



UNIVERSIDAD
NACIONAL
DE COLOMBIA

**Metodología para el diseño y la
implementación de sensores inferenciales
basados en datos de proceso, desarrollados
con redes neuronales artificiales, sistemas
neuro-difusos y máquinas de vectores de
soporte**

Jorge Eduardo Cote Ballesteros

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, Colombia
2017

Metodología para el diseño y la implementación de sensores inferenciales basados en datos de proceso, desarrollados con redes neuronales artificiales, sistemas neuro-difusos y máquinas de vectores de soporte

Jorge Eduardo Cote Ballesteros

Tesis o trabajo de investigación presentado como requisito parcial para optar al título de:
Magister en Ingeniería – Automatización Industrial

Director:

Victor Hugo Grisales Palacio, Ph.D

Codirector:

Iván Dario Gil Chaves, Ph.D

Línea de Investigación:

Inteligencia Computacional y Automática

Grupo de Investigación:

Grupo de automática de la Universidad Nacional, GAUNAL

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica

Bogotá, Colombia

2017

Este documento está dedicado a mi familia por su apoyo incondicional en mi desarrollo profesional y personal.

Agradecimientos

Agradezco el apoyo brindado para el desarrollo de este trabajo a todas aquellas personas que de una u otra manera intervinieron para la finalización del mismo.

Agradezco a mis padres Virginia Ballesteros y Jorge Cote por su paciencia y apoyo en el desarrollo de mis estudios y en particular de este proyecto.

A mi director de Tesis Victor Hugo Grisales, Ph.D, agradezco su apoyo y guía en el desarrollo del presente trabajo, así como en mi formación académica e investigativa.

A mi co-director Iván Dario Gil, Ph.D agradezco su invaluable aporte en mi trabajo, el compartir sus conocimientos y su tiempo para el desarrollo de esta tesis.

En general agradezco a todos quienes aportaron de alguna manera en el desarrollo y consecución de los objetivos propuestos en esta tesis.

Resumen

El presente trabajo de tesis de maestría se desarrolló para el establecimiento de una metodología, que permita el diseño e implementación de sensores inferenciales basados en datos, bajo tres técnicas seleccionadas, por la creciente atención que vienen despertando en tareas de modelamiento y estimación de variables industriales. Las técnicas seleccionadas son las redes neuronales artificiales, las máquinas de vectores de soporte y los sistemas de inferencia neuro-difusa, que adicionalmente utilizan bases distintas para la solución del problema de regresión. Teniendo en cuenta lo anterior, se diseñaron 3 sensores inferenciales basados en estas técnicas, explorando las etapas necesarias para establecer una predicción de buen desempeño y por consiguiente diferentes técnicas para el preprocesamiento de los datos, selección de modelo y el entrenamiento y validación de las técnicas. Para llevar a cabo estas tareas se obtuvo el modelo matemático de una columna de destilación con el fin de acercar el entorno de simulación a datos con características típicas del ambiente industrial. Con el desarrollo de los 3 sensores inferenciales se estableció una metodología que puede ser aplicada en la industria y que desemboca en un desempeño adecuado para predecir variables de difícil o impráctica medición.

Palabras clave: ANN, SVM, ANFIS, Sensor Inferencial, regresión, entrenamiento, validación.

Abstract

The present master's thesis work was developed for the establishment of a methodology, which allows the design and implementation of data-based inferential sensors, under three selected techniques. Due to the increasing attention they have been arousing in modeling and estimation of industrial variables. The techniques selected are artificial neural networks, support vector machines and neuro-diffuse inference systems, which additionally use different methods for the solution of the regression problem. Taking into account the above, 3 inferential sensors based on these techniques has been designed, by exploring the steps to establish a prediction of good performance and by means of different techniques for pre-processing of data, model selection and training of the techniques. To carry out these tasks, the mathematical model of a distillation column was obtained in order to bring the simulation environment closer to data with typical characteristics of the industrial environment. With the development of the 3 inferential sensors, a methodology was established that can be applied in the industry and that leads to adequate performance to predict variables of difficult or impractical measurement.

Keywords: ANN, SVM, ANFIS, soft-sensor, regression,

Contenido

	Pág.
Resumen	IX
Lista de figuras.....	XIV
Lista de tablas	XVII
Lista de Símbolos.....	19
1. Introducción	21
1.1 Contexto	21
1.1.1 Predicción en tiempo real.....	22
1.1.2 Monitoreo de procesos y detección de fallas.....	23
1.1.3 Detección de falla de sensores	23
1.2 Motivación	23
1.3 Antecedentes.....	24
1.3.1 Redes Neuronales Artificiales (ANN).....	24
1.3.2 Sistemas de Inferencia Neuro-Difusa (ANFIS)	26
1.3.3 Máquinas de vectores de soporte (SVM).....	27
1.4 Objetivos.....	28
1.5 Contribuciones.....	29
1.6 Organización del documento	30
2. Técnicas de Aprendizaje de Máquina	33
2.1 Redes Neuronales Artificiales (ANN)	34
2.1.1 Arquitectura.....	35
2.1.2 Entrenamiento.....	35
2.2 Sistemas Neuro-Difusos de inferencia (ANFIS)	39
2.2.1 Arquitectura.....	40
2.2.2 Entrenamiento.....	41
2.3 Máquinas de vectores de soporte (SVM)	48
2.3.1 Regresión Lineal	49
2.3.2 Kernel	51
2.3.3 Regresión no lineal	53
2.3.4 Entrenamiento.....	55
2.4 Análisis Comparativo	59
2.4.1 Arquitectura.....	60
2.4.2 Entrenamiento y Convergencia	67
2.4.3 Calidad de datos	71
2.5 Conclusiones	77

3.	Generación de datos – Columna de Destilación Binaria.....	79
3.1	Columna de destilación binaria.....	80
3.1.1	Modelo.....	83
3.1.2	Implementación en código	87
3.1.3	Resultados Modelo	87
3.2	Generación de conjunto de datos.....	90
3.3	Conclusiones.....	95
4.	Pre - procesamiento de Datos.....	97
4.1	Datos Fuera de tendencia “Outliers”	97
4.1.1	Detección de Datos fuera de Tendencia	97
4.1.2	Técnicas de Filtrado.....	98
4.1.3	Comparativo de Técnicas de Filtrado.....	99
4.2	Reducción dimensión conjunto de datos	104
4.2.1	Selección de variables o características	104
4.2.2	Extracción de características	117
4.3	Conclusiones.....	123
5.	Selección de Modelo.....	125
5.1	Estructuras de modelos dinámicos	125
5.1.1	NARX	126
5.1.2	NARMAX	127
5.1.3	Selección de auto-regresores	129
5.1.4	Modelos obtenidos.....	133
5.2	Comparativo de modelos.....	134
5.4	Conclusiones	135
6.	Metodología.....	137
6.1	Etapas Fundamentales.....	137
6.1.1	Preparación del conjunto de datos.....	139
6.1.2	Preprocesamiento de datos	140
6.1.3	Selección de modelo.....	141
6.1.4	Entrenamiento y validación del modelo.....	142
6.2	Sensores inferenciales obtenidos.....	144
6.2.1	Sensor Inferencial basado en ANN.....	145
6.2.2	Sensor Inferencial basado en SVM.....	145
6.2.3	Sensor Inferencial basado en ANFIS	146
6.3	Pruebas de desempeño	149
6.3.1	Datos fuera de tendencia	149
6.3.2	Muestreo Insuficiente.....	150
6.3.3	Mínimo número de datos de entrenamiento.....	151
6.4	Conclusiones.....	153
7.	Conclusiones	155
7.1	Resumen del proyecto.....	155
7.2	Conclusiones generales	156
7.3	Trabajo futuro.....	157
8.	Anexos.....	159
8.1	Código Redes Neuronales Artificiales (ANN).....	159

8.2	Código Sistemas de Inferencia ANFIS	161
8.3	Estimación parámetros Takagi-Sugeno	163
8.4	Código Algoritmo clustering substractivo	164
8.5	Algoritmo clustering substractivo	166
8.6	Código Máquinas de vectores de soporte (SMV)	167
9.	Bibliografía	170

Lista de figuras

	Pág.
Figura 2.1 Características de una neurona biológica y su modelo básico por McCulloch y Pitts. Tomada de (Babuška, 2009).....	34
Figura 2.2 Arquitectura red neuronal artificial perceptrón multicapa	35
Figura 2.3 Diagrama de flujo del entrenamiento de ANN por retro-propagación.....	36
Figura 2.4 Notación para la arquitectura de una red neuronal artificial tipo perceptrón multicapa	37
Figura 2.5 Estructura de un sistema difuso. Tomado de (Funsten, 2015; J. S. R. Jang, 1993)	40
Figura 2.6 Arquitectura sistema ANFIS (J. S. R. Jang, 1993).....	41
Figura 2.7 Diagrama de Flujo para entrenamiento de red ANFIS por algoritmo de retro-propagación híbrido	43
Figura 2.8 Interpretación gráfica de los parámetros de las funciones de pertenencia (Hamdan, 2013)	45
Figura 2.9 Interpretación grafica de la regresión lineal por SVM (A. Smola et al., 1996). 49	49
Figura 2.10 Mapeo de conjunto de datos original a espacio de características	52
Figura 2.11 Interpretación gráfica de la regresión no lineal por SVM	54
Figura 2.12 Diagrama de flujo para entrenamiento de máquinas de vectores de soporte por algoritmo Kernel-Adatron	58
Figura 2.13 Conjuntos de datos de entrenamiento a) Sistema estático (Superficie) b) Sistema dinámico (Serie de Box-Jenkins)	60
Figura 2.14 Arquitectura a) ANN, b) SVM y c) ANFIS. Tomado de (Bishop, 2006; J. S. R. Jang, 1993; Alexander J Smola & Schölkopf, 2004).....	60
Figura 2.15 Variación parámetro η en ANN a) Sistema estático, b) Sistema dinámico....	63
Figura 2.16 Variación parámetro η en ANFIS a) Sistema estático, b) Sistema dinámico.63	63
Figura 2.17 Variación parámetro α en ANN a) Sistema estático b) Sistema Dinámico	65
Figura 2.18 Variación parámetro σ en kernel SVM a) Sistema estático b) Sistema dinámico	65
Figura 2.19 Variación parámetro p kernel polinómico SVM a) Sistema Estático b) Sistema Dinámico.....	66
Figura 2.20 Variación parámetro máxima desviación ϵ en SVM a) Sistema Estático b) Sistema Dinámico.....	67
Figura 2.21 Entrenamiento y validación para análisis de interpolación a) ANN, b) SVM, c) ANFIS.....	69
Figura 2.22 Entrenamiento y validación sistema estático para análisis mínima cantidad de datos a) ANN, b) SVM, c) ANFIS	73

