar técnicas de *machine learning* como el *k-fold cross-validation* para medir sin sesgos la efectividad de los clasificadores, y a la vez, mejorar la calidad de la base de datos [49]. Resultados preliminares del uso de esta técnica pueden observarse en Anexo 4 – Prueba con *K-Fold Cross-Validation*.

Por otro lado, con la información recopilada por la reducción de variables se puede construir un algoritmo capaz de detectar estos episodios en tiempo real. Al observar el comportamiento segundo a segundo del examen mientras se realiza y al actuar cuando los valores de las señales se sitúen en los valores que un clasificador como RF define como los característicos de un episodio, para así poder actuar de manera oportuna ante los eventos del SAHOS.

Bibliografía

- [1] Ministerio de Salud de Chile, «<http://www.deis.cl/indicadores-basicos-de-salud/>,» 2014. [En línea]. Available: <http://www.deis.cl/indicadores-basicos-de-salud/>.
- [2] D. J. Jorquera, *La apnea del sueño tiene grandes implicancias para la salud*, Santiago: El Mercurio, 2016.
- [3] L. E. L. F. & M. P. Terry Young, «Estimation of the Clinically Diagnosed Proportion of Sleep,» *Sleep*, pp. 705-706, 1997.
- [4] American Academy of Otolaryngology, «Snoring and Sleep Apnea,» 2016. [En línea]. Available: <http://www.entnet.org/content/snoring-and-sleep-apnea>.
- [5] Ş. Y. S. G. Kemal Polat, «Comparison of Different Classifier Algorithms on the Automated Detection of Obstructive Sleep Apnea Syndrome,» *Journal of Medical Systems*, n° 32, pp. 243-250, 2008.
- [6] American Academy of Sleep Medicine, «Glossary,» de *International classification of sleep disorders, revised: Diagnostic and coding manual.*, Chicago, American Academy of Sleep Medicine, 2001, pp. 337-351.
- [7] American Academy of Sleep Medicine, «Obstructive Sleep Apnea Syndrome (780.53-0),» de *International classification of sleep disorders, revised: Diagnostic and coding manual*, Chicago, American Academy of Sleep Medicine, 2001, pp. 52-58.
- [8] American Academy of Sleep Medicine. , «Central Sleep Apnea Syndrome (780.51-0),» de *International classification of sleep disorders, revised: Diagnostic and coding manual*, Chicago, American Academy of Sleep Medicine. , 2001, pp. 58-61.
- [9] D. P. W. W. T. A. C. A. C. D. J. S. F. C. E. H. L. J. O. G. P. R. R. W. T. Y. Virend K. Somers, «Sleep Apnea and Cardiovascular Disease,» *Journal of the American College of Cardiology*, vol. 52, n° 8, pp. 686-717, 2008.
- [10] G. K. G. G. J. S. G. L. J. B. Filippou Triposkiadis, «The Sympathetic Nervous System in Heart Failure: Physiology, Pathophysiology, and Clinical Implications,» *Journal of the American College of Cardiology*, vol. 54, n° 19, pp. 1747 - 1762, 2009.
- [11] Biblioteca Nacional de Medicina de los EE.UU., «Polisomnografía,» 18 Marzo 2014. [En línea]. Available: <https://www.nlm.nih.gov/medlineplus/spanish/ency/article/003932.htm>.
- [12] A. K. A. Rechtschaffen, «A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects,» UCLA Brain Information Service/Brain Research Institute, Los Angeles, 1968.
- [13] S. A. R. Lili Erazoa, «A benchmark on automatic obstructive sleep apnea screening,» *Procedia Computer Science*, vol. 35, p. 739 – 746, 2014.
- [14] E. G, «What is e-health?,» *J Med Internet Res*, vol. 3, n° e20, 2001.
- [15] McGraw-Hill Education, *McGraw-Hill Dictionary of Scientific and Technical Terms*, 6th ed. ed., New York: McGraw-Hill, 2006.

- [16] M. Engin, «ECG beat classification using neuro-fuzzy network,» *Pattern Recognition Letters*, vol. 25, pp. 1715-1722, 2004.
- [17] N. S. A. R. H. ., T. T. S. ., T. A. Fatin A. Elhaj, «Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals,» *Computer Methods and Programs in Biomedicine*, pp. 52-63, 2016.
- [18] R. B. P. Varun Bajaj, «Classification of Seizure and Nonseizure EEG,» *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, pp. 1135-1142, 2012.
- [19] A. Subasi, «Classification of EMG signals using combined features and soft computing techniques,» *Applied Soft Computing*, p. 2188–2198, 2012.
- [20] R. J. G. A. P. FITZ, «FINGERPRINT CLASSIFICATION USING A HEXAGONAL,» *Pattern Recognition*, vol. 29, n° 10, pp. 1587-1597, 1996.
- [21] C.-C. J. K. Tong Zhang, «HIERARCHICAL CLASSIFICATION OF AUDIO DATA FOR ARCHIVING AND RETRIEVING,» de *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, AZ, USA, 1999.
- [22] S. Soliman y S.-Z. Hsue, «Signal classification using statistical moments,» *IEEE Transactions on Communications*, vol. 40, n° 5, pp. 908 - 916, 1992.
- [23] Y. Tian y H. Qi, «TARGET DETECTION AND CLASSIFICATION USING SEISMIC SIGNAL PROCESSING IN UNATTENDED GROUND SENSOR SYSTEMS,» de *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando, FL, USA, 2002.
- [24] J. M. Mendel, «Use of higher-order estadistics in signal processing and system theory; An update,» de *SPIE Conf. on Advanced algorithms and Architectures for Signal Processing III*, San Diego, CA, 1988.
- [25] J. M. MENDEL, «Tutorial on Higher-Order Statistics (Spectra) in Signal Processing and System Theory: Theoretical Results and Some Applications,» de *Proceedings of the IEEE*, 1991.
- [26] Z. Dokur, T. Ölmez y E. Yazgan, «Comparison of discrete wavelet and Fourier transforms for ECG beat classification,» *Electronics Letters*, vol. 35, n° 18, pp. 1502-1504, 1999.
- [27] S. G. Kemal Polat, «Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform,» *Applied Mathematics and Computation*, pp. 1017-1026, 2007.
- [28] ACHS, «Aranceles Hospital del Trabajador, Precios aplicados a prevision FONASA.,» 25 11 2015. [En línea]. Available: <http://www.achs.cl/ht/Comunidad/Aranceles/Consolidado-Aranceles-Fonasa-HTS-25-11-2015-2.pdf>. [Último acceso: 19 06 2017].
- [29] Clinica Tabancura, «Informacion a clientes, Aranceles,» [En línea]. Available: <http://www.clinicatabancura.cl/aranceles/ambito/2/descripcion/hospitalario/prestaciones/6/descripcion/procedimientos-y-examenes>. [Último acceso: 19 06 2017].