Figura 2.23 Entrenamiento y validación sistema dinámico para análisis mínima cantidad de datos a) ANN, b) SVM, c) ANFIS.....	73
Figura 3.1 Esquema general de columna de destilación binaria. Tomado de (Alzate Ibañez, 2010)	80
Figura 3.2 Flujo de líquido y vapor en el interior de columna de destilación. (Alzate Ibañez, 2010)	81
Figura 3.3 Perfil de temperatura obtenido.....	88
Figura 3.4 a) Concentración molar de etanol en cima. b) Concentración molar de etanol en fondo	89
Figura 3.5 Variación paramétrica de concentración molar de etanol a través de flujo molar de alimentación. a) Cima, b) Fondo.....	91
Figura 3.6 Variación paramétrica de concentración molar de etanol a través de concentración molar de alimentación. a) Cima, b) Fondo	91
Figura 3.7 Variación paramétrica de concentración molar de etanol a través de relación de reflujo (R). a) Cima, b) Fondo.	92
Figura 3.8 Variación paramétrica de concentración molar de etanol a través de flujo de destilado (D). (arriba) Cima, (abajo) Fondo.	92
Figura 3.9 Secuencia pseudo-aleatoria aleatoria (PRBS).....	93
Figura 3.10 Fragmento de serie de tiempo generada a) concentración molar etanol, b) Temperatura etapa 1 c) Relación de reflujo variada a través de PRBS	94
Figura 3.11 Serie completa de tiempo para la concentración de etanol del modelo matemático de la columna de destilación binaria.....	95
Figura 4.1 Error de datos filtrados: a)regla 3- σ con 2% “outliers”, b) MAD con 2% “outliers”, c) Filtro Hampel con 2% “outliers”, d) regla 3- σ con 25% “outliers”,e) MAD con 25% “outliers”, f) Filtro Hampel con 25% “outliers”.....	102
Figura 4.2 Error de datos filtrados: a)regla 3- σ con 2.5 σ de amplitud, b) MAD con 2.5 σ de amplitud, c) Filtro Hampel con 2.5 σ de amplitud, d) regla 3- σ con 3.5 σ de amplitud , e) MAD con 3.5 σ de amplitud, f) Filtro Hampel con 3.5 σ de amplitud.	103
Figura 4.3 Diagrama de flujo algoritmo MIFS. Adaptado de (Battiti, 1994).....	109
Figura 4.4 Proceso aplicación MIFS	114
Figura 4.5 Interpretación geométrica PCA.....	118
Figura 5.1 Esquema sistema NARX. Tomado de (Acuña et al., 2014).....	126
Figura 5.2 Esquema sistema NARMAX. Tomado de (Acuña et al., 2014).	128
Figura 5.3 Metodología de entrenamiento sistema NARMAX. Tomado de (Acuña et al., 2014).....	128
Figura 5.4 Metodología Hill Climbing. Tomado de (Kohavi & John, 1997).	131
Figura 6.1 Fases de desarrollo de sensor inferencial. Tomado de (Kadlec, 2009).....	138
Figura 6.2 Etapas fundamentales para el desarrollo de un sensor inferencial	139
Figura 6.3 Esquema de metodología de preprocesamiento de datos	141
Figura 6.4 Metodología de diseño de un sensor inferencial basado en ANN, SVM o ANFIS	143
Figura 6.5 Resultados de validación de sensor inferencial ANN. a) Validación concentración de etanol. b) Error residual conjunto de datos de validación y predicción sensor inferencial	145

Figura 6.6 Resultados de validación de sensor inferencial SVM. a) Validación concentración de etanol. b) Error residual conjunto de datos de validación y predicción sensor inferencial.....	146
Figura 6.7 Resultados de validación de sensor inferencial ANFIS. a) Validación concentración de etanol. b) Error residual conjunto de datos de validación y predicción sensor inferencial.....	147
Figura 6.8 Validación de sensores frente a conjunto del 25% del total. a) Validación ANN, b) Error residual de predicción para ANN c) Validación SVM, d)Error residual de predicción para SVM, e)Validación ANFIS, f)Error residual de predicción para ANFIS .	151
Figura 6.9 Comportamiento del MSE de validación para conjuntos con diferentes números de muestras. a) ANN, b) SVM, c) ANFIS.....	152

Lista de tablas

	Pág.
Tabla 2.1 Métodos de entrenamiento ANFIS.....	42
Tabla 2.2 MSE de entrenamiento y validación para ANN y ANFIS frente a variación de cantidad de neurodos y reglas respectivamente. Sistema estático.....	61
Tabla 2.3 MSE de entrenamiento y validación para ANN y ANFIS frente a variación de cantidad de neurodos y reglas respectivamente. Sistema dinámico.....	62
Tabla 2.4 MSE de entrenamiento para ANN, SVM y ANFIS frente a número de iteraciones necesarias.....	68
Tabla 2.5 Tiempo de ejecución de algoritmos entrenamiento ANN, SVM y ANFIS. Sistema estático.....	70
Tabla 2.6 Tiempo de ejecución de algoritmos entrenamiento ANN, SVM y ANFIS. Sistema dinámico.....	70
Tabla 2.7 MSE de entrenamiento y validación para ANN, SVM y ANFIS frente a conjuntos con muestreo insuficiente. Sistema estático.....	71
Tabla 2.8 MSE de entrenamiento y validación para ANN, SVM y ANFIS frente a conjuntos con muestreo insuficiente. Sistema dinámico.....	72
Tabla 2.9 MSE de validación para conjuntos con datos fuera de tendencia. Sistema estático.....	74
Tabla 2.10 MSE de entrenamiento y validación frente a pérdida de datos en el conjunto. Sistema estático.....	75
Tabla 2.11 MSE de entrenamiento y validación para ANN frente a pérdida de datos en el conjunto. Sistema dinámico.....	75
Tabla 2.12 MSE de entrenamiento y validación frente a conjuntos de datos con ruido. Sistema estático.....	76
Tabla 2.13 MSE de entrenamiento y validación frente a conjuntos de datos con ruido. Sistema dinámico.....	76
Tabla 3.1 Características columna de destilación. Tomado de (Alzate Ibañez, 2010).....	82
Tabla 3.2 Características hidráulicas de la columna de destilación. Tomado de (Alzate Ibañez, 2010).....	82
Tabla 3.3 Condiciones de estado estable de columna de destilación. Tomado de (Alzate Ibañez, 2010).....	83
Tabla 3.4 Complejidad del modelo.....	87
Tabla 3.5 Resultados perfil de temperatura modelo columna de destilación.....	88
Tabla 3.6 Valores de estado estable para concentración molar de etanol.....	89

Tabla 4.1 Detección de datos fuera de tendencia con las técnicas 3-sigma, MAD y filtro de Hampel	101
Tabla 4.2 Error promedio (ME) con diferentes filtros	102
Tabla 4.3 Variables conjunto de datos inicial	110
Tabla 4.4 Variables conjunto de datos reducido por conocimiento experto	111
Tabla 4.5 Coeficientes de correlación para conjunto de 20 variables	112
Tabla 4.6 MSE de entrenamiento y validación para ANN, SVM y ANFIS frente a eliminación de variables	113
Tabla 4.7 Porcentaje de variación de MSE frente a la eliminación de variables candidatas. Para ANN, SVM y ANFIS	114
Tabla 4.8 Primeras variables del ranking de MIFS	115
Tabla 4.9 Coeficiente de ajuste y MSE de validación para diferentes subconjuntos de variables	116
Tabla 4.10 Características obtenidas para modelo con sistema ANFIS	116
Tabla 4.11 Varianza de variables en el espacio de los valores propios	120
Tabla 4.12 Porcentaje de varianza acumulada para los primeros 5 componentes del conjunto	121
Tabla 4.13 Variables de mayor peso dentro de los primeros 5 componentes principales	122
Tabla 5.1 Correspondencia de variables y notación de sistemas auto-regresivos	130
Tabla 5.2 Subconjuntos obtenidos para estructura NARX en metodología "Hill climbing" y métricas de desempeño	132
Tabla 5.3 Subconjuntos para estructura NARMAX en metodología "Hill climbing" y métricas de desempeño	133
Tabla 5.4 Variables obtenidas para cada técnica de aprendizaje de máquina en estructura NARX	133
Tabla 5.5 Variables obtenidas para cada técnica de aprendizaje de máquina en estructura NARMAX	134
Tabla 5.6 MSE de validación con técnicas ANN, SVM y ANFIS bajo estructura NARX y NARMAX	135
Tabla 6.1 Comparativo desempeño sensores inferenciales basados en ANN, SVM y ANFIS	148
Tabla 6.2 MSE de validación con conjuntos contaminados con datos fuera de tendencia	149
Tabla 6.3 MSE de validación frente a conjuntos con pocas muestras	150
Tabla 6.4 Cantidades mínimas de datos para conseguir una predicción adecuada	152

Lista de Símbolos

η	Tasa de aprendizaje
θ	Parámetro de regresión
\hat{y}	Predicción
Π	Productoria
Σ	Sumatoria o Matriz de covarianza
α	Parámetro exponencial
σ	Ancho de función Gaussiana o desviación típica
μ_{A_i}	Función de pertenencia
ξ y ξ^*	Margen de puntos desviados de SVM
ε	Margen de error de predicción
$\langle x_i, x \rangle$	Producto punto
α_i, α_i^*	Vectores de soporte
$\Phi(x)$	x mapeado a espacio de características
$k(x, z)$	Kernel
L	Lagrangiano
x_i	Conjunto de datos de entrada o concentración de etanol líquido
y_i	Conjunto de datos de salida o concentración de vapor de etanol
L	Flujo molar de líquido
B	Flujo molar de fondo
D	Flujo molar de destilado

F	Flujo molar de alimentación
M	Masa retenida en cada plato
z_f	Concentración molar de alimentación
μ	Media aritmética
ρ	Correlación
β	Peso del coeficiente de información mutua en MIFS
C_p	Coeficiente Cp Mallows

1. Introducción

Medir el valor de las diferentes variables que intervienen en un proceso, en cualquier instante de tiempo, es el pilar fundamental para poder desarrollar alguna estrategia de control sobre el mismo. Es por esto que a lo largo de la historia se han desarrollado múltiples técnicas y dispositivos para la medición de variables fundamentales como temperatura, presión, flujo, entre otras, las cuales son comunes a muchos procesos industriales. De otro lado existen variables que son relativas a cada proceso, relacionadas con aspectos como concentraciones de material, texturas, tamaños e índices de calidad de productos finales. Para algunos de estos casos, se presentan dificultades en su medición entre las que se pueden contar, altos costos de los dispositivos de medición, dificultades de calibración de los sensores existentes o que simplemente no existe la tecnología con la cual se pueda medir la variable de interés en tiempo real, sino que es necesario esperar resultados de laboratorio para obtener datos de la variable en cuestión. Como una solución a esto se han propuesto numerosos trabajos que implementan técnicas matemáticas que realizan una estimación de la variable de interés a partir de variables de fácil medición como temperatura, presión, flujo y nivel; denominados sensores inferenciales o sensores virtuales.

1.1 Contexto

Los denominados sensores inferenciales, son diseñados e implementados a partir de 2 enfoques principales, los cuales son, sensores inferenciales basados en modelo y sensores inferenciales basados en datos.

El enfoque basado en modelo requiere de las ecuaciones fenomenológicas para poder establecer el comportamiento del sistema y de esta manera poder estimar las variables de interés, este enfoque depende que se haga un buen modelado incluyendo reacciones químicas si el proceso las requiere, lo que supone la necesidad de conocer todos los detalles del proceso (Kadlec, Gabrys, & Strandt, 2009). En contraste, el enfoque basado en datos presenta una alternativa para relacionar datos de entrada y salida por medio de diferentes técnicas de optimización o inteligencia computacional, de esta manera realiza la predicción de la variable de interés a partir de la instrumentación disponible en el proceso (Kadlec et al., 2009; Sliškovíc, Grbic, & Hocenski, 2011a).

En el enfoque basado en datos, existe especial interés por el uso de técnicas como las redes neuronales artificiales, sistemas neuro - difusos y más recientemente las máquinas

de vector de soporte, por su característica de modelar sistemas no lineales mediante regresiones y la capacidad de generalización inherente a estas técnicas.

Aplicaciones

Los sensores inferenciales, también conocidos como sensores virtuales, son empleados en la industria, en múltiples funciones donde hay interés de estimar el valor de algunas variables importantes del proceso, entre las aplicaciones más frecuentes de los sensores virtuales tenemos:

1.1.1 Predicción en tiempo real

Es una de las aplicaciones más comunes de los sensores inferenciales, consiste en estimar el valor de variables de proceso cuya medición resulta imposible o impráctica por razones como altos costos de la sensorica, frecuentes problemas del sensor (descalibración, averías, cambio de partes, etc.), las mediciones se realizan en laboratorio por lo tanto hay retardos en la toma de decisiones o porque definitivamente no existe una tecnología para realizar mediciones de la variable en tiempo real.

Las variables que más frecuentemente se miden con sensores inferenciales son aquellas que están relacionadas con la calidad del producto, por ejemplo en la industria de fabricación de plásticos es de interés medir el índice de fusión del plástico (melt index) (Chen, Gao, & Chen, 2004) que es un indicador de la calidad del mismo, en esta misma industria también es un indicador de calidad el grado de polimerización del plástico (Huang, Zhu, & Tao, 2010) para lo cual no existe un sensor físico y por lo tanto es práctico implementar sensores inferenciales. En el mismo sentido, en la industria petroquímica es deseable conocer la concentración de componentes de los subproductos del petróleo entre los que se ha encontrado aplicaciones en la medición de la concentración de C5 y C4 en la producción de gasolina (Fortuna, Graziani, & Xibilia, 2005), e índices de calidad como el ASTM 95% (Bolf, Ivandic, & Galinec, 2008) que es estándar dentro de la producción de hidrocarburos.