- [30] Clinica Alemana, «Arancel Clinica Alemana,» 2017. [En línea]. Available: <http://www.alemana.cl/wls/arancel/buscarExamenes.htm?idClasificacion=3&pagina=1&cantidad=100>. [Último acceso: 19 06 2017].
- [31] B. Kemp, «European Data Format (EDF),» [En línea]. Available: <http://www.edfplus.info/>. [Último acceso: 25 01 2017].
- [32] Q. Fang, «JSONlab: a toolbox to encode/decode JSON files,» 03 01 2017. [En línea]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/33381-jsonlab--a-toolbox-to-encode-decode-json-files>. [Último acceso: 03 2017].
- [33] «Introducing JSON,» [En línea]. Available: www.json.org.
- [34] I. A. A. H. M. B. Rosaria Silipo, «Seven Techniques for Dimensionality Reduction,» Knime, 2014.
- [35] Philips, «Philips Sleepware G3,» [En línea]. Available: <http://www.usa.philips.com/healthcare/product/HC1082462/sleepware-g3-sleep-diagnostic-software>. [Último acceso: 25 01 2017].
- [36] MathWorks, «Matlab Features,» [En línea]. Available: <https://www.mathworks.com/products/matlab/features.html>. [Último acceso: 25 01 2017].
- [37] KNIME.com AG, «KNIME analytics Platform,» [En línea]. Available: <https://www.knime.org/knime-analytics-platform>.
- [38] S. M. Y. L. Min Chen, «Big Data: A Survey,» *Mobile Netw Appl*, n° 19, pp. 171-209, 2014.
- [39] J.-M. P. C. T.-M. Robin Genuer, «Variable selection using random forests,» *Pattern Recognition Letters*, n° 31, p. 2225–2236, 2010.
- [40] S. M. K. Rajendra Lambodari, «Review of Classification Methods for Character Recognition in Neural Networks,» *International Journal of Electronics Communication and Computer Engineering*, vol. 4, n° 2, pp. 24-28, 2013.
- [41] A. B. G. R. K. T. Eugene Tuv, «Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination,» *Journal of Machine Learning Research*, n° 10, pp. 1341-1366, 2009.
- [42] L. A. Z. Nikraves, *Studies in Fuzziness and Soft Computing volume 207, Feature Extraction*, Springer, 2006.
- [43] K.-R. Koch, *Bayesian Inference with Geodetic Applications*, Springer, 2006, pp. 4-8.
- [44] I. Rish, «An empirical study of the naive Bayes classifier,» IBM Research Division, Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich, 2001.
- [45] E. S. Vincent Bombardier, «Fuzzy rule classifier: Capability for generalization in wood color recognition,» *Engineering Applications of Artificial Intelligence*, n° 23, pp. 978-988, 2010.
- [46] S. Haykin, *Neural Networks A Comprehensive Foundation*, PAtparganj, Delhi: Pearson Education, 2005.

- [47] D. F. Specht, «Probabilistic Neural Networks,» *Neural Networks*, , vol. 3, pp. 109-118, 1990.
- [48] J. G. L. M. H. P. G. J. B. P. M. B. Afina S. Glas, «The diagnostic odds ratio: a single indicator of test performance,» *Journal of Clinical Epidemiology*, nº 56, p. 1129–1135, 2003.
- [49] Y. Bengio y Y. Grandvalet, «No Unbiased Estimator of the Variance of K-Fold Cross-Validation,» *Journal of Machine Learning Research* , vol. 5, pp. 1089-1105, 2004.

Anexo 1 – Scripts MATLAB®

blockEdfLoad

```
function varargout = blockEdfLoad(varargin)
% blockEdfLoad Load EDF with memory block reads.
% Function inputs an EDF file text string and returns the header,
% header and each of the signals.
%
% The loader is designed to load the EDF file described in:
%
% Bob Kemp, Alpo Vuori, Agostinho C. Rosa, Kim D. Nielsen and
John Gade
% "A simple format for exchange of digitized polygraphic
recordings"
% Electroencephalography and Clinical Neurophysiology, 82
(1992):
% 391-393.%
%
% An online description of the EDF format can be found at:
% http://www.edfplus.info/
%
% Our EDF tools can be found at:
% http://sleep.partners.org/edf/
%
% Updated versions will be posted:
% http://https://github.com/DennisDean/
%
% Additional Documentation and examples will be posted:
% http://sleepdata.org/tools
%
% Requirements: Self contained, no external references
% MATLAB Version: Requires R14 or newer, Tested with MATLAB
7.14.0.739
%
% Input (VARARGIN):
% edfFN : File text string
```



```

%     signalLabels : Cell array of signal labels to return
%     (optional)
%
% Function Prototypes:
%
%         header = blockEdfLoad(edfFN)
%         header, signalHeader] = blockEdfLoad(edfFN)
%     [header, signalHeader, signalCell] = blockEdfLoad(edfFN)
%     [header, signalHeader, signalCell] = blockEdfLoad(edfFN,
signalLabels)
%     [header, signalHeader, signalCell] = blockEdfLoad(edfFN,
signalLabels, epochs)
%
% Output (VARARGOUT):
%     header : A structure containing variables for each
header entry
%     signalHeader : A structured array containing signal
information,
%
%         for each structure present in the data
%     signalCell : A cell array that contains the data for each
signal
%
% Output Structures:
%     header:
%         edf_ver
%         patient_id
%         local_rec_id
%         recording_startdate
%         recording_starttime
%         num_header_bytes
%         reserve_1
%         num_data_records
%         data_record_duration
%         num_signals
%     signalHeader (structured array with entry for each signal):
%         signal_labels
%         tranducer_type
%         physical_dimension
%         physical_min
%         physical_max

```

```

%     digital_min
%     digital_max
%     prefiltering
%     samples_in_record
%     reserve_2
%
% Examples:
%
% Get EDF header information
%
%     edfFn3 = 'file.edf';
%     header = blockEdfLoad(edfFn3);
%
%     edfFn3 = 'file.edf';
%     [header signalHeader] = blockEdfLoad(edfFn3);
%
%
% Load Signals
%
%     edfFn3 = 'file.edf';
%     [header signalHeader signalCell] = blockEdfLoad(edfFn3);
%
%     edfFn3 = 'file.edf';
%     signalLabels = {'Pleth', 'EKG-R-EKG-L', 'Abdominal Resp'};
%     [header signalHeader signalCell] = blockEdfLoad(edfFn3,
signalLabels);
%
%     epochs = [1 2]; % Load first through second epoch
%     signalLabels = {'Pleth', 'Abdominal Resp', 'EKG-R-EKG-L'};
%     [header signalHeader signalCell] = ...
%         blockEdfLoad(edfFn3, signalLabels, epochs);
%
%
% Version: 0.1.21
%
% -----
% Dennis A. Dean, II, Ph.D
%

```

```
% Program for Sleep and Cardiovascular Medicine
% Brigham and Women's Hospital
% Harvard Medical School
% 221 Longwood Ave
% Boston, MA 02149
%
% File created: October 23, 2012
% Last updated: January 23, 2014
%
% Copyright [2012] The Brigham and Women's Hospital, Inc.
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS
% "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
NOT LIMITED
% TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
% PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT HOLDER
% OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL,
% EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
TO,
% PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
OR
% PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF
% LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING
% NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS
% SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
%
%-----
Process input

% Operation Flags
RETURN_PHYSICAL_VALUES = 1;

% Defaults for optional parameters
```

```

signalLabels = {};           % Labels of signals to return
epochs = [];                 % Start and end epoch to return

% Process input
if nargin == 1
    edfFN = varargin{1};
    signalLabels = {};
elseif nargin == 2 & nargout == 3
    edfFN = varargin{1};
    signalLabels = varargin{2};
elseif nargin == 3 & nargout == 3
    edfFN = varargin{1};
    signalLabels = varargin{2};
    epochs = varargin{3};
else
    % Echo supported function prototypes to console
    fprintf('header = blockEdfLoad(edfFN)\n');
    fprintf('[header, signalHeader] = blockEdfLoad(edfFN)\n');
    fprintf('[header, signalHeader, signalCell] =
blockEdfLoad(edfFN)\n');
    fprintf('[header, signalHeader, signalCell] =
blockEdfLoad(edfFN, signalLabels)\n');
    fprintf('[header, signalHeader, signalCell] =
blockEdfLoad(edfFN, signalLabels, epochs)\n');

    % Call MATLAB error function
    error('Function prototype not valid');
end

%-----
Input check
% Check that first argument is a string
if ~ischar(edfFN)
    msg = ('First argument is not a string. ');
    error(msg);
end
% Check that first argument is a string
if ~iscellstr(signalLabels)