Teniendo en cuenta lo anteriormente dicho, es posible observar que los sensores inferenciales tienen aplicaciones prácticamente en cualquiera de los sectores de procesos permitiendo extraer una estimación de la información de calidad del producto final lo cual trae consigo beneficios en costos debido a que en un caso se puede evitar la compra de sensores físicos en extremo costosos y de otro lado permite la toma de decisiones oportunas evitando pérdidas de producto por mala calidad.

1.1.2 Monitoreo de procesos y detección de fallas

Otra aplicación de los sensores virtuales es el monitoreo de procesos, donde el sistema de inferencia es entrenado para indicar si el proceso está operando en condiciones normales de funcionamiento y en caso de falla identificaría las posibles causas. Para este tipo de aplicaciones se utilizan frecuentemente técnicas de estadística multivariable, con las cuales se analizarían los datos que entregue el proceso.

Este tipo de aplicaciones son implementadas como una herramienta para los operadores del proceso facilitando el manejo de los altos volúmenes de información que entrega el proceso y la detección de los puntos donde el proceso está siendo llevado a operar fuera de sus condiciones normales, lo cual traerá mayor rapidez en la toma de decisiones por parte del personal que este encargado de operar el proceso (Kadlec et al., 2009).

Se han reportado diferentes trabajos en la aplicación de estos sensores inferenciales para el monitoreo de procesos, entre los cuales se resalta, los resultados presentados en (Chang, Lv, Wang, Mao, & Wang, 2006), donde se plantean sensores inferenciales para la detección de fallas en una torre de destilación, desarrollados mediante redes neuronales artificiales, alimentadas por múltiples variables de proceso y realizando la predicción de 3 de las fallas más comunes en este tipo de procesos químicos.

1.1.3 Detección de falla de sensores

Por último, los sensores virtuales también son utilizados en la predicción de falla de los sensores presentes en el proceso, para realizar cambios y mantenimiento en la sensorica de una manera oportuna y evitando que se presenten problemas en el proceso a causa del mal funcionamiento de algunos de los sensores físicos (Kadlec, 2009; Kadlec et al., 2009).

Sin embargo, debido a que la correlación entre las variables de proceso y el estado de los sensores es baja, las técnicas que se utilizan para este tipo de aplicaciones deben representar comportamientos dinámicos bastante particulares, lo cual implica una alta dificultad en la implementación de estos sensores inferenciales.

1.2 Motivación

En las últimas décadas, los sensores inferenciales han tomado cierta relevancia en procesos industriales donde existen variables de difícil o impráctica medición. Esto debido principalmente a sus características de bajo error en la estimación de las variables y bajos costos de implementación en el modelamiento de procesos continuos de carácter industrial, por ejemplo, aquellos donde se desea conocer el valor de diferentes índices de calidad, concentración de productos, texturas, entre otras. Aunque se han reportado diseños e implementaciones de sensores virtuales para solucionar problemas de procesos

puntuales, hasta el momento no se ha encontrado en la literatura consultada un trabajo con suficientes detalles para reproducir completamente los resultados y menos una metodología específica para el diseño de sensores inferenciales como los basados en redes neuronales artificiales, sistemas neuro – difusos y máquinas de vector de soporte

En diferentes publicaciones se encuentran múltiples diseños e implementaciones de sensores inferenciales basados en estas técnicas mostrando estimaciones muy cercanas a mediciones realizadas en laboratorios o con sensores de alto costo, sin embargo, en los trabajos reportados no se presentan criterios claros para el diseño de dichos sensores en aspectos como selección de la técnica de aprendizaje, selección del modelo, selección de técnicas involucradas en el preprocesamiento de los datos, entre otras; lo cual genera múltiples interrogantes, ya que cada uno de los aspectos antes mencionados comprende una variedad de técnicas que exhiben diferentes desempeños. Teniendo en cuenta esto, el presente trabajo se realiza con el fin de establecer una metodología donde se establezcan criterios claramente definidos para la toma de decisiones en el diseño de cada una de las etapas de un sensor inferencial teniendo como premisa el equilibrio entre velocidad de desarrollo y el desempeño del sensor virtual en cuanto a su predicción.

1.3 Antecedentes

Como se dijo anteriormente, en la actualidad la tendencia de los sensores inferenciales está en el uso de las técnicas de redes neuronales artificiales, sistemas de inferencia difusa, sistemas de inferencia neuro - difusa, máquinas de vector de soporte y la combinación de algunas de estas con el fin de obtener las ventajas más relevantes de cada una.

1.3.1 Redes Neuronales Artificiales (ANN)¹

Las redes neuronales artificiales, son una emulación simplificada del funcionamiento de las neuronas biológicas, las cuales toman información en sus entradas para procesar una respuesta en su salida (Alpaydin, 2010). Aprovechando esta característica fundamental las redes neuronales artificiales han sido ampliamente usadas en el desarrollo de sensores virtuales para la predicción de variables de difícil o impráctica medición, mediante una regresión a través de los datos de proceso.

Dentro de los diseños más frecuentes encontramos aplicaciones en la industria petroquímica, como en (Fortuna et al., 2005) donde, se presentan uno de sus resultados más relevantes, es un sensor inferencial utilizando una red neuronal artificial (ANN) del tipo

¹ ANN por Artificial Neural Network

perceptrón multicapa (MLP)², para la estimación del porcentaje de propileno, uno de los subproductos del petróleo; de lo presentado se destaca que el diseño se realizó a partir de datos obtenidos de una refinería en Sicilia, y que del conjunto inicial de datos tomo una parte para entrenar el sistema y lo restante para el proceso de validación. También expone el diseño de un sistema de inferencia difusa para una fácil transición entre el comportamiento del modelo en época de verano y de invierno. Sin embargo, no presenta en sus resultados ninguna prueba de desempeño frente a problemas en los datos de entrenamiento.

Son numerosos los autores que exponen resultados en el diseño de sensores inferenciales para estimar la concentración de subproductos del petróleo como C5, Butano C4, nafta ligera, entre otros, como en (Graziani, Pagano, & Xibilia, 2010) que diseñaron un sensor inferencial con ANN del tipo MLP entrenada mediante el algoritmo Levenberg-Mardquart a partir de datos de una refinería en Syracuse, Italia, y obteniendo un error muy bajo en la validación presentada en los resultados. Así mismo en la industria petroquímica es altamente deseable poder medir el ASTM 95%³, que es un índice de calidad estándar en la destilación del petróleo (Bhartiya & Whiteley, 2001) y son variados los trabajos que buscan diseñar un sensor virtual para solucionar esto, entre los trabajos destacados se encuentran los resultados de (Warne, Prasad, Rezvani, & Maguire, 2004) que propone una ANN del tipo MLP entrenada por el algoritmo de retro-propagación a partir de datos obtenidos desde simulación en el software SimDist⁴. Así mismo en (Bolf et al., 2008), se publica una solución al mismo problema de ASTM 95%, con una ANN del tipo MLP entrenada mediante el algoritmo de Levenberg-Mardquart a partir de datos de planta (DCS)⁵ de una refinería que no es mencionada. En ambos casos se obtuvieron resultados satisfactorios en la validación las cuales fueron realizadas con subconjuntos de los datos de entrenamiento.

Sin embargo, hay otras aplicaciones donde se implementan sensores inferenciales, una de ellas en la industria de fabricación de plástico donde hay índices de calidad que es deseable medirlos en tiempo real; uno de ellos es el índice de fusión el cual es tratado en el trabajo (Shi, Liu, & Sun, 2006), se propone un sensor inferencial mediante múltiples redes neuronales entrenadas con Levenberg-Mardquardt y datos obtenidos de planta; así mismo, en (Gonzaga, Meleiro, Kiang, & Maciel Filho, 2009) se reporta el diseño de un sensor inferencial para medir la viscosidad del plástico, aunque existen sensores para realizar esta medida estos presentan frecuente descalibración, y la solución presentada es mediante una ANN tipo MLP a partir de datos de una planta de botellas PET⁶ que no es especificada. En ambos casos se presentan resultados de validación a partir de un subconjunto de los datos de entrenamiento, sin embargo, en la estimación de la viscosidad

² MLP por Multilayer Perceptron

³ ASTM por American Society of the International Association for Testing and Materials

⁴ SimDist es un software comercial de simulación de torres de destilación

⁵ DCS por Distributed Control System

⁶ PET por Polyethylene Terephthalate

del plástico, el autor presenta un análisis del comportamiento del sensor en lazo cerrado lo cual da una mirada al desempeño del sensor inferencial diseñado dentro de un lazo de control.

Por último, existen aplicaciones que no están directamente ligadas a la fabricación de un producto sino al cumplimiento de normatividades ambientales en la industria entre las cuales se pueden enumerar las plantas de tratamiento de agua (Wang, 2010) y gases contaminantes lanzados al aire por causa de procesos industriales, como el diseño presentado en (Luigi Fortuna, Salvatore Graziani, 2005), con un diseño de 3 sensores virtuales basados en ANN del tipo MLP, RBF⁷ y un sistema neuro-difuso, para la estimación de la concentración de 3 contaminantes H₂S⁸, SO₂⁹, NO_x¹⁰, que son lanzados por las refinerías de petróleo; en su trabajo validan su diseño contra datos de planta obteniendo un error bajo en la predicción.

Por su parte en (Liukkonen, Hälikkä, Hiltunen, & Hiltunen, 2012) se reportan resultados con el diseño de un sensor inferencial para la estimación de NO_x en la chimenea de una caldera, utilizando ANN del tipo MLP entrenada por el algoritmo de retro-propagación y en (Liu & Yan, 2011) se presenta el sensor inferencial diseñado para estimar NO_x lanzado por un motor de combustión, a partir de datos obtenidos de un motor en condiciones de laboratorio; el diseño está basado en redes neuronales de regresión generalizadas y el entrenamiento lo realizaron con un algoritmo desarrollado dentro del marco del mismo trabajo. Para los dos casos la validación es realizada con un subconjunto de los datos iniciales que se tenían de cada una de las plantas.

1.3.2 Sistemas de Inferencia Neuro-Difusa (ANFIS)

Es muy frecuente encontrar la combinación de varias técnicas con el fin de obtener las mejores características en cuanto a tiempo de diseño, tiempo de entrenamiento e inclusive error de estimación por lo cual los sistemas neuro-difusos de inferencia (NFIS)¹¹ merecen ser mencionados como técnicas base para el diseño de sensores virtuales, aprovechando las características de las redes neuronales artificiales y los sistemas de inferencia difusa. Resultados utilizando esta técnica se encuentran en diferentes sectores como por ejemplo la domótica, donde (Jassar, Liao, & Zhao, 2009) presentan varias propuestas para la estimación de la temperatura promedio dentro de edificaciones, variable que según se indica en el trabajo es posible medirla con una gran cantidad de sensores lo que eleva costos, mientras que al implementar un sensor inferencial la cantidad de dispositivos físicos

⁷ Radial basis function

⁸ H₂S fórmula química de Sulfuro de Hidrógeno

⁹ SO₂ fórmula química del Óxido de Azufre

¹⁰ NO_x fórmula química para la familia de los óxidos de nitrógeno

¹¹ NFIS por Neuro – Fuzzy Inferential Systems

de medida será mucho menor. Teniendo en cuenta esto, el artículo propone varios diseños basados en los sistemas adaptativos de inferencia neuro-difusa (ANFIS)¹² y sistemas neuro-difusos retroalimentados concluyendo con análisis de error de estimación y validación de los sensores a partir de subconjuntos de los datos iniciales. Los datos fueron obtenidos de un laboratorio ubicado en Milán cuya temperatura es monitoreada por el proyecto EU CRAFT¹³.

Por otra parte, en (Xu & Liu, 2014), se reporta el desarrollo de un sensor inferencial basado en un sistema neuro-difuso dinámico, para la estimación del índice de fusión (MI)¹⁴ en el proceso de fabricación del plástico. En (Gao Lin, 2011) se propone un sensor virtual, para la estimación de la cantidad de éster presente en el plástico utilizado para la producción de botellas PET, basado en ANFIS a partir de datos de planta. En ambos casos los autores presentan análisis de error de estimación y validación de los modelos mediante el uso de un subconjunto de los datos iniciales.

1.3.3 Máquinas de vectores de soporte (SVM)

Debido al fondo teórico, basado en aprendizaje estadístico, las máquinas de vector de soporte (SVM) han venido cobrando importancia dentro de la comunidad de investigadores en aprendizaje de máquina (Kadlec et al., 2009) y por ende entre sus aplicaciones encontramos sensores inferenciales basados en SVM y algunas de sus variantes.

Se encuentran disponibles múltiples trabajos a cerca de sensores virtuales basados en SVM, aplicados a diferentes sectores industriales, como, por ejemplo, el caso de la industria de fabricación de plásticos se destaca con los resultados reportados por (Cheng & Liu, 2015), donde se propone un sensor inferencial basado en SVM, optimizado por mínimos cuadrados (LS-SVM)¹⁵ para la estimación del índice de fusión (MI) en la fabricación de plásticos. En el mismo sentido en (Chitrlekha & Shah, 2010b), se reporta resultados en el diseño de un sensor inferencial para la estimación de la misma variable aplicando SVM en el modelamiento de sistemas no lineales; en ambos casos los autores reportan el error de estimación conseguido y comparan con algunos métodos como redes neuronales, obteniendo como resultado tiempos de entrenamiento menores en los sensores basados en SVM.

Por otro lado, en la industria petroquímica, se destacan varias aplicaciones de las SVM por ejemplo, la estimación de la tasa de absorción de la gasolina reportada en (Sun et al., 2014), presenta una comparación entre sensores basados en SVM y LS-SVM usando como kernel¹⁶ una función de base radial (RBF) encontrando unos tiempos de

¹² ANFIS por Adaptive Neuro – Fuzzy Inference Systems

¹³ EU CRAFT es un proyecto para el manejo de información en proyectos de investigación

¹⁴ Melt Index

¹⁵ LS-SVM por Least Squares Support Vector Machines

¹⁶ Kernel es una función matemática que permite un cambio de espacio en un conjunto de datos

entrenamiento menores en el sensor basado en LS-SVM pero un mayor grado de generalización en el diseño basado en SVM. Así mismo, otras variables de la industria de derivados del petróleo como la concentración de C₄¹⁷ es de interés estimarla, caso que se abordó en (Behnasr & Jazayeri-Rad, 2015), diseñando un sensor inferencial basado en SVM con kernel RBF y realizando una comparación con un sistema ANFIS, el autor reportó resultados muy similares con la salvedad que la complejidad y tiempo de desarrollo del sensor basado en SVM es mucho menor que el caso con ANFIS.

En la producción de Kerosene un indicador de calidad del mismo es su punto de congelación, el cual es medido mediante pruebas de laboratorio, para la estimación de ésta variable (Yafen, Qi, Huijuan, & Ningsheng, 2006) propone dos sensores virtuales, uno basado en SVM y otro basado en LS-SVM utilizando como kernel una función de Gauss, a partir de datos de una planta refinadora de petróleo (DCS) y reportando errores de estimación muy bajos y tiempos de entrenamiento cortos con respecto a técnicas tradicionales como ANN o ANFIS.

Por último, en (Iswandy & König, 2011) se propone un sensor para la estimación de la concentración de NO_x presente en las emisiones de la caldera de una estación generadora de electricidad, lo cual constituye un indicador de que tan limpia es la energía producida. Para esto los autores proponen un sensor inferencial basado en LS-SVM a partir de datos medidos en el laboratorio de la generadora; se destaca por que presenta los cambios del diseño al usar diferentes kernel como RBF, función sigmoidea, función lineal y función polinomial, de esta manera se obtiene un panorama de la influencia del kernel en la estimación de la variable.

1.4 Objetivos

El objetivo general del presente proyecto es el de realizar un estudio comparativo de los sensores inferenciales basados en redes perceptrón multicapa (MLP - ANN), sistemas de inferencia neuro - difusa (ANFIS) y máquinas de vector de soporte (SVM), con el propósito de establecer una metodología de diseño e implementación de este tipo de sensores para ser aplicados en modelos de procesos continuos con orientación industrial. Para la consecución de esto se plantean las siguientes tareas:

Caracterizar los aspectos necesarios para el uso de las técnicas seleccionadas, como lo son arquitectura, algoritmos de entrenamiento, función de activación, kernel, etc., que constituyen la base para la implementación de los sensores inferenciales

¹⁷ C₄ es la fórmula química para un gas producto de la destilación del petróleo

Establecer tanto el proceso de naturaleza continua y carácter industrial como la manera en que se van a generar los conjuntos de datos que se utilizarán para el desarrollo de los sensores.

Diseñar e implementar sensores inferenciales basados en cada una de las técnicas ya definidas a partir de la identificación de las etapas requeridas y los conjuntos de datos del proceso establecido con anterioridad.

Desarrollar una comparación entre los sensores virtuales basados en cada técnica, a partir de la definición de criterios y métricas específicos para cuantificar el desempeño de los sensores inferenciales teniendo en cuenta aspectos como error de estimación, así mismo el efecto de fenómenos como la pérdida de datos y los datos fuera de tendencia.

Documentar todos los aspectos metodológicos encontrados en el proceso de diseño, implementación y pruebas desarrollados.

1.5 Contribuciones

En la investigación, descrita en el presente documento la principal contribución está constituida en la propuesta de una metodología de diseño e implementación de sensores inferenciales basados en redes neuronales artificiales, sistemas de inferencia neuro-difusa y máquinas de vectores de soporte.

Adicionalmente se presentan algunas contribuciones prácticas producto de la metodología aplicada para la consecución de los objetivos propuestos, tales como:

Implementación de técnicas de aprendizaje de máquina (ANN, ANFIS y SVM) mediante funciones parametrizadas para su uso general. Desarrolladas en lenguaje Octave/Matlab y disponibles en el repositorio <https://github.com/jorgecote/>.

Implementación de técnicas de preprocesamiento de datos (filtrado) de carácter univariado, mediante funciones parametrizadas para uso general. Desarrolladas en lenguaje Octave/Matlab. Disponible en el repositorio <https://github.com/jorgecote/Datafilter>

Análisis comparativo del desempeño de las técnicas de aprendizaje de máquina ANN, ANFIS y SVM frente a aspectos como su arquitectura, gasto computacional y características de datos de entrenamiento y validación.

Análisis comparativo del desempeño de las técnicas de aprendizaje de máquina ANN, ANFIS y SVM frente al uso de diferentes técnicas de preprocesamiento de datos de tipo univariado.

Análisis del desempeño de las técnicas de aprendizaje de máquina ANN, ANFIS y SVM frente a la selección de sistemas NARX o NARMAX como modelo de comportamiento de los datos de entrenamiento y los métodos para el cálculo adecuado de los auto-regresores.

Para efectos de validación de la metodología propuesta, desarrollo de código en lenguaje Octave/Matlab para la solución de las ecuaciones de balance de masa y equilibrio térmico de un modelo matemático para una columna de destilación binaria (agua-etanol) cuya ejecución genera datos típicos de proceso como presión, temperatura, flujo y concentración de productos. El código está disponible en el repositorio <https://github.com/jorgecote/DIstill-column>

Análisis de comportamiento del modelo de una columna de destilación y diseño de experimentos con la misma para la generación de datos que permitan un entrenamiento adecuado de técnicas de aprendizaje de máquina basadas en datos.

Análisis del comportamiento dinámico del modelo de la columna de destilación enfocado a las variables de proceso que afectan la concentración de producto (variable a predecir por el sensor inferencial) así como el diseño de experimentos para la generación de datos que permitan un entrenamiento y validación adecuados asociados a las técnicas de aprendizaje de máquina consideradas.

Diseño, implementación y validación de 3 sensores inferenciales basados en ANN, ANFIS y SVM a nivel de simulación buscando el mejor desempeño en lo referente a predicción de la variable y tiempo de implementación. Desarrollados en lenguaje Octave/Matlab.

1.6 Organización del documento

Con el fin de realizar la discusión de los diferentes aspectos que se trataran en el análisis de los sensores inferenciales y el desarrollo de la respectiva metodología el documento se plantea de la siguiente manera:

En el capítulo 2 se establece un comparativo entre las técnicas seleccionadas para el estudio de los sensores inferenciales, con respecto a diferentes aspectos que se consideran relevantes en el desempeño de las mismas, tales como, arquitectura, entrenamiento, convergencia y características de los conjuntos de datos. Teniendo en cuenta que cada una de las técnicas constituyen la base de cálculo del sensor inferencial y por consiguiente estas características de desempeño definirán el funcionamiento de los mismos.

En el capítulo 3, se plantea el modelo matemático de una columna de destilación binaria y una serie de experimentos llevando a la columna a trabajar en diferentes puntos de operación mediante simulación, con el objetivo de generar un conjunto de datos apropiado para el diseño de cada uno de los sensores inferenciales

En el capítulo 4 se plantean diferentes alternativas para tratar aspectos relacionados con los conjuntos de datos que pueden afectar el desempeño de los sensores inferenciales. Entre los aspectos a tratar están características de los datos (datos fuera de tendencia,

pérdida de datos y ruido) y dimensionalidad de los conjuntos de datos (selección o reducción de características).

En el capítulo 5 se discuten los aspectos referentes a la selección del modelo (NARX¹⁸ o NARMAX¹⁹) y posibles estrategias en la definición de la cantidad de auto-regresores necesarios para cada una de las técnicas ya establecidas, frente a un conjunto de datos del proceso de orientación industrial.

Por último, en el capítulo 6, se establece la metodología de diseño de sensores inferenciales, teniendo en cuenta los resultados obtenidos en los capítulos anteriores y los criterios necesarios para la toma de decisiones durante el proceso de diseño de un sensor inferencial basado en datos.

¹⁸ Por sus siglas en inglés Nonlinear Auto-regressive with exogenous inputs

¹⁹ Por sus siglas en inglés Nonlinear Auto regressive moving average with exogenous inputs

2. Técnicas de Aprendizaje de Máquina

En (Alpaydm, 2010) se define el aprendizaje de máquina como programas de computador para optimizar un criterio de desempeño usando muestras de datos o experiencia pasada, en otras palabras, el aprendizaje de máquina consiste en lograr que un sistema estime valores a partir de un patrón determinado que contiene el conjunto de datos inicial.

Con el desarrollo de la tecnología computacional en el anterior siglo, las técnicas de aprendizaje de máquina han tomado especial relevancia en la solución de problemas en diferentes campos como la robótica, biología, procesos químicos, entre otros. El aprendizaje de máquina está enfocado a solucionar dos problemas principales que son la clasificación; reconocimiento de diferentes subconjuntos dentro de un conjunto universo (Bishop, 2006), y por otra parte la regresión; que es la predicción o estimación de valores que obedecen unos patrones determinados (Bishop, 2006). Teniendo en cuenta que el objetivo del presente documento es el modelamiento de sistemas, el problema en el que se centraran todas las secciones siguientes es el de regresión.

En este capítulo se presenta una discusión, a cerca de los aspectos de diseño e implementación de las redes neuronales artificiales (ANN), máquinas de vectores de soporte (SVM) y sistemas de inferencia neuro-difusos (ANFIS) los cuales en la actualidad reportan gran cantidad de trabajos de aplicación entre los cuales encontramos implementaciones para visión de máquina, reconocimiento facial, sistemas control, sensores inferenciales, entre otros.

Tipos de entrenamiento

Las técnicas de aprendizaje de máquina requieren de un proceso donde se analizan los conjuntos de datos (base de conocimiento), denominado entrenamiento. Existen varios tipos de entrenamiento como el entrenamiento supervisado, no supervisado, aprendizaje profundo, etc., siendo los dos primeros los paradigmas clásicos del aprendizaje de máquina.

- **Entrenamiento Supervisado**

La técnica de aprendizaje procesa datos de entrenamiento como parejas entrada – salida con el fin de que pueda establecerse una relación entre unos patrones de entrada y unas salidas deseadas que corresponden a cada uno de los patrones (Bishop, 2006).

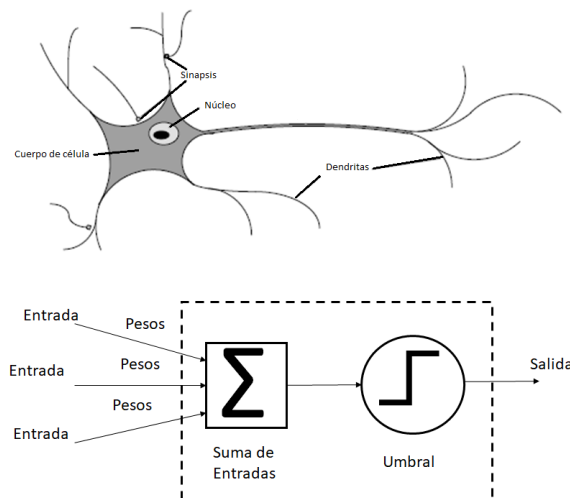
▪ **Entrenamiento No Supervisado**

La técnica de aprendizaje procesa conjuntos de datos donde se tienen únicamente unos patrones de entrada y la técnica en su proceso de análisis debe encontrar correlaciones en los datos para establecer el comportamiento de los mismos.