```

```

    msg = ('Second argument is not a valid text string.');
```

error(msg);

end

```

% Check that first argument is a string
if and(nargin ==3, length(epochs)~=2)
    msg = ('Specify epochs = [Start_Epoch End_Epoch.>');
    error(msg);
end
```

end

```

%----- Load File
Information
% Load edf header to memory
[fid, msg] = fopen(edfFN);

% Proceed if file is valid
if fid <0
    % file id is not valid
    error(msg);
end
```

end

```

% Open file for reading
% Load file information not used in this version but will be used
in
% class version
[filename, permission, machineformat, encoding] = fopen(fid);

%-----
Load Header
try
    % Load header information in one call
    edfHeaderSize = 256;
    [A count] = fread(fid, edfHeaderSize);
catch exception
    msg = 'File load error. Check available memory.';
    error(msg);
end
```

```

%----- Process
Header Block
% Create array/cells to create struct with loop
headerVariables = {...
    'edf_ver';          'patient_id';          'local_rec_id';
    ...

'recording_startdate'; 'recording_starttime'; 'num_header_bytes';
...
    'reserve_1';          'num_data_records';
'data_record_duration';...
    'num_signals'};
headerVariablesConF = {...
    @strtrim;    @strtrim;    @strtrim; ...
    @strtrim;    @strtrim;    @str2num; ...
    @strtrim;    @str2num;    @str2num;...
    @str2num};
headerVariableSize = [ 8; 80; 80; 8; 8; 8; 44; 8; 8; 4];
headerVarLoc = vertcat([0],cumsum(headerVariableSize));
headerSize = sum(headerVariableSize);

% Create Header Structure
header = struct();
for h = 1:length(headerVariables)
    conF = headerVariablesConF{h};
    value =
conF(char((A(headerVarLoc(h)+1:headerVarLoc(h+1))))');
    header = setfield(header, headerVariables{h}, value);
end

% End Header Load section

%----- Load
Signal Header
if narginout >= 2
    try
        % Load signal header into memory in one load
        edfSignalHeaderSize = header.num_header_bytes -
headerSize;
        [A count] = fread(fid, edfSignalHeaderSize);

```

```

catch exception
    msg = 'File load error. Check available memory.';
    error(msg);
end

%----- Process Signal
Header Block
% Create arrau/cells to create struct with loop
signalHeaderVar = {...
    'signal_labels'; 'tranducer_type'; 'physical_dimension';
...
    'physical_min'; 'physical_max'; 'digital_min'; ...
    'digital_max'; 'prefiltering'; 'samples_in_record'; ...
    'reserve_2' };
signalHeaderVarConvF = {...
    @strtrim; @strtrim; @strtrim; ...
    @str2num; @str2num; @str2num; ...
    @str2num; @strtrim; @str2num; ...
    @strtrim };
num_signal_header_vars = length(signalHeaderVar);
num_signals = header.num_signals;
signalHeaderVarSize = [16; 80; 8; 8; 8; 8; 8; 80; 8; 32];
signalHeaderBlockSize = sum(signalHeaderVarSize)*num_signals;
signalHeaderVarLoc =
vertcat([0],cumsum(signalHeaderVarSize*num_signals));
signalHeaderRecordSize = sum(signalHeaderVarSize);

% Create Signal Header Struct
signalHeader = struct(...
    'signal_labels', {}, 'tranducer_type',
    {}, 'physical_dimension', {}, ...
    'physical_min', {}, 'physical_max', {}, 'digital_min',
    {}, ...
    'digital_max', {}, 'prefiltering', {}, 'samples_in_record',
    {}, ...
    'reserve_2', {});

% Get each signal header varaible
for v = 1:num_signal_header_vars

```

```

        varBlock =
A(signalHeaderVarLoc(v)+1:signalHeaderVarLoc(v+1))';
        varSize = signalHeaderVarSize(v);
        conF = signalHeaderVarConvF{v};
        for s = 1:num_signals
            varStart = varSize*(s-1)+1;
            varEnd = varSize*s;
            value = conF(char(varBlock(varStart:varEnd)));

            structCmd = ...
                sprintf('signalHeader(%.0f).%s = value;',s,
signalHeaderVar{v});
            eval(structCmd);
        end
    end
end % End Signal Load Section

%----- Load
Signal Block
if nargin >=3
    % Read digital values to the end of the file
    try
        % Set default error message
        errMsg = 'File load error. Check available memory.';

        % Load strategy is dependent on input
        if nargin == 1
            % Load entire file
            [A count] = fread(fid, 'int16');
        else
            % Get signal label information
            edfSignalLabels = arrayfun(...
                @(x)signalHeader(x).signal_labels,
[1:header.num_signals],...
                'UniformOutput', false);
            signalIndexes = arrayfun(...
                @(x)find(strcmp(x,edfSignalLabels)),
signalLabels,...
                'UniformOutput', false);

```



```

% Check that specified signals are present
signalIndexesCheck = cellfun(...
    @(x)~isempty(x), signalIndexes, 'UniformOutput',
false);

signalIndexesCheck =
int16(cell2mat(signalIndexesCheck));
if sum(signalIndexesCheck) == length(signalIndexes)
    % Indices are specified
    signalIndexes = cell2mat(signalIndexes);
else
    % Couldn't find at least one signal label
    errMsg = 'Could not identify signal label';
    error(errMsg);
end

edfSignalSizes = arrayfun(...
    @(x)signalHeader(x).samples_in_record,
[1:header.num_signals]);
edfRecordSize = sum(edfSignalSizes);

% Identify memory locations to record
endLocs = cumsum(edfSignalSizes)';
startLocs = [1;endLocs(1:end-1)+1];
signalLocs = [];
for s = signalIndexes
    signalLocs = [signalLocs;
[startLocs(s):1:endLocs(s)]]';
end
sizeSignalLocs = length(signalLocs);

% Load only required signals reduce memory calls
loadedSignalMemory = header.num_data_records*...
    sum(edfSignalSizes(signalIndexes));
A = zeros(loadedSignalMemory,1);
for r = 1:header.num_data_records
    [a count] = fread(fid, edfRecordSize, 'int16');
    A([1+sizeSignalLocs*(r-1):sizeSignalLocs*r]) =
a(signalLocs);

```

```

        end

        % Reset global variables, which enable reshape
functions to
        % work correctly
        header.num_signals = length(signalLabels);
        signalHeader = signalHeader(signalIndexes);
        num_signals = length(signalIndexes);
    end

    %num_data_records
    catch exception
        error(errMsg);
    end

    %----- Process
Signal Block
    % Get values to reshape block
    num_data_records = header.num_data_records;
    getSignalSamplesF = @(x)signalHeader(x).samples_in_record;
    signalSamplesPerRecord =
arrayfun(getSignalSamplesF,[1:num_signals]);
    recordWidth = sum(signalSamplesPerRecord);

    % Reshape - Each row is a data record
    A = reshape(A, recordWidth, num_data_records)';

    % Create raw signal cell array
    signalCell = cell(1,num_signals);
    signalLocPerRow =
horzcat([0],cumsum(signalSamplesPerRecord));
    for s = 1:num_signals
        % Get signal location
        signalRowWidth = signalSamplesPerRecord(s);
        signalRowStart = signalLocPerRow(s)+1;
        signalRowEnd = signalLocPerRow(s+1);

        % Create Signal
        signal = reshape(A(:,signalRowStart:signalRowEnd)',...
            signalRowWidth*num_data_records, 1);

```

```

    % Get scaling factors
    dig_min = signalHeader(s).digital_min;
    dig_max = signalHeader(s).digital_max;
    phy_min = signalHeader(s).physical_min;
    phy_max = signalHeader(s).physical_max;

    % Assign signal value
    value = signal;

    % Convert to physical units
    if RETURN_PHYSICAL_VALUES == 1
        % Convert from digital to physical values
        value = (signal-dig_min)/(dig_max-dig_min);
        value = value.*double(phy_max-phy_min)+phy_min;
    else
        fprintf('Digital to Physical conversion is NOT
performned: %s\n',...
            edfFN);
    end

    signalCell{s} = value;
end

end % End Signal Load Section

%----- Create
return value
if nargout < 2
    varargout{1} = header;
elseif nargout == 2
    varargout{1} = header;
    varargout{2} = signalHeader;
elseif nargout == 3

% Check if a reduce signal set is requested
if ~isempty(epochs)
    % Determine signal sampling rate

```

```

        signalSamples = arrayfun(...
            @(x) signalHeader(x).samples_in_record,
            [1:num_signals]);
        signalIndex = ones(num_signals, 1)*[epochs(1)-1
epochs(2)]*30;
        samplesPerSecond =
(signalSamples/header.data_record_duration)';
        signalIndex = signalIndex .* [samplesPerSecond
samplesPerSecond];
        signalIndex(:,1) = signalIndex(:,1)+1;

        % Redefine signals to include specified epochs
        signalIndex = int64(signalIndex);
        for s = 1:num_signals
            signal = signalCell{s};
            index = [signalIndex(s,1):signalIndex(s,2)];
            signalCell{s} = signal(index);
        end
    end

    % Create Output Structure
    varargout{1} = header;
    varargout{2} = signalHeader;
    varargout{3} = signalCell;
end % End Return Value Function

% Close file explicitly
if fid > 0
    fclose(fid);
end

end % End of blockEdfLoad function

```

Paso1_edf2json.m

```
clear
clc
%%
% guardamos directorios con los que se trabajara en memoria
thisdir          = pwd;
csvdir           = [thisdir '\DataHarvest'];
edfdir           = [thisdir '\edf'];
episodedir       = [thisdir '\Episodes'];
%creamos un directorio donde guardar los datos json que se
generaran
if ~exist('Episodes', 'dir')
    mkdir(episodedir);
end

% cargamos la lista que posee la lista de exámenes que se
procesaran, mas alguna informacion de los pacientes
% esta lista ya se encuentra anonimizada
[~, ~, List]     = xlsread([csvdir
'\ListaExámenes.xlsx']);

%reemplazamos algunos caracteres para poder trabajar en un struct
List(1,:) = regexp(List(1,:), ' ','_');
List(1,:) = regexp(List(1,:), '['','_');
List(1,:) = regexp(List(1,:), ']', '');
%transformamos la lista en un struct, que facilitara su
procesamiento
List          = cell2struct(
List(2:length(List(:,1)),:),List(1,:),2);

%procesamos la data
for i=1:1:length(List.EDF_Name)
    %obtenemos el nombre del archivo
    EDF_Name = List(i).EDF_Name;

    %cargamos el csv correspondiente a memoria
```

```

csv = csv2cell([csvdir '\\ EDF_Name '.csv'],'fromfile');
%eliminamos caracteres conflictivos
csv = regexprep(csv,' ','_');
%eliminamos expresiones conflictiva
csv = regexprep(csv,'_\\(.*\\)','');
%eliminamos columna extra vacia
csv = csv(:,1:7);
%transformamos a struct para trabajar mas facilmente
csv = cell2struct(csv(2:length(csv(:,1)),:),csv(1,:),2);

%cargamos el edf a memoria
[header, signalHeader, signalCell] = blockEdfLoad([edfdir
'\\ EDF_Name '.edf']);
%anonimizamos el edf
header.patient_id = EDF_Name;

%Obtenemos algunos datos relevantes
Age = List(i).Age;
Gender = List(i).Gender;

Start_time = [num2str(List(i).St_Hour) ':'
num2str(List(i).St_Min) ':' num2str(List(i).St_Sec)];
Start_time = datetime([List(i).Acq_Date ' '
Start_time],'Format','dd/MM/yyyy H:m:s');

%tiempo inicio
if Age < 18
    %saltamos el analisis si no es adulto

else
    %rescatamos los episodios
    episode = 1;
    for j = 1:length(csv)
        if ~strcmp(csv(j).Event_type, 'Central_apnea') &&
~strcmp(csv(j).Event_type, 'Obstructive_apnea') &&
~strcmp(csv(j).Event_type, 'Mixed_apnea') &&
~strcmp(csv(j).Event_type, 'Hypopnea')
            %saltamos cuando no es un evento relevante al
estudio
        else

```

```

        if strcmp(csv(j).Validation, '-')
            %saltamos el caso no validado por medico ni
por software
        else
            %creamos structs que tendran la informacion
            Data = struct;
            Tiempo = struct;
            Frecuencia = struct;

            %recogemos datos del episodio
            Event_type =
csv(j).Event_type;
            Epoch = csv(j).Epoch;
            Duration =
csv(j).Duration;
            Duration =
str2double(Duration);
            Stage = csv(j).Stage;
            Episode_time =
datetime([csv(j).Date ' ' csv(j).Time], 'Format','dd/MM/yyyy
H:m:s');

            % Obtenemos los datos del episodio
            g = 1;
            last_label = '';
            for k = 1:length(signalHeader(1,:))
                signal_label =
signalHeader(k).signal_labels;
                signal_label = strrep(signal_label,'
','_');
                signal_label = strrep(signal_label,'-
','_');

                %hay se¶s que repiten el nombre, para no
sobrecribir se realiza lo siguiente
                if strcmp(last_label,
signalHeader(k).signal_labels)
                    g = g+1;
                    signal_label = strcat(signal_label,
'_', num2str(g));
                else

```

```

        g=1;
    end
    last_label =
signalHeader(k).signal_labels;

    samples =
signalHeader(k).samples_in_record;
    if samples == 1
        %en caso de estar muestreado a 1Hz no
podemos tener un numero decimal en la duracion
        if mod(Duration,1) == 0.5
            Duration = Duration + 0.5;
        end
    end
    %rescato la seccion</span>
    aux = cell2mat(signalCell(k));
    aux = aux(seconds(Episode_time -
Start_time)*samples:(seconds(Episode_time -
Start_time)+Duration)*samples);
    signal = aux;

    % procesamos
    AX =
cumtrapz(signal);

    integral = struct('Mean',
mean(AX), 'Std', std(AX), 'Var', var(AX), 'Skewness',
skewness(AX), 'Kurtosis', kurtosis(AX));
    AX = signal;
    Fcn = struct('Mean',
mean(AX), 'Std', std(AX), 'Var', var(AX), 'Skewness',
skewness(AX), 'Kurtosis', kurtosis(AX));
    AX = diff(signal);
    Derivative = struct('Mean',
mean(AX), 'Std', std(AX), 'Var', var(AX), 'Skewness',
skewness(AX), 'Kurtosis', kurtosis(AX));
    % formar estructura de datos.
    Time =
struct('Data', signal,'Fs', 1/samples, 'Integral', integral,
'Signal', Fcn, 'Derivative',Derivative);

    % Frecuency Domain

```



```

                                [aux, f]           =
fourier(signal,samples);
                                % procesamos
                                AX                 = cumtrapz(aux);
                                integral           = struct('Mean',
mean(AX), 'Std', std(AX), 'Var', var(AX), 'Skewness',
skewness(AX), 'Kurtosis', kurtosis(AX));
                                AX                 = aux;
                                Fcn               = struct('Mean',
mean(AX), 'Std', std(AX), 'Var', var(AX), 'Skewness',
skewness(AX), 'Kurtosis', kurtosis(AX));
                                AX                 = diff(aux);
                                Derivative        = struct('Mean',
mean(AX), 'Std', std(AX), 'Var', var(AX), 'Skewness',
skewness(AX), 'Kurtosis', kurtosis(AX));
                                % formar estructura de datos
                                Frequency         = struct('Data',
aux, 'frecuencias', f(1:length(f)-1)', 'Integral', integral,
'Signal', Fcn, 'Derivative',Derivative);

                                eval(['Tiempo(:).' signal_label '=
Time;']);
                                eval(['Frecuencia(:).' signal_label '=
Frecuency;']);

                                %nombramos el episodio
                                if 10 > episode
                                    temp           = [EDF_Name
'_0000' num2str(episode) ];
                                elseif 10 <= episode && 100 > episode
                                    temp           = [EDF_Name
'_000'  num2str(episode) ];
                                elseif 100 <= episode && 1000 > episode
                                    temp           = [EDF_Name '_00'
num2str(episode) ];
                                elseif 1000 <= episode && 10000 >
episode
                                    temp           = [EDF_Name '_0'
num2str(episode) ];
                                elseif 10000 <= episode
                                    temp           = [EDF_Name '_ '
num2str(episode) ];