2.1 Redes Neuronales Artificiales (ANN)

Las redes neuronales artificiales denominadas en la literatura como ANN por sus siglas en inglés (Artificial Neural Networks), son un modelo simplificado de procesamiento de información, inspirado en la estructura y funcionamiento del cerebro humano. La figura 2.1 muestra una neurona con 3 partes fundamentales cuerpo, axón y dendritas, donde el cuerpo procesa señales recibidas desde otras células por medio de las dendritas y emite una respuesta por medio del axón. Dicha respuesta está directamente relacionada con las características del estímulo recibido (Babuška, 2009). En este sentido, las ANN emulan la interconexión de múltiples neuronas dentro del sistema nervioso.

Figura 2.1 Características de una neurona biológica y su modelo básico por McCulloch y Pitts. Tomada de (Babuška, 2009)



Desde el punto de vista matemático, una neurona se puede modelar como:

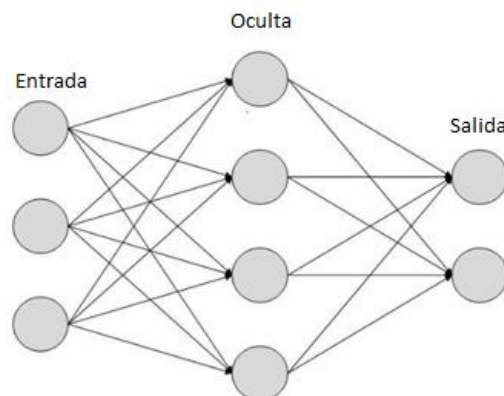
$$\hat{y} = g^{salida}(x^{salida}) = \sum_{n=1}^j x_n^{in} * w_{n,m}^{oculta} + x_0^{in}$$

Donde, $w_{n,m}^{oculta}$ corresponde a los pesos de las interconexiones entre las entradas y el perceptrón, y salida corresponde a la salida del modelo obtenido a partir de los datos de entrenamiento.

2.1.1 Arquitectura

Las ANN se configuran como la interconexión de múltiples neurodos (equivalente a neuronas), vinculados por sus entradas y salidas y organizados por capas. En la figura 2.2 se observa una ANN del tipo perceptrón multicapa²⁰ con 3 capas, una capa de entrada con 3 neurodos de entrada, una de salida con 2 neurodos y una capa oculta con 4 neurodos los cuales están interconectados en su totalidad. En la literatura se encuentran múltiples variaciones de esta arquitectura como, por ejemplo, redes neuronales artificiales donde los neurodos no están interconectados en su totalidad, con más de una capa oculta o inclusive con interconexiones entre capas que no son adyacentes.

Figura 2.2 Arquitectura red neuronal artificial perceptrón multicapa



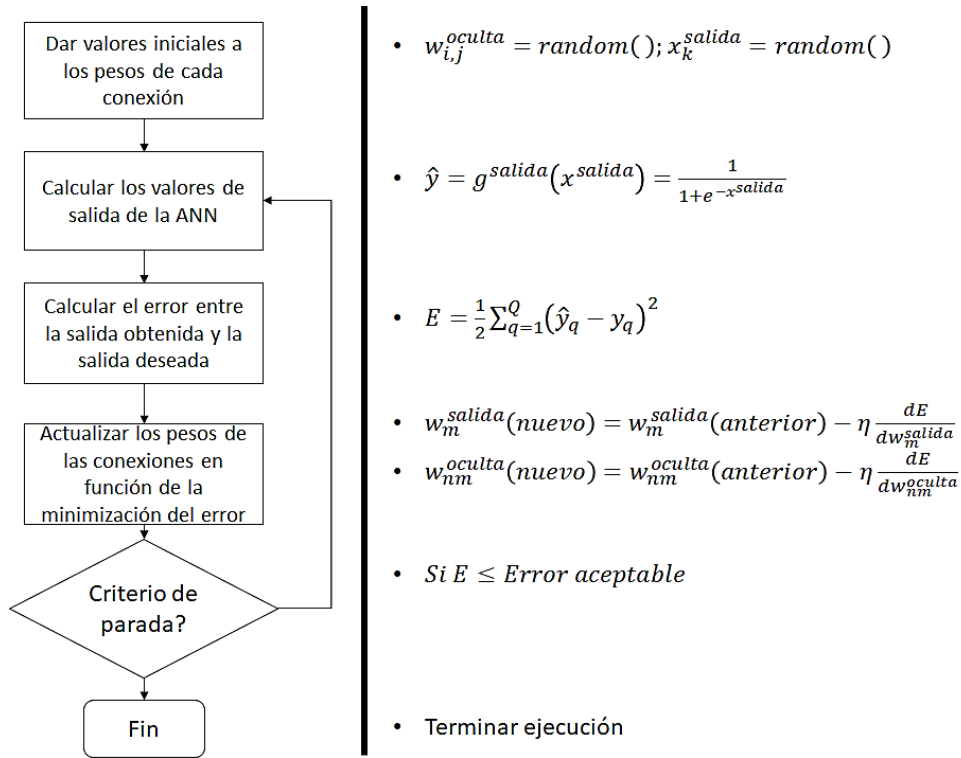
En el presente documento se considera la arquitectura que es la base para las otras arquitecturas, denominada MLP con una capa oculta y un neurodo en la capa de salida, la cuál es la forma más clásica de ANN y una de las más aplicadas en el desarrollo de sensores inferenciales para diferentes procesos industriales.

2.1.2 Entrenamiento

Debido a que las ANN son una técnica de aprendizaje supervisado, el proceso de entrenamiento se ilustra en la figura 2.3.

²⁰ MLP Multilayer perceptron

Figura 2.3 Diagrama de flujo del entrenamiento de ANN por retro-propagación



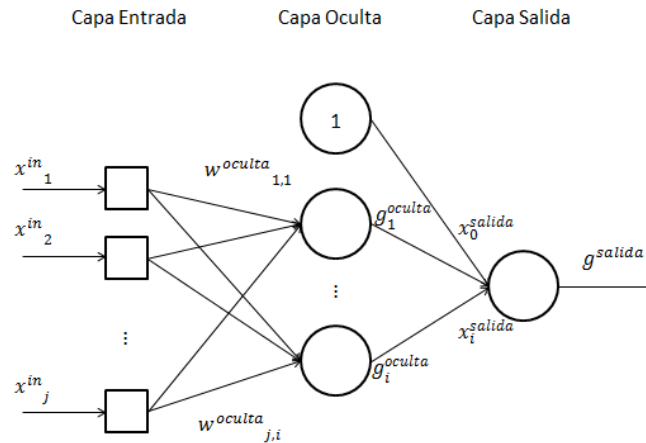
Para llevar a cabo este proceso existen múltiples algoritmos reportados en la literatura entre ellos encontramos el más clásico y conocido por su facilidad de implementación, algoritmo de retro-propagación.

Algoritmo de retro-propagación

El algoritmo de retro-propagación consta de dos fases, la primera conocida como propagación hacia adelante²¹ la cual consiste en calcular la salida de la ANN a partir del conjunto de datos de entrenamiento, la segunda denominada retro-propagación del error²² la cual consiste en la variación de los pesos de las interconexiones basándose en la variación del error.

²¹ Feedforward

²² Backward

Figura 2.4 Notación para la arquitectura de una red neuronal artificial tipo perceptrón multicapa

Propagación hacia adelante

Tal como se observa en la figura 2.4 la capa de entrada representa el número de dimensiones que tiene el conjunto de datos los cuales se multiplicaron por los pesos de cada una de las interconexiones.

Posteriormente se aplica la función de activación sigmoial de tipo exponencial que tiene la forma presentada en (2.1)

$$g(z) = \frac{1}{1+e^{-z}} \quad (2.1)$$

Por último, se ponderan los pesos de salida y se aplica nuevamente la función de activación de salida del mismo tipo

$$\hat{y} = g^{salida}(x^{salida}) = g^{salida}\left(\sum_{m=1}^i g_m^{oculta}\left(\sum_{n=1}^j x_n^{in} * w_{n,m}^{oculta} + x_0^{in}\right) * w_m^{salida} + x_0^{salida}\right) \quad (2.2)$$

Donde j es el número de dimensiones del conjunto de datos y m representa los neurodos de la capa oculta y $m = [0, 1, 2, \dots, i]$. Adicionalmente los super-índices representan la capa a la cual pertenece cada miembro de la fórmula.

Obteniendo de esta manera la salida correspondiente a cada una de las muestras que componen el conjunto de datos de entrenamiento. En algunos casos en la capa de salida se utiliza una función lineal en cuyo caso la última parte de la fórmula (2.2) no se incluye (J.-S. R. Jang, Sun, & Mizutani, 1997). Para más detalles en el desarrollo matemático ver (Alpaydın, 2010; Bishop, 2006).

Retro-Propagación

La segunda fase inicia con el cálculo del error de predicción y su variación con respecto a los pesos de cada una de las interconexiones entre neurodos por medio de (2.3):

$$E = \frac{1}{2} \sum_{q=1}^Q (\hat{y}_q - y_q)^2 \quad (2.3)$$

Siguiendo la notación Q es la cantidad de muestras o patrones del conjunto de datos de entrenamiento.

El objetivo es minimizar el error E para de esta manera garantizar que la ANN responde siguiendo el patrón del conjunto de datos. Para esto es necesario utilizar algún método de optimización entre los que podemos encontrar gradiente conjugado, métodos newtonianos o gradiente descendiente que por su simplicidad de implementación y por ser uno de los más aplicados para el entrenamiento de ANN se enuncia a continuación.

El método de gradiente descendiente, el cual se presenta a continuación en su forma más general:

$$\theta_{new} = \theta_{old} - \eta \frac{dE}{d\theta} \quad (2.4)$$

Donde Θ es el parámetro que se está variando para lograr el objetivo de optimización, en otras palabras, la minimización del error se logra con variaciones del parámetro Θ buscando que el error decrezca progresivamente (Alpaydin, 2010).

En el caso de la arquitectura que se considera en el presente documento, las derivadas del error que corresponde calcular se presentan en (2.5) y (2.6)

$$\frac{dE}{dw_m^{salida}} = \frac{dE}{d\hat{y}} \frac{d\hat{y}}{dw_m^{salida}} \quad (2.5)$$

$$\frac{dE}{dw_{nm}^{oculta}} = \frac{dE}{d\hat{y}} \frac{d\hat{y}}{dg_m^{oculta}} \frac{dg_m^{oculta}}{dw_{nm}^{oculta}} \quad (2.6)$$

Por último, los pesos de las interconexiones de capa oculta y salida se actualizan aplicando la definición de gradiente descendiente enunciada en (2.4)

$$w_m^{salida}(nuevo) = w_m^{salida}(anterior) - \eta \frac{dE}{dw_m^{salida}} \quad (2.7)$$

$$w_{nm}^{oculta}(nuevo) = w_{nm}^{oculta}(anterior) - \eta \frac{dE}{dw_{nm}^{oculta}} \quad (2.8)$$

Los cálculos anteriormente presentados deben ser realizados para cada uno de los datos del conjunto de datos de entrenamiento y repetir el procedimiento hasta alcanzar el error máximo admisible como criterio para detener el entrenamiento (Bishop, 2006).

El algoritmo para el entrenamiento de ANN en arquitectura MLP se resume de la siguiente manera:

-
1. Inicializar los parámetros de la ANN como w (pesos en conexiones entre neurodos), m (número de neurodos en la/s capa/s oculta/s), $iter$ (número de iteraciones del entrenamiento)
 2. Calcular el paso hacia adelante para una muestra de los datos
 3. Calcular retro-propagación para la misma muestra
 4. Actualizar los pesos w
 5. Repetir desde el paso 2 hasta completar todo el conjunto de datos de entrenamiento
 6. Repetir el paso 5 hasta completar el número de iteraciones ($iter$) propuestas para alcanzar el ajuste de los datos
-

Algoritmo 2.1 Entrenamiento ANN por retro-propagación con gradiente descendente

2.2 Sistemas Neuro-Difusos de inferencia (ANFIS)

Propuestos por (J. S. R. Jang, 1993) los sistemas ANFIS²³ buscan reunir las mejores características de los sistemas de inferencia difusa²⁴ y las posibilidades de entrenamiento de las redes neuronales artificiales. La idea central de estos sistemas es modelar un sistema difuso el cual representa el conocimiento de una manera interpretable casi en un lenguaje natural y dar capacidad de aprendizaje por medio del ajuste de los parámetros de las funciones lingüísticas a través de una arquitectura similar a una red neuronal artificial

- **Sistemas de inferencia difusa (FIS)**

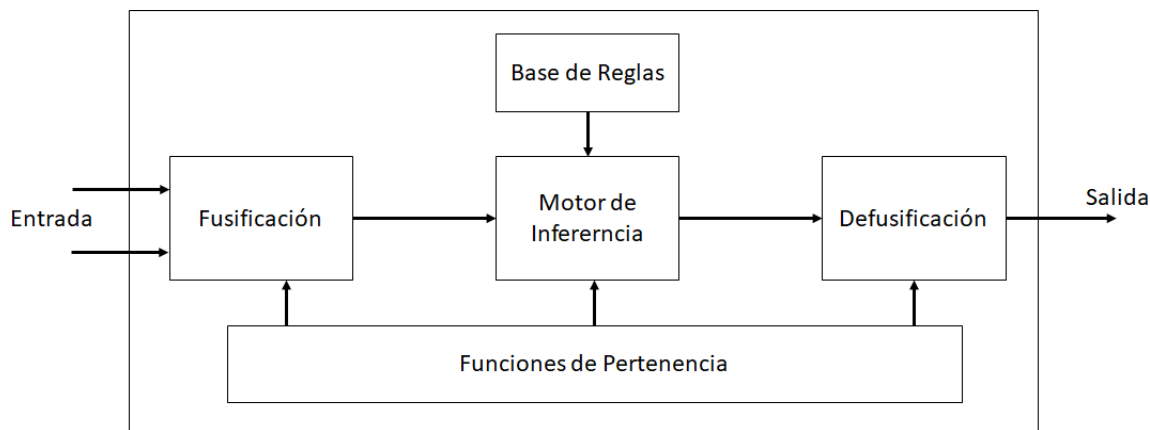
Los sistemas de inferencia difusa son frecuentemente utilizados para emular el razonamiento humano y sus modos imprecisos en cuanto a la habilidad de tomar decisiones en ambientes de incertidumbre e imprecisión (J. S. R. Jang, 1993). Esto se logra por medio del uso de reglas del tipo **IF A THEN B** donde A y B son etiquetas lingüísticas que definen un conjunto difuso caracterizadas por funciones de pertenencia apropiadas (Babuška, 2009).

En general, el proceso de razonamiento difuso se compone de las siguientes etapas:

²³ ANFIS por sus siglas en inglés Adaptive Neuro-Fuzzy Inference System

²⁴ FIS por sus siglas en inglés Fuzzy Inference System

Figura 2.5 Estructura de un sistema difuso. Tomado de (Funsten, 2015; J. S. R. Jang, 1993)



1. Fusificación: En esta etapa se transforman los datos de entrada en grados de pertenencia a funciones definidas previamente que corresponden a cada una de las etiquetas lingüísticas en el antecedente de cada regla.
2. Inferencia: Se combinan los valores de las funciones de pertenencia por medio de algún operador de conjuntos difusos y la base de reglas para obtener un conjunto difuso en la salida (Babuška, 2009).
3. Defusificación: En esta etapa se transforman los valores del conjunto difuso obtenido en valores limpios. Para esto se dispone de varios métodos de defusificación (Babuška, 2009).

Existen varios tipos de inferencia propuestos en la literatura, los cuales presentan diferencias en el tipo de razonamiento difuso utilizado y la forma que tenga el consecuente de la base de reglas. Para el presente documento se considera las bases de reglas tipo 3 las cuales incluyen combinaciones lineales para realizar la defusificación (J. S. R. Jang, 1993)

Desde el punto de vista matemático los sistemas ANFIS obedecen al siguiente modelo:

$$output = \sum \frac{w_i \sum x_i r_i}{\sum w_i} \quad (2.9)$$

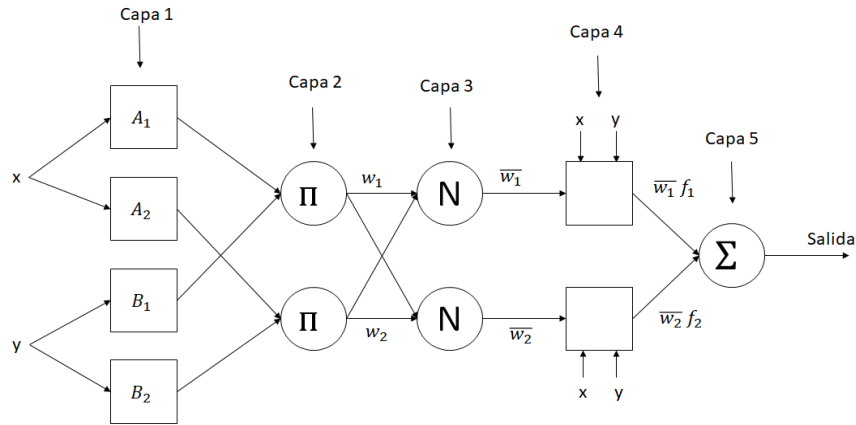
Donde w_i corresponde a la inferencia obtenida en cada regla, x_i las entradas del sistema y r_i son los parámetros de las funciones takagi-sugeno utilizadas para la defusificación.

2.2.1 Arquitectura

En (J. S. R. Jang, 1993) se propone una arquitectura basada en las redes adaptadas que son la generalización de las ANN, donde el objetivo es minimizar el error a través de la variación de los parámetros que definen las funciones de pertenencia en el antecedente de las reglas y los parámetros en el consecuente, para el caso de los sistemas tipo 2 y 3.

En la figura 2.6 se observa la arquitectura propuesta por (J. S. R. Jang, 1993) que está compuesta por 5 capas en las cuales están distribuidas cada una de las etapas del FIS.

Figura 2.6 Arquitectura sistema ANFIS (J. S. R. Jang, 1993).



En la capa 1 se encuentran las funciones de pertenencia para cada una de las etiquetas lingüísticas y que tendrán como entrada los datos que se tienen para realizar el modelamiento. En la capa 2 se calcula el grado de pertenencia de cada regla, por medio de cualquiera de los métodos conocidos para obtener el operador lógico AND, típicamente se utiliza el producto, sin embargo, es posible utilizar cualquiera de los operadores T-norm desarrollados en la lógica difusa. En la capa 3 se normaliza el peso de cada regla dividiendo por la suma del grado de pertenencia de todas las reglas. En la capa 4 se calcula el aporte de cada uno de los consecuentes en la salida. Por último, en la capa 5 se obtiene la salida general sumando cada uno de los aportes de los consecuentes de cada regla.

En la literatura se reportan modificaciones de sistemas ANFIS con capas adicionales o conexiones adicionales entre capas, así como FIS tipo 1, tipo 2 y tipo 3 en su estructura. Debido a que la mayor parte de las aplicaciones utilizan la estructura clásica reportada por (J. S. R. Jang, 1993), en las siguientes subsecciones se detallará el procedimiento y algoritmo de entrenamiento para un sistema ANFIS de 5 capas como el de la figura 2.6, con FIS tipo 3, n entradas, m reglas y una salida.

2.2.2 Entrenamiento

ANFIS se entrena de manera supervisada al igual que las ANN, por lo tanto, es necesario contar con datos de entrada y las salidas deseadas para obtener un modelo de buen desempeño. En (J. S. R. Jang, 1993) se plantea 3 maneras de entrenar un sistema ANFIS los cuales se especifican en la tabla 2.1.

Tabla 2.1 Métodos de entrenamiento ANFIS

Antecedente	Consecuente	Entrenamiento
Fijo	Variable	Feedforward ²⁵
Variable	Fijo	Backpropagation ²⁶
Variable	Variable	Híbrido

En el entrenamiento híbrido, se debe actualizar parámetros tanto en su etapa de paso hacia adelante como en su etapa de retro-propagación. A continuación, se presenta una breve descripción del procedimiento de entrenamiento híbrido y los algoritmos para el cálculo de los parámetros correspondientes reportados en (J. S. R. Jang, 1993).

Como se dijo en subsecciones anteriores, los sistemas ANFIS se pueden resumir en sistemas FIS dispuestos en una red adaptada con el fin de obtener parámetros adecuados en el antecedente y consecuente de la base de reglas. Para lograr este objetivo es necesario variar los parámetros anteriormente mencionados por medio de la minimización del error de dicha red.

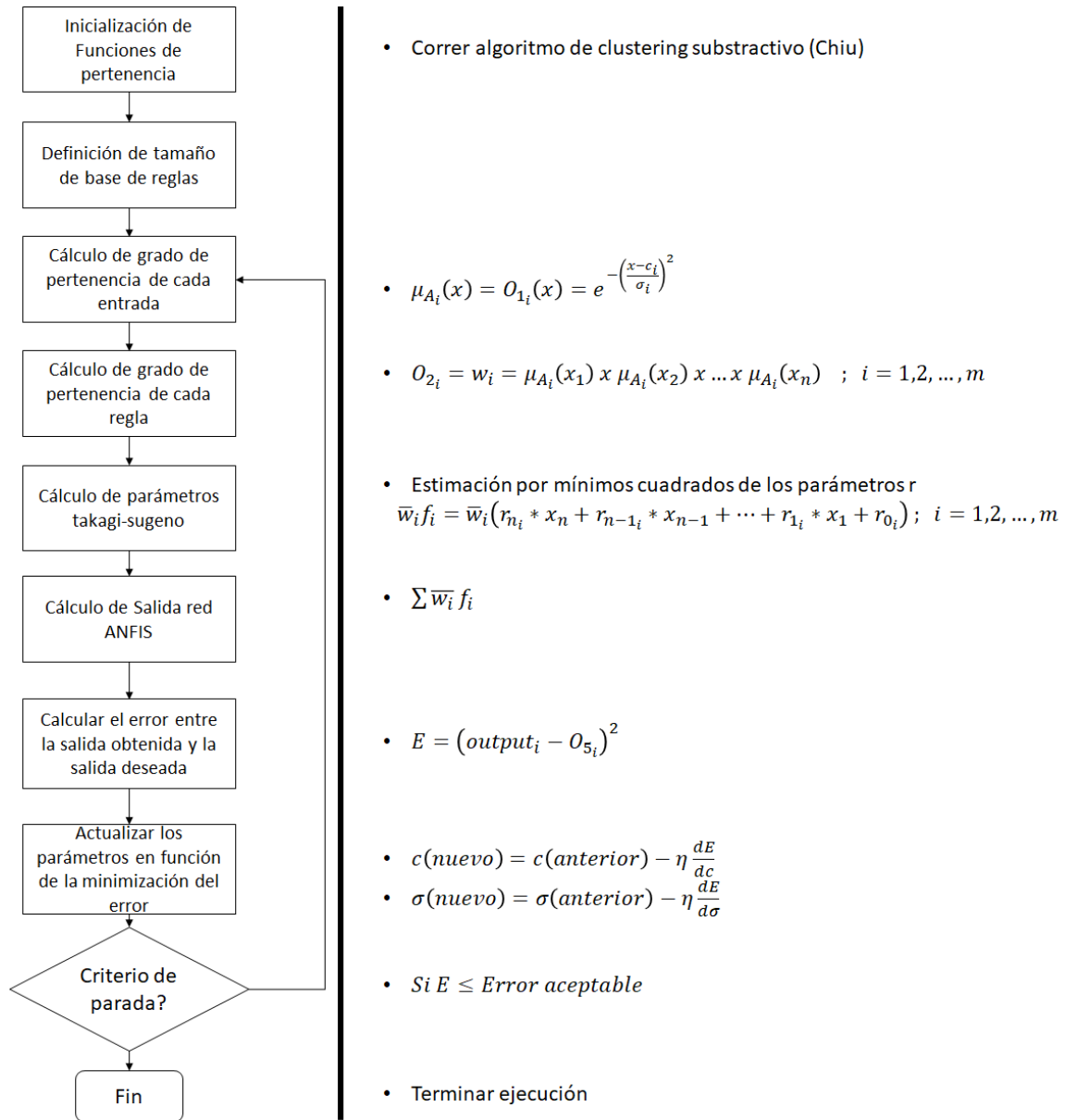
Para esto, es necesario el denominado entrenamiento híbrido, el cuál en el paso hacia adelante calcula los parámetros iniciales del consecuente, a partir del conjunto de datos de entrenamiento por medio de cualquier algoritmo de regresión lineal; típicamente se utiliza estimación por mínimos cuadrados. En contraste, la retro-propagación se realiza con el método de gradiente descendente para calcular los parámetros propios de las funciones de pertenencia en el antecedente de la base de reglas (J.-S. R. Jang et al., 1997).