```

```

        end

        eval('Data(:).Time = Tiempo;');
        eval('Data(:).Frecuency = Frecuencia;');
    end

    %creamos la estructura de datos
    eval(['temp '=struct;']);
    eval(['temp '(:).Patient =
header.patient_id;'])
    eval(['temp '(:).Age = Age;']);
    eval(['temp '(:).Gender = Gender;']);
    eval(['temp '(:).Event_type = Event_type;']);
    eval(['temp '(:).Duration = Duration;']);
    eval(['temp '(:).Stage = Stage;']);
    eval(['temp '(:).Validation =
csv(j).Validation;']);
    eval(['temp '(:).Exam_time =
posixtime(Start_time);']); %fecha y hora de toma del examen,
formato Posix/Unix
    eval(['temp '(:).Time = Tiempo;']);
    eval(['temp '(:).Frecuency = Frecuencia;']);

    %guardamos al HD
    eval( ['json = savejson(' '''''' ',', temp
');' ] );

    fid = fopen(
[episodedir '\\' temp '.json'], 'wt');
    fprintf(fid, json);
    fclose all;
    clear json
    eval(['clear ' temp]);
end
end
episode = episode +1;
end
episode = 1;
end

```

end

Paso1_edf2json_Healthy.m

```
clear
clc
%%
thisdir          = pwd;
csvdir           = [thisdir '\DataHarvest'];
edfdir           = [thisdir '\edf'];
episodedir       = [thisdir '\Healthy'];

%creamos un directorio donde guardar los datos json que se
generaran
if ~exist('Healthy', 'dir')
    mkdir(episodedir);
end

% cargamos la lista que posee la lista de exámenes que se
procesaran, mas alguna informacion de los pacientes
% esta lista ya se encuentra anonimizada
[~, ~, List]     = xlsread([csvdir
'\ListaExámenes.xlsx']);

%reemplazamos algunos caracteres para poder trabajar en un struct
List(1,:) = regexprep(List(1,:), ' ', '_');
List(1,:) = regexprep(List(1,:), '[' , '_');
List(1,:) = regexprep(List(1,:), ']', '');
%transformamos la lista en un struct, que facilitara su
procesamiento
List        = cell2struct(
List(2:length(List(:,1)),:),List(1,:),2);

%procesamos la data
for i=1:1:length(List)
    %obtenemos el nombre del archivo
    EDF_Name = List(i).EDF_Name;

    %cargamos el csv correspondiente a memoria
    csv = csv2cell([csvdir '\' EDF_Name '.csv'],'fromfile');
```

```

%eliminamos caracteres conflictivos
csv = regexprep(csv, ' ', '_');
%eliminamos expresiones conflictiva
csv = regexprep(csv, '_\ (.*\)', '');
%eliminamos columna extra vacia
csv = csv(:,1:7);
%transformamos a struct para trabajar mas facilmente
csv = cell2struct(csv(2:length(csv(:,1)),:), csv(1,:), 2);

%cargamos el edf a memoria
[header, signalHeader, signalCell] = blockEdfLoad([edfdir
'\ EDF_Name '.edf']);
%anonimizamos el edf
header.patient_id = EDF_Name;

%Obtenemos algunos datos relevantes

Age = List(i).Age;
Gender = List(i).Gender;

Start_time = [num2str(List(i).St_Hour) ':'
num2str(List(i).St_Min) ':' num2str(List(i).St_Sec)];
Start_time = datetime([List(i).Acq_Date ' '
Start_time], 'Format', 'dd/MM/yyyy H:m:s');
%tiempo inicio
if Age < 18
    %saltamos el analisis si no es adulto

else
    episode = 1;

    [~, order] = sort(datetime(strcat({csv.Date}, '_ ',
{csv.Time}), 'Format', 'dd-MM-yyyy_H:m:s'));
    csv = csv(order);
    clearvars order
    timeline = datetime(strcat({csv.Date}, '_ ',
{csv.Time}), 'Format', 'dd-MM-yyyy_H:m:s');
    lapse = 30;

```

```

for j = 1:1:length(csv)-1

    if ~isnan(second(str2double(csv(j).Duration)))
        if seconds((timeline(j+1) - timeline(j))) >
ceil(str2double(csv(j).Duration)+lapse)
            %estare en espacios sin ningun evento, con un
margen de 15
            %desde el ultimo evento y 15 segundos al
proximo

            %veremos si es suficiente para alojar un
episodio (al menos 10 segundos)

            if seconds((timeline(j+1) - timeline(j))) -
lapse > 10

                %creamos structs que tendran la
informacion

                Data = struct;
                Tiempo = struct;
                Frecuencia = struct;

                %recogemos datos del episodio
                Event_type           = 'NO_EVENT';
                Epoch                 = [];
                Duration              = randi([10
40],1,1);

                Stage                = [];
                Episode_time         =
datetime(timeline(j), 'Format', 'dd/MM/yyyy H:m:s') +
seconds(ceil(str2double(csv(j).Duration)+lapse/2));

                % Obtenemos los datos del episodio
                g = 1;
                last_label = '';
                for k = 1:1:length(signalHeader(1,:))
                    signal_label =
signalHeader(k).signal_labels;
                    signal_label = strrep(signal_label, '
', '_');

```

```

        signal_label = strrep(signal_label, '-
', '_');
        signal_label =
strrep(signal_label, '.', '');

        %hay se¶s que repiten el nombre, para
no sobrecribir se realiza lo siguiente
        if strcmp(last_label,
signalHeader(k).signal_labels)
            g = g+1;
            signal_label =
strcat(signal_label, '_', num2str(g));
        else
            g=1;
        end
        last_label =
signalHeader(k).signal_labels;

        samples =
signalHeader(k).samples_in_record;
        if samples == 1
            %en caso de estar muestreado a
1Hz no podemos tener un numero decimal en la duracion
            if mod(Duration,1) == 0.5
                Duration = Duration + 0.5;
            end
        end
        %rescato la se¶/span>
        aux = cell2mat(signalCell(k));
        aux = aux(seconds(Episode_time -
Start_time)*samples:(seconds(Episode_time -
Start_time)+Duration)*samples);
        signal = aux;

        % procesamos
        AX =
cumtrapz(signal);
        integral =
struct('Mean', mean(AX), 'Std', std(AX), 'Var', var(AX),
'Skewness', skewness(AX), 'Kurtosis', kurtosis(AX));
        AX = signal;

```

```

                                Fcn                    =
struct('Mean', mean(AX), 'Std', std(AX), 'Var', var(AX),
'Skewness', skewness(AX), 'Kurtosis', kurtosis(AX));
                                AX                    =
diff(signal);

                                Derivative            =
struct('Mean', mean(AX), 'Std', std(AX), 'Var', var(AX),
'Skewness', skewness(AX), 'Kurtosis', kurtosis(AX));
                                % formar estructura de datos.
                                Time                    =
struct('Data', signal, 'Fs', 1/samples, 'Integral', integral,
'Signal', Fcn, 'Derivative', Derivative);

                                % Frecuency Domain
                                [aux, f]                =
fourier(signal, samples);

                                % procesamos
                                AX                    =
cumtrapz(aux);

                                integral                =
struct('Mean', mean(AX), 'Std', std(AX), 'Var', var(AX),
'Skewness', skewness(AX), 'Kurtosis', kurtosis(AX));
                                AX                    = aux;
                                Fcn                    =
struct('Mean', mean(AX), 'Std', std(AX), 'Var', var(AX),
'Skewness', skewness(AX), 'Kurtosis', kurtosis(AX));
                                AX                    = diff(aux);
                                Derivative            =
struct('Mean', mean(AX), 'Std', std(AX), 'Var', var(AX),
'Skewness', skewness(AX), 'Kurtosis', kurtosis(AX));
                                % formar estructura de datos
                                Frecuency                =
struct('Data', aux, 'frecuencias', f(1:length(f)-1)', 'Integral',
integral, 'Signal', Fcn, 'Derivative', Derivative);

                                eval(['Tiempo(:).' signal_label '=
Time;']);

                                eval(['Frecuencia(:).' signal_label
'= Frecuency;']);

                                %nombramos el episodio
                                if 10 > episode