De manera general, el entrenamiento híbrido se ilustra en la figura 2.7.

²⁵ Paso hacia adelante, típicamente se calculan los parámetros por el método de mínimos cuadrados

²⁶ Retro-propagación del error típicamente se aplica combinada con el método de gradiente descendente

Figura 2.7 Diagrama de Flujo para entrenamiento de red ANFIS por algoritmo de retro-propagación híbrido



En los siguientes párrafos se especifica el cálculo de cada una de las etapas del entrenamiento y los algoritmos a utilizar para el cálculo de los parámetros necesarios.

Paso hacia Adelante

▪ Inicialización y tamaño de base de reglas

A diferencia de las ANN, en los sistemas ANFIS la inicialización de parámetros no se realiza de manera aleatoria, por el contrario, es necesario calcular los valores iniciales de las funciones de pertenencia y el tamaño de la base de reglas a partir de los datos de entrenamiento. Para esto, en (J. S. R. Jang, 1993) se sugiere utilizar el algoritmo de clustering denominado "grid partition" para conjuntos de datos pequeños por su fácil aplicación. En contraste, frente a conjuntos de datos medianos y grandes se sugiere el uso del algoritmo de clustering por sustracción propuesto en (Chiu, 1994) el cual calcula una cantidad de reglas apropiada para el entrenamiento de los sistemas ANFIS, aunque su implementación conlleva un esfuerzo mayor.

El algoritmo de clustering substractivo reportado por (Chiu, 1994) consiste en asumir cada datos como un centro de cluster y a través de una métrica denominada potencial ($P_i = \sum_{j=1}^n e^{-\alpha |x_i - x_j|^2}$; $\alpha = \frac{4}{r_a^2}$) se establecen cuáles de los datos son aceptados como centro de cluster y cuales se rechazan; este procedimiento se realiza hasta examinar todos los datos que componen el conjunto. En el Anexo 8.4 se incluye una implementación en código Octave/Matlab de algoritmo de clustering substractivo propuesto por (Chiu, 1994)

El algoritmo de clustering substractivo solamente depende de los datos de entrenamiento y su resultado entrega la información suficiente para inicializar el FIS, interpretando los centros de cluster como los centros para cada una de las funciones de pertenencia y la cantidad de clusters como el tamaño de la base de reglas. En la figura 2.8 se presenta la interpretación gráfica de estos parámetros

Adicionalmente en el ANFIS toolbox de Matlab, Chiu implementa el algoritmo de clustering substractivo e incluye la siguiente fórmula para calcular los parámetros sigma σ de cada función de pertenencia

$$\sigma = \frac{r_a(\max(input) - \min(input))}{\sqrt{8}} \quad (2.10)$$

▪ Fusificación

Corresponde a la capa 1 de la arquitectura presentada en la figura 2.6, donde cada nodo es una etiqueta lingüística caracterizada por una función que indica el grado de pertenencia que tiene la entrada x o y a cada conjunto.

Las funciones de pertenencia más utilizadas para los sistemas ANFIS tienen forma de campana de Gauss como las siguientes:

$$\mu_{A_i}(x) = O_{1_i}(x) = \frac{1}{1 + \left(\left(\frac{x - c_i}{a_i}\right)^2\right)^b} \quad (2.11)$$

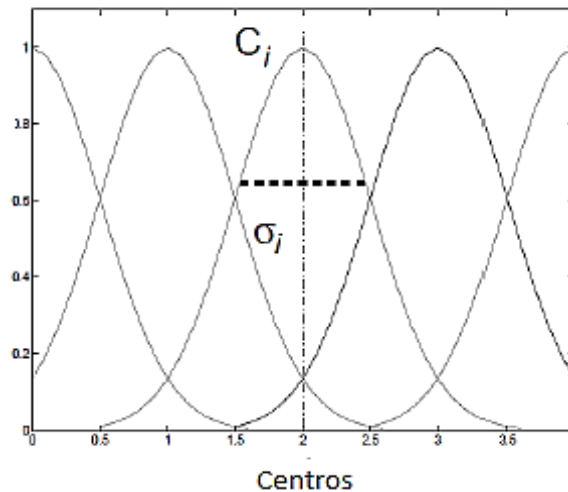
Donde a define el ancho de la función de pertenencia, b define el cruce con otras funciones de pertenencia y c define la ubicación del centro de la función de pertenencia.

$$\mu_{A_i}(x) = O_{1_i}(x) = e^{-\left(\frac{x - c_i}{\sigma_i}\right)^2} \quad (2.12)$$

Donde c corresponde al centro de la función de pertenencia y σ es el ancho de la campana de Gauss

Por utilizar menos parámetros, los cálculos que se realizarán más adelante serán implementados con la forma exponencial (2.12). En la figura 2.8 se presenta la interpretación grafica de los parámetros.

Figura 2.8 Interpretación gráfica de los parámetros de las funciones de pertenencia (Hamdan, 2013)



▪ Inferencia

En los sistemas ANFIS la inferencia se realiza a partir de la aplicación del operador T-norm, que para este caso se utiliza el producto. Sin embargo, también se pueden utilizar operadores como max o min. Teniendo en cuenta lo anteriormente mencionado, el producto se calcula de la siguiente manera:

$$O_{2_i} = w_i = \mu_{A_i}(x_1) \times \mu_{A_i}(x_2) \times \dots \times \mu_{A_i}(x_n) \quad ; \quad i = 1, 2, \dots, m \quad (2.13)$$

Donde m es el número de funciones de pertenencia y n es la cantidad de entradas (características) del sistema.

▪ **Defusificación**

Corresponde a la capa 3, 4 y 5, donde se pretende llevar todos los valores obtenidos en la capa anterior a un valor en el espacio original a través de la aplicación de las funciones takagi-sugeno. En (2.14) se realiza la normalización de la inferencia para ponderar el cálculo de los valores defusificados

$$O_{3_i} = \bar{w}_i = \frac{w_i}{w_1 + w_2 + \dots + w_i} ; i = 1, 2, \dots, m \quad (2.14)$$

Donde m es la cantidad de reglas definidas por el algoritmo de clustering substractivo

En (2.15) se calcula la función takagi sugeno que corresponde al aporte de cada regla en la salida

$$O_{4_i} = \bar{w}_i f_i = \bar{w}_i (r_{n_i} * x_n + r_{n-1_i} * x_{n-1} + \dots + r_{1_i} * x_1 + r_{0_i}) ; i = 1, 2, \dots, m \quad (2.15)$$

Donde n es la cantidad de características del conjunto de datos

Por último, se suman todas las funciones takagi sugeno ponderadas en (2.16)

$$O_{5_i} = \sum \bar{w}_i f_i ; i = 1, 2, \dots, m \quad (2.16)$$

Para el cálculo de los parámetros r_i , (Hamdan, 2013) recomienda utilizar el método de los mínimos cuadrados²⁷ el cual es utilizado para solucionar problemas del tipo $||AX - B||^2$. Para el caso de ANFIS el término A corresponde a la sumatoria de todos los términos de la capa 4, B son las salidas deseadas del conjunto de datos de entrenamiento y X es el conjunto de parámetros que se desean calcular, siguiendo con la notación de (2.15) tenemos:

$$||A * X - output||^2$$

Donde,

$$A = [\bar{w}_n x_n * \bar{w}_{n-1} x_{n-1} * \dots * \bar{w}_0 x_0] \quad output = \begin{bmatrix} output_1 \\ output_2 \\ \vdots \\ output_n \end{bmatrix} \quad X = \begin{bmatrix} r_n \\ r_{n-1} \\ \vdots \\ r_1 \\ r_0 \end{bmatrix}$$

El cálculo se realiza por medio de la estimación de mínimos cuadrados, debido a que se trata de una combinación lineal. Para casos en los que la matriz A resulte singular se

²⁷ LSE por sus siglas en inglés Least Squares Estimation

sugiere el método iterativo propuesto en (J. S. R. Jang, 1993) . En el Anexo 8.3 se presenta la aplicación de la estimación por mínimos cuadrados (LSE) de una manera más detallada.

Retro-Propagación

Al igual que con las ANN en la anterior sección, para los sistemas ANFIS en la fase de retro-propagación el objetivo de optimización es la minimización del error, para lograr que el sistema ANFIS responda de acuerdo al conjunto de datos de entrenamiento. Por lo tanto, se considera el error de predicción como:

$$E = (\text{output}_i - o_{5_i})^2 \quad (2.17)$$

Donde se compara la salida del sistema ANFIS o_{5_i} y las salidas deseadas del conjunto de datos de entrenamiento output_i . Donde i pertenece a $[1, 2, \dots, p]$, con p como la cantidad de patrones en el conjunto de entrenamiento.

Para lograr el objetivo de optimización se utiliza nuevamente el método de gradiente descendente de la fórmula (2.7), aplicado a los parámetros c_i y σ . En el mismo sentido, por tratarse de una red adaptada, es posible calcular las derivadas del error aplicando regla de la cadena, utilizando las derivadas en cada uno de los nodos.

$$\frac{dE}{dc} = \frac{dE}{do_5} \frac{do_5}{do_4} \frac{do_4}{do_3} \frac{do_3}{do_2} \frac{do_2}{do_1} \frac{do_1}{dc} \quad (2.18)$$

$$\frac{dE}{d\sigma} = \frac{dE}{do_5} \frac{do_5}{do_4} \frac{do_4}{do_3} \frac{do_3}{do_2} \frac{do_2}{do_1} \frac{do_1}{d\sigma} \quad (2.19)$$

Como se dijo previamente, el desarrollo se realiza para la arquitectura propuesta por (J. S. R. Jang, 1993) que se muestra en la figura 2.6. El desarrollo de las derivadas es presentado en (Hamdan, 2013).

Por último, se procede con la actualización de los parámetros aplicando el método del gradiente descendente de la fórmula (2.7) (Hamdan, 2013)

$$c(\text{nuevo}) = c(\text{anterior}) - \eta \frac{dE}{dc} \quad (2.20)$$

$$\sigma(\text{nuevo}) = \sigma(\text{anterior}) - \eta \frac{dE}{d\sigma} \quad (2.21)$$

Los cálculos anteriormente presentados deben ser realizados para cada uno de los datos del conjunto de datos de entrenamiento y repetir el procedimiento hasta alcanzar el error máximo admisible como criterio para detener el entrenamiento.

El algoritmo para el entrenamiento de sistemas ANFIS se resume de la siguiente manera:

1. Inicializar los parámetros m (tamaño base de reglas), c (centros de las funciones de pertenencia) y σ (ancho de las funciones de pertenencia) en el sistema ANFIS por medio del algoritmo de clustering substractivo
 2. Calcular el paso hacia adelante para una muestra de los datos, utilizando el método de estimación de mínimos cuadrados para calcular los parámetros iniciales del consecuente
 3. Calcular retro-propagación para la misma muestra
 4. Actualizar los parámetros c y σ
 5. Repetir desde el paso 2 hasta completar todo el conjunto de datos de entrenamiento
 6. Repetir el paso 5 hasta completar el número de iteraciones ($iter$) propuestas para alcanzar el ajuste de los datos
-

Algoritmo 2.2 Entrenamiento ANFIS por retro-propagación con gradiente descendente

2.3 Máquinas de vectores de soporte (SVM)

A diferencia de las técnicas descritas anteriormente, cuya arquitectura y método de entrenamiento es aplicable tanto para el problema de clasificación como para el problema de regresión, las máquinas de vectores de soporte²⁸ realizan un enfoque ligeramente diferente para dar solución a cada uno de estos problemas. En el presente documento se describirán las máquinas de vectores de soporte para regresión²⁹ y los algoritmos de entrenamiento correspondientes, en consecuencia, no se abordará la problemática de clasificación resuelta por medio de SVM.