```



```

temp = [EDF_Name
'_0000' num2str(episode) ];
elseif 10 <= episode && 100 > episode
temp = [EDF_Name
'_000' num2str(episode) ];
elseif 100 <= episode && 1000 >
episode
temp = [EDF_Name
'_00' num2str(episode) ];
elseif 1000 <= episode && 10000 >
episode
temp = [EDF_Name
'_0' num2str(episode) ];
elseif 10000 <= episode
temp = [EDF_Name
 '_' num2str(episode) ];
end

eval('Data(:).Time = Tiempo;');
eval('Data(:).Frecuency =
Frecuencia;');

end
%creamos la estructura de datos
eval(['temp '=struct;']);
eval(['temp '(:).Patient =
header.patient_id;'])
eval(['temp '(:).Age = Age;']);
eval(['temp '(:).Gender = Gender;'])
eval(['temp '(:).Event_type =
Event_type;']);
eval(['temp '(:).Duration = Duration;']);
eval(['temp '(:).Stage = [];']);
eval(['temp '(:).Validation = [];']);
eval(['temp '(:).Exam_time =
posixtime(Start_time);']); %fecha y hora de toma del examen,
formato Posix/Unix
eval(['temp '(:).Time = Tiempo;']);
eval(['temp '(:).Frecuency =
Frecuencia;']);

%guardamos al HD

```

```

temp ');' ] );
eval( ['json = savejson(' '''''' ',',
fid = fopen(
[episodesdir '\' temp '.json'], 'wt');
fprintf(fid, json);
fclose all;
clear json
eval(['clear ' temp]);
end
episode = episode +1;
end
end
end

episode = 1;
end

```

end

Paso2_json2mat.m

clc

clear

tic

%vamos al directorio

files = dir('Episodes');

num = length(files) - 2; %temporal

%SS Sleep Stage

ET = javaArray('java.lang.String',
num);

Age = javaArray('java.lang.Double',
num);

dur = javaArray('java.lang.Double',
num);

```

SS                                = javaArray('java.lang.String',
num);
val                                = javaArray('java.lang.String',
num);
Gender                             = javaArray('java.lang.String',
num);
Patient                             = javaArray('java.lang.String',
num);
Event_time                          = javaArray('java.lang.String',
num);
Exam_time                           = javaArray('java.lang.String',
num);

```

```

%% Estructura

```

```

mean                                = javaArray('java.lang.Double',
num);
std                                  = javaArray('java.lang.Double',
num);
var                                  = javaArray('java.lang.Double',
num);
skew                                 = javaArray('java.lang.Double',
num);
kurt                                 = javaArray('java.lang.Double',
num);

```

```

Derivative                          = struct('Mean', mean, 'Std',
std, 'Var', var, 'Skewness', skew, 'Kurtosis', kurt);

```

```

Function                          = struct('Mean', mean, 'Std',
std, 'Var', var, 'Skewness', skew, 'Kurtosis', kurt);

```

```

Integral                             = struct('Mean', mean, 'Std',
std, 'Var', var, 'Skewness', skew, 'Kurtosis', kurt);

```

```

Others                               = struct('Patient',
Patient, 'Age', Age, 'Gender', Gender, 'Exam_time', Exam_time,
'Event_type', ET, 'Event_time', Event_time, 'Duration', dur,
'Stage', SS, 'Validation', val);

```

```

Time                                 = struct('Integral',
Integral, 'Signal', Function, 'Derivative', Derivative);

```

```

Frecuency                                = struct('Integral',
Integral,'Signal', Function, 'Derivative', Derivative);

data                                       = [];

%debemos crear un arreglo para cada uno de las se¶s
data.Others                               = Others;
clearvars -except files Frecuency Time data

var1 = {'.Time' '.Frecuency'};
var2 = {'Integral' '.Signal' '.Derivative'};
var3 = {'.Mean' '.Std' '.Var' '.Skewness' '.Kurtosis'};

%% Recopilamos Datos
%Event_Type(1) = java.lang.String('hola'); para guardar.

for i = 1:1:length(files)
    if files(i).isdir == 0 &&
~isempty(strfind(files(i).name, '.json'))
        temp                               = loadjson(['Episodes\'
files(i).name]);

        eval(['data.Others.Patient(i-2)=java.lang.String(' ''''
temp.Patient '''' ');']);
        eval(['data.Others.Age(i-2)=java.lang.Double( temp.Age
);']);
        eval(['data.Others.Gender(i-2)=java.lang.String(' ''''
temp.Gender '''' ');']);
        eval(['data.Others.Event_type(i-2)=java.lang.String('
'''' temp.Event_type '''' ');']);
        eval(['data.Others.Duration(i-2)=java.lang.Double(
temp.Duration );']);
        eval(['data.Others.Stage(i-2)=java.lang.String(' ''''
temp.Stage '''' ');']);
        eval(['data.Others.Validation(i-2)=java.lang.String('
'''' temp.Validation '''' ');']);
        eval(['data.Others.Exam_time(i-2)=java.lang.String(' ''''
datestr(datetime(temp.Exam_time, 'ConvertFrom', 'posixtime'))
'''' ');']);

```

```

        eval(['data.Others.Event_time(i-2)=java.lang.String('
''' datestr(datetime(temp.Event_time, 'ConvertFrom',
'posixtime')) ''' ');]);
        names = fieldnames(temp.Time);

        for j = 1:length(names)

            if ~any(strcmp(names{j},fieldnames(data)))
                eval( [names{j} '= struct;']);
                eval( [names{j} '.Time = Time;']);
                eval( [names{j} '.Frecuency = Frecuency;']);
                eval( ['data.' names{j} '=' names{j} ';']);
            end

            for k = 1:length(var1)
                for l = 1:length(var2)
                    for m = 1:length(var3)
                        eval(['data.' names{j} var1{k} var2{l}
var3{m} '= cell(data.' names{j} var1{k} var2{l} var3{m} ');']);
                        eval(['data.' names{j} var1{k} var2{l}
var3{m} '{i-2}= temp' var1{k} '.' names{j} var2{l} var3{m} ');']);
                    end
                end
            end

        end

    else
        % se salta el caso
    end

    clearvars -except files Frecuency Time data i var1 var2 var3
end
%%

data.Others.Patient = cell(data.Others.Patient);
data.Others.Age = cell(data.Others.Age);
data.Others.Gender = cell(data.Others.Gender);

```

```
data.Others.Exam_time      = cell(data.Others.Exam_time);
data.Others.Event_type    = cell(data.Others.Event_type);
data.Others.Event_time    = cell(data.Others.Event_time);
data.Others.Duration      = cell(data.Others.Duration);
data.Others.Stage         = cell(data.Others.Stage);
data.Others.Validation    = cell(data.Others.Validation);

toc
clearvars -except data
save('datos.mat');
```

Paso2_json2mat_Healthy.m

```
clc
clear
tic

%vamos al directorio
files = dir('Healthy');

num = length(files) - 2; %temporal

%SS Sleep Stage
ET = javaArray('java.lang.String',
num);
Age = javaArray('java.lang.Double',
num);
dur = javaArray('java.lang.Double',
num);
SS = javaArray('java.lang.String',
num);
val = javaArray('java.lang.String',
num);
Gender = javaArray('java.lang.String',
num);
Patient = javaArray('java.lang.String',
num);
Event_time = javaArray('java.lang.String',
num);
Exam_time = javaArray('java.lang.String',
num);

%% Estructura

mean = javaArray('java.lang.Double',
num);
std = javaArray('java.lang.Double',
num);
var = javaArray('java.lang.Double',
num);
```

```

skew                                = javaArray('java.lang.Double',
num);
kurt                                = javaArray('java.lang.Double',
num);