En general, las máquinas de vectores de soporte son algoritmos no lineales de generalización desarrollados inicialmente en (Kovalev, Kotlyar, Porfirev, & Kalinkina, 2016; Vapnik & Lerner, 1963), hasta llegar a su forma actual, han sido ampliamente desarrolladas por Vapnik y su equipo en los laboratorios de AT&T Bell; debido a que se desarrollaron dentro de un contexto industrial las máquinas de vectores de soporte o SVM están

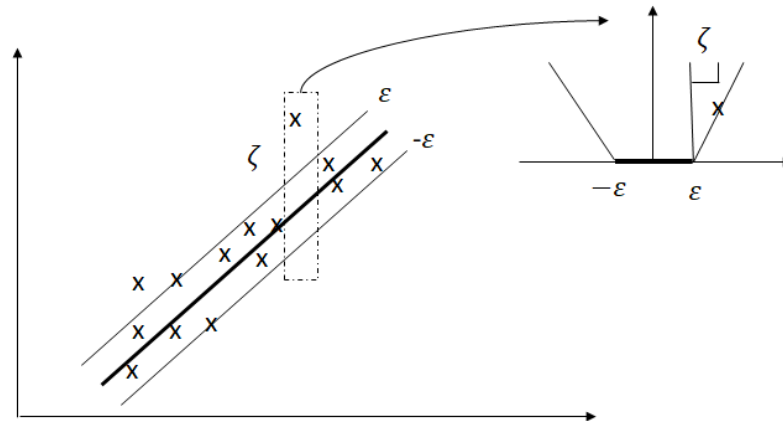
²⁸ SVM por sus siglas en inglés Support Vector Machines

²⁹ SVR por sus siglas en inglés Support vector regression

altamente basadas en datos y se encuentran aplicaciones para dar solución a problemas en entornos industriales.

La regresión por medio de SVM, presenta un enfoque distinto a las redes adaptadas como los sistemas ANFIS o las ANN, con las cuales se pretende aproximar una función $f(x)$ por medio de combinaciones lineales en diferentes regiones; en contraste, las SVM pretenden aproximar una función $f(x)$ con una desviación máxima denominada ε con respecto al conjunto de datos de entrenamiento, la cual bajo ningún motivo puede ser excedida, en otras palabras, el modelo se obtiene encerrando en un “tubo” los datos de entrenamiento tal como se ilustra en la figura 2.9 (A. Smola et al., 1996).

Figura 2.9 Interpretación grafica de la regresión lineal por SVM (A. Smola et al., 1996).



2.3.1 Regresión Lineal

Inicialmente se describirá la solución a una regresión lineal por medio de SVM, en la siguiente subsección se extenderá el concepto a las regresiones no lineales.

Por tratarse de una regresión lineal, se asume que puede ser aproximada por la siguiente función:

$$f(x) = \sum w * X + b ; \text{ con } w \in \mathcal{X} \text{ y } b \in \mathbb{R} \quad (2.22)$$

Debido a que w y X son matrices el término $\sum w * X$ puede ser interpretado como el producto punto entre w y X por lo tanto en lo que resta del desarrollo de las SVM esta operación se denotará como $\langle w, x \rangle$. En (2.22) w es denominado margen y corresponde a la proyección ortogonal de cada punto en el espacio.

Una de las características que se busca garantizar con las SVM es que la función sea muy plana, esto se interpreta como garantizar que w sea lo más pequeño posible. Una manera de lograr esto es minimizando la norma de w así (A. Smola et al., 1996):

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 \\ \text{con} \quad & \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x \rangle + b - y_i \leq \varepsilon \end{cases} \end{aligned} \quad (2.23)$$

En (2.23) se asume que existe un f que puede aproximar todos los patrones del conjunto de datos con una desviación máxima ε , sin embargo, no siempre se da esto por lo tanto se incluye el concepto de "margen suave" reportado por (A. Smola et al., 1996), donde lo que se pretende es agregar unas variables de holgura ξ y ξ^* con el fin de penalizar la función f para aquellos patrones que quedan por fuera del margen de desviación ε . En consecuencia, el objetivo de optimización se escribe como:

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ \text{con} \quad & \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (2.24)$$

El término C maneja la tolerancia de la función f para permitir patrones del conjunto de datos por fuera del margen w . La anterior definición se traduce en la denominada función de pérdida ε -insensitiva. Los conceptos anteriormente enunciados tienen su interpretación gráfica en la figura 2.9.

Para la solución del problema formulado en (2.24) se lleva a su forma dual por medio del uso de multiplicadores de Lagrange y la aplicación de las restricciones, de lo cual se obtiene el siguiente problema de optimización

$$\begin{aligned} \max \quad & \left\{ \begin{aligned} & -\frac{1}{2} \sum_{j,i=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \\ & \text{con} \quad \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \text{ y } \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \right. \end{aligned} \quad (2.25)$$

Por lo tanto, es posible reformular la solución al problema de regresión en términos de los multiplicadores de Lagrange y los patrones de entrenamiento x_i

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) x_i \quad (2.26)$$

$$\text{Entonces} \quad f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \quad (2.27)$$

De (2.27) se observa que la complejidad de la función f , no depende de la dimensionalidad del conjunto de datos x_i , sino de la cantidad de α_i, α_i^* que son diferentes de 0.

Por último, es necesario calcular el término b de f , el cual es análogo al término de bias presente en las ANN o al término r_0 presente en el consecuente de los sistemas ANFIS.

Las condiciones de Karush-Kuhn-Tucker (Karush, 1939, Kuhn and Tucker, 1951) establecen que en el punto de solución el producto entre las variables duales y las condiciones es 0.

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) &= 0 \\ \alpha_i^*(\varepsilon + \xi_i^* - y_i + \langle w, x_i \rangle + b) &= 0 \end{aligned} \quad (2.28)$$

$$\begin{aligned} (C - \alpha_i)\xi_i &= 0 \\ (C - \alpha_i^*)\xi_i^* &= 0 \end{aligned} \quad (2.29)$$

De (2.58) y (2.59) es posible concluir lo siguiente:

Si la desviación de $f(x)$ supera ε los multiplicadores de Lagrange son diferentes de 0

$$\text{Si } |f(x_i) - y_i| \geq \varepsilon \quad \text{entonces} \quad \alpha_i, \alpha_i^* \neq 0$$

Si la desviación de $f(x)$ no supera ε los multiplicadores de Lagrange son iguales a 0

$$\text{Si } |f(x_i) - y_i| < \varepsilon \quad \text{entonces} \quad \alpha_i, \alpha_i^* = 0$$

De lo anterior el nombre máquinas de vectores de soporte, ya que todos aquellos patrones cuyos multiplicadores sean diferentes de 0 son denominados vectores de soporte, debido a que son los que definirán en última instancia la función $f(x)$.

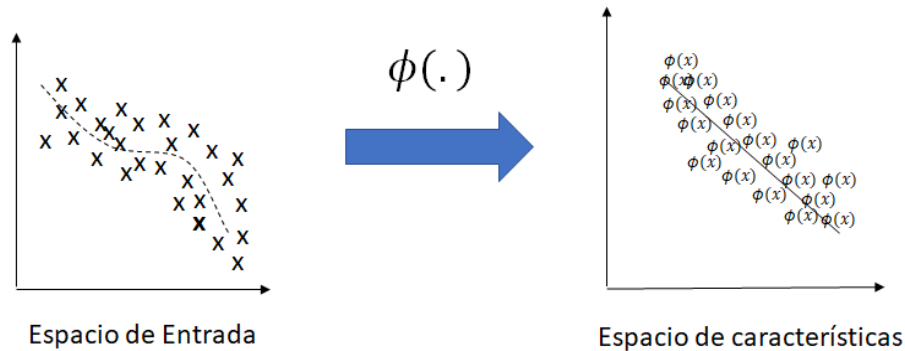
Hasta aquí se describe el planteamiento para solucionar el problema de regresión lineal, sin embargo, es necesario solucionar el problema de optimización de (2.25) para obtener los parámetros correspondientes. Para esto, existen múltiples herramientas matemáticas, tales como, métodos numéricos para problemas de variable dual y algoritmos analíticos y recursivos para la solución de este tipo de problemas. En subsecciones siguientes se describirá uno de estos algoritmos para la implementación de SVM (Alex J Smola, Sch, & Schölkopf, 2004).

2.3.2 Kernel

Los kernel son funciones que mapean un conjunto de características a un espacio denominado “espacio de características” donde dicho conjunto de datos se transforma en linealmente separable, para el problema de clasificación o aproximable linealmente, para el caso del problema de regresión. El espacio de características toma su nombre por que

corresponde a una comparación de las características del conjunto de datos original y por consiguiente en este espacio la dimensión del conjunto de características aumenta. En la figura 2.10 se presenta el concepto del mapeo que realizan las funciones de kernel (A. Smola et al., 1996).

Figura 2.10 Mapeo de conjunto de datos original a espacio de características



El objetivo de la transformación de la función kernel consiste en:

$$\Phi: \mathcal{X} \rightarrow \mathcal{H} \text{ por lo tanto } x \rightarrow \mathbf{X} = \Phi(x)$$

Y generar el nuevo espacio a través del producto punto define la función de kernel como:

$$k(x, z) = \Phi(x) \cdot \Phi(z) \tag{2.30}$$

Ejemplo 2.1

Suponga un conjunto de datos con características $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ a la cual se le aplica la siguiente transformación φ :

$$\Phi \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2) \tag{2.31}$$

Aplicando el producto punto se obtiene:

$$\langle \Phi \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right), \Phi \left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right) \rangle = (1 + x_1y_1 + x_2y_2)^2 \tag{2.32}$$

El cómputo de la transformación y el producto punto conlleva un alto gasto de recursos computacionales debido a que las características aumentan, por lo cual surge el concepto de “truco kernel” donde en lugar de definir la función de kernel como el producto punto de

las transformaciones, se establece analíticamente una expresión que dependa de las características del conjunto de entrada.

Para el caso de este ejemplo

$$k(x, y) = (1 + x_1y_1 + x_2y_2)^2 \quad (2.33)$$

De esta manera, se evita el cálculo de las transformaciones ϕ para cada una de las características.

Construcción de funciones de Kernel

Para que una función de kernel sea válida debe cumplir con la condición de Mercer (A. Smola et al., 1996)

En lo que a SVM se refiere, los kernel más utilizados son los siguientes:

- *Polinómico:*

$$k(x, z) = (x^T z + 1)^p \quad (2.34)$$

Donde p es el orden de polinomio

- *Radial basis o Gaussiano*

$$k(x, z) = e^{-\frac{\|x-z\|^2}{2\sigma^2}} \quad (2.35)$$

Donde σ es el ancho de la campana de Gauss

- *Lineal*

$$k(x, z) = x^T z \quad (2.36)$$

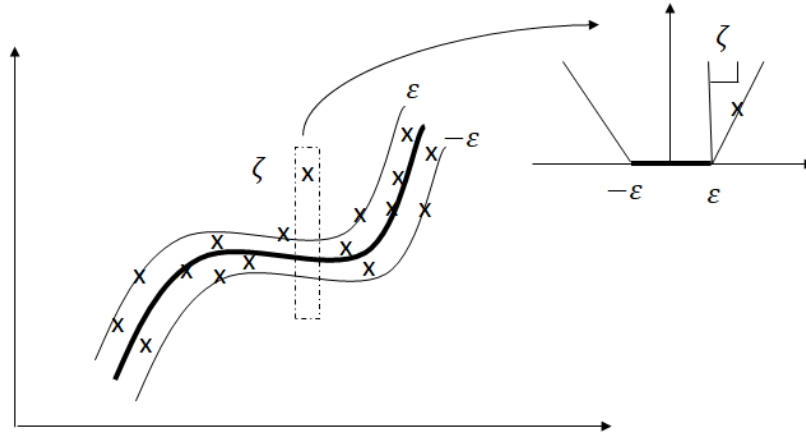
Nótese que, en los 3 casos anteriormente enunciados, las funciones de kernel solamente dependen de los datos de entrada, lo cual disminuye el consumo de recursos computacionales comparando con la aplicación del kernel en su definición, presentada en (2.33)

2.3.3 Regresión no lineal

Como se dijo en la sección 2.3.2, las funciones de kernel son utilizadas para transformar espacios no lineales en lineales, por lo tanto, esta característica se aprovecha en la solución del problema de regresión no lineal a través de SVM. Al igual que en el caso lineal

asumiremos que es posible encontrar una función f que se aproximara a los datos de entrenamiento con una desviación máxima de ε . Sin embargo, esta función pertenecerá al “espacio de características” generado por medio de la función de kernel lo que nos llevará nuevamente a un problema