```

```

Derivative                          = struct('Mean', mean, 'Std',
std, 'Var', var, 'Skewness', skew, 'Kurtosis', kurt);
Function                          = struct('Mean', mean, 'Std',
std, 'Var', var, 'Skewness', skew, 'Kurtosis', kurt);
Integral                            = struct('Mean', mean, 'Std',
std, 'Var', var, 'Skewness', skew, 'Kurtosis', kurt);

```

```

Others                              = struct('Patient',
Patient, 'Age', Age, 'Gender', Gender, 'Exam_time', Exam_time,
'Event_type', ET, 'Event_time', Event_time, 'Duration', dur,
'Stage', SS, 'Validation', val);

```

```

Time                                = struct('Integral',
Integral, 'Signal', Function, 'Derivative', Derivative);
Frecuency                          = struct('Integral',
Integral, 'Signal', Function, 'Derivative', Derivative);

```

```

data                                = [];

```

```

%debemos crear un arreglo para cada uno de las se□s

```

```

data.Others                        = Others;
clearvars -except files Frecuency Time data

```

```

var1 = {'.Time' '.Frecuency'};
var2 = {'.Integral' '.Signal' '.Derivative'};
var3 = {'.Mean' '.Std' '.Var' '.Skewness' '.Kurtosis'};

```

```

%% Recopilamos Datos

```

```

%Event_Type(1) = java.lang.String('hola'); para guardar.

```

```

for i = 1:1:length(files)
    if files(i).isdir == 0 &&
~isempty(strfind(files(i).name, '.json'))

```



```

temp = loadjson(['Healthy\'
files(i).name]);

eval(['data.Others.Patient(i-2)=java.lang.String(' ''
temp.Patient '' ' ');']);
eval(['data.Others.Age(i-2)=java.lang.Double( temp.Age
);']);
eval(['data.Others.Gender(i-2)=java.lang.String(' ''
temp.Gender '' ' ');']);
eval(['data.Others.Event_type(i-2)=java.lang.String('
'' temp.Event_type '' ');']);
eval(['data.Others.Duration(i-2)=java.lang.Double(
temp.Duration );']);
eval(['data.Others.Stage(i-2)=java.lang.String(' ''
temp.Stage '' ' ');']);
eval(['data.Others.Validation(i-2)=java.lang.String('
'' temp.Validation '' ');']);
eval(['data.Others.Exam_time(i-2)=java.lang.String(' ''
datestr(datetime(temp.Exam_time, 'ConvertFrom', 'posixtime'))
'' ');']);
eval(['data.Others.Event_time(i-2)=java.lang.String('
'' '' ');']);
names = fieldnames(temp.Time);

for j = 1:1:length(names)

    if ~any(strcmp(names{j},fieldnames(data)))
        eval( [names{j} '= struct;']);
        eval( [names{j} '.Time = Time;']);
        eval( [names{j} '.Frecuency = Frecuency;']);
        eval( ['data.' names{j} '=' names{j} ');']);
    end

    for k = 1:1:length(var1)
        for l = 1:1:length(var2)
            for m = 1:1:length(var3)
                eval(['data.' names{j} var1{k} var2{l}
var3{m} '= cell(data.' names{j} var1{k} var2{l} var3{m} ');']);
                eval(['data.' names{j} var1{k} var2{l}
var3{m} '{i-2}= temp' var1{k} '.' names{j} var2{l} var3{m} ');']);
            end
        end
    end
end

```

```

                                end
                            end
                        end

                    end
                else
                    % se salta el caso
                end
            clearvars -except files Frecuency Time data i var1 var2 var3
        end
    %%

    data.Others.Patient          = cell(data.Others.Patient);
    data.Others.Age              = cell(data.Others.Age);
    data.Others.Gender           = cell(data.Others.Gender);
    data.Others.Exam_time        = cell(data.Others.Exam_time);
    data.Others.Event_type       = cell(data.Others.Event_type);
    data.Others.Event_time       = cell(data.Others.Event_time);
    data.Others.Duration         = cell(data.Others.Duration);
    data.Others.Stage            = cell(data.Others.Stage);
    data.Others.Validation       = cell(data.Others.Validation);

    toc
    clearvars -except data
    save('datos_healthy.mat');

```

Paso3_mat2txt.m

```
clear
clc
load('datos.mat');
fields = fieldnames(data);
T = table();
for i = 1:1:length(fields)
    %Others tiene profundidad de 2
    subfields = fieldnames(eval(['data.' fields{i} ';' ]));
    for j = 1:1:length(subfields)
        if strcmp(fields(i), 'Others')
            T = [T cell2table(eval(['data.' fields{i} '.'
subfields{j}]), 'VariableNames', subfields(j))];
        else
            ssubfields = fieldnames(eval(['data.' fields{i} '.'
subfields{j} ';' ]));
            for k = 1:1:length(ssubfields)
                sssubfields = fieldnames(eval(['data.' fields{i}
'.' subfields{j} '.' ssubfields{k} ';' ]));
                for l = 1:1:length(sssubfields)
                    T = [T cell2table(eval(['data.' fields{i} '.'
subfields{j} '.' ssubfields{k} '.' sssubfields{l}]),
'VariableNames', {[fields{i} '__' subfields{j} '__' ssubfields{k}
'__' sssubfields{l}]})];
                end
            end
        end
    end
end
end
end
end
end
writetable(T, 'data.txt');
```

Paso3_mat2txt_Healthy.m

```
clear
clc
load('datos_healthy.mat');
fields = fieldnames(data);
T = table();
for i = 1:1:length(fields)
    %Others tiene profundidad de 2
    subfields = fieldnames(eval(['data.' fields{i} ';' ]));
    for j = 1:1:length(subfields)
        if strcmp(fields(i), 'Others')
            T = [T cell2table(eval(['data.' fields{i} '.'
subfields{j}]), 'VariableNames', subfields(j))];
        else
            ssubfields = fieldnames(eval(['data.' fields{i} '.'
subfields{j} ';' ]));
            for k = 1:1:length(ssubfields)
                sssubfields = fieldnames(eval(['data.' fields{i}
'.' subfields{j} '.' ssubfields{k} ';' ]));
                for l = 1:1:length(sssubfields)
                    T = [T cell2table(eval(['data.' fields{i} '.'
subfields{j} '.' ssubfields{k} '.' sssubfields{l}]),
'VariableNames', {[fields{i} '__' subfields{j} '__' ssubfields{k}
'__' sssubfields{l}]})];
                end
            end
        end
    end
end
end
end
end
writetable(T, 'data_Healthy.txt');
```

csv2cell.m

```
function data = csv2cell(varargin)
% CSV2CELL - parses a Windows CSV file into an NxM cell array,
% where N is
% the number of lines in the CSV text and M is the number of
% fields in the
% longest line of the CSV file. Lines are delimited by carriage
% returns
% and/or newlines.
%
% A Windows CSV file format allows for commas (,) and double
% quotes (") to
% be contained within fields of the CSV file. Regular fields are
% just text
% separated by commas (e.g. foo,bar,hello world). Fields
% containing commas
% or double quotes are surrounded by double quotes (e.g.
% foo,bar,"item1,item2,item3",hello world). In the previous
% example,
% "item1,item2,item3" is one field in the CSV text. For double
% quotes to be
% represented, they are written in pairs in the file, and
% contained within
% a quoted field, (e.g. foo,"this field contains
%"quotes"",bar). Spaces
% within fields (even leading and trailing) are preserved.
%
% All fields from the CSV file are returned as strings. If the
% CSV text
% contains lines with different numbers of fields, then the
% "missing"
% fields will appear as empty arrays, [], in the returned data.
% You can
% easily convert the data you expect to be numeric using
% str2num() and
% num2cell().
%
% Examples:
% >> csv2cell('foo.csv','fromfile') % loads and parses entire
% file
```

```

% >> csv2cell(',,,') % returns cell array {' ',' ',' ',' '}
% >> csv2cell(',,,','text') % same as above, declaring text
input
% >> csv2cell(sprintf('%s\r\n%s',...
%     '"Ten Thousand",10000,,"10,000",""It''s ""10 Grand""',
baby",10k',...
%     ',foo,bar,soo'))
% ans =
%     'Ten Thousand'     '10000'     ''     '10,000'     [1x22
char]     '10k'
%
%           ''     'foo'     'bar'     'soo'
[]         []
% >> % note the two empty [] cells, because the second line has
two fewer
% >> % fields than the first. The empty field '' at the
beginning of the
% >> % second line is due to the leading comma on that line,
which is
% >> % correct behavior. A trailing comma will do the same to
the end of a
% >> % line.
%
% Limitations/Exceptions:
% * This code is untested on large files. It may take a long
time due to
% variables growing inside loops (yes, poor practice, but easy
coding).
% * This code has been minimally tested to work with a variety
of weird
% Excel files that I have.
% * Behavior with improperly formatted CSV files is untested.
% * Technically, CSV files from Excel always separate lines
with the pair
% of characters \r\n. This parser will also separate lines that
have only
% \r or \n as line terminators.
% * Line separation is the first operation. I don't think the
Excel CSV
% format has any allowance for newlines or carriage returns
within
% fields. If it does, then this parser does not support it and
would not

```

```

% return bad output.
%
% Copyright 2008 Arthur Hebert

% Process arguments
if nargin == 1
    text = varargin{1};
elseif nargin == 2
    switch varargin{2}
        case 'fromfile'
            filename = varargin{1};
            fid = fopen(filename);
            text = char(fread(fid));
            fclose(fid);
        case 'text'
            text = varargin{1};
        otherwise
            error('Invalid 2nd argument %s. Valid options are
''fromfile'' and ''text''',varargin{2})
    end
else
    error('CSV2CELL requires 1 or 2 arguments.')
end

% First split it into lines
lines = regexp(text, '(\r\n|[\r\n])', 'split'); % lines should now
be a cell array of text split by newlines

% a character is either a delimiter or a field
inField = true;
inQuoteField = false;
% if inField && ~inQuoteField --> then we're in a raw field

skipNext = false;
data = {};
field = '';
for lineNumber = 1:length(lines)

```

```

    nChars = length(lines{lineNumber}); % number of characters in
this line
    fieldNumber = 1;
    for charNumber = 1:nChars
        if skipNext
            skipNext = false;
            continue
        end
        thisChar = lines{lineNumber}(charNumber);
        if thisChar == ','
            if inField
                if inQuoteField % this comma is part of the field
                    field(end+1) = thisChar;
                else % this comma is the delimiter marking the
end of the field
                    data{lineNumber,fieldNumber} = field;
                    field = '';
                    fieldNumber = fieldNumber + 1;
                end
                else % we are not currently in a field -- this is the
start of a new delimiter
                    inField = true;
                end
                if charNumber == nChars % this is a hanging comma,
indicating the last field is blank
                    data{lineNumber,fieldNumber} = '';
                    field = '';
                    fieldNumber = fieldNumber + 1;
                end
            elseif thisChar == '"'
                if inField
                    if inQuoteField
                        if charNumber == nChars % it's the last
character, so this must be the closing delimiter?
                            inField = false;
                            inQuoteField = false;
                            data{lineNumber,fieldNumber} = field;
                            field = '';
                            fieldNumber = fieldNumber + 1;

```



```

        else
            if lines{lineNumber}(charNumber+1) == '"'
% this is translated to be a double quote in the field
                field(end+1) = '"';
                skipNext = true;
            else % this " is the delimiter ending
this field
                data{lineNumber,fieldNumber} = field;
                field = '';
                inField = false;
                inQuoteField = false;
                fieldNumber = fieldNumber + 1;
            end
        end
        else % this is a delimiter and we are in a new
quote field
            inQuoteField = true;
        end
        else % we are not in a field. This must be an opening
quote for the first field?
            inField = true;
            inQuoteField = true;
        end
        else % any other character ought to be added to field
            field(end+1) = thisChar;
            if charNumber == nChars
                data{lineNumber,fieldNumber} = field;
                field = '';
                fieldNumber = fieldNumber + 1;
            elseif charNumber == 1 % we are starting a new raw
field
                inField = true;
            end
        end
    end
end
end
end

```

fourier.m

```
function [ X , f ] = fourier( x, Fs )
%FOURIER Summary of this function goes here
% entrega la transformada de fourier de x,

N          = length(x);
NFFT       = 2^nextpow2(N);
f          = Fs/2*linspace(0,1,1+NFFT/2);
aux        = fft(x,NFFT/2)/N;
X          = abs(aux(1:NFFT/2));

end
```

Anexo 2 – Configuración KNIME®

Disponible en <https://goo.gl/aE3ycH> (Google Drive)

Anexo 3 – Archivos Excel

Disponible en <https://goo.gl/PWXdRU> (Google Drive)

Anexo 4 – Prueba con *K-Fold Cross-Validation*

Se añade un *k-Fold cross-validation* con $k=10$ al experimento 1, para los casos de los clasificadores de la hipopnea. Este experimento pretende ilustrar la manera en que la aplicación de esta técnica afecta a los resultados.

Línea Base más *10-fold cross validation*

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	2625	732	2625	381	6363	64.40%	24.7071
DT	2908	526	4152	564	8150	86.60%	40.6993
SVM	0	0	4680	3472	8152	57.41%	1.3479
MLP	1426	1947	2733	2046	8152	51.02%	0.9783
PNN	1585	1236	3444	1887	8152	61.69%	2.3405
NB	1729	1075	3605	1743	8152	65.43%	3.3266

TABLA 59: MATRIZ DE CONFUSIÓN PARA LÍNEA BASE UTILIZANDO *CROSS-VALIDATION*, HIPOPNEA.

Esta tabla ha de compararse con Tabla 3.

Reducción por random forest más *10-fold cross validation*

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	2618	412	3852	608	7490	79.37%	40.2583
DT	2862	597	4083	609	8151	85.19%	32.1409
SVM	0	0	4680	3472	8152	57.41%	1.3479
MLP	1295	1768	2912	2175	8150	51.61%	0.9807
PNN	1680	1531	3149	1792	8152	59.24%	1.9283
NB	2384	1770	2910	1088	8152	64.94%	3.6024

TABLA 60: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR RANDOM FOREST UTILIZANDO *CROSS-VALIDATION*, HIPOPNEA

Esta tabla ha de compararse con Tabla 7.

Reducción por tree ensemble más *10-fold cross validation*

Hipopnea	TP	FP	TN	FN	Total	Exactitud	DOR
FR	2611	397	3809	630	7447	78.75%	39.7637
DT	2849	590	4088	622	8149	85.10%	31.7366
SVM	0	0	4680	3472	8152	57.41%	1.3479
MLP	2027	2711	1969	1445	8152	49.02%	1.0188
PNN	1680	1531	3149	1792	8152	59.24%	1.9283
NB	2315	1722	2958	1157	8152	64.68%	3.4370

TABLA 61: MATRIZ DE CONFUSIÓN PARA PRUEBA CON REDUCCIÓN DE CARACTERÍSTICAS POR TREE ENSEMBLE UTILIZANDO *CROSS-VALIDATION*, HIPOPNEA.

Esta tabla ha de compararse con Tabla 11.

Discusión de los resultados.

En algunos casos el método de *k-fold cross-validation* ofrece mejores índices que los que reporta el experimento sin este paso, pero por la naturaleza misma del proceso, que implica entrenar el clasificador bajo distintos conjuntos de entrenamiento para simular una mayor cantidad de datos. Dadas estas condiciones, no se obtiene un clasificador que se pueda utilizar sobre el conjunto de pruebas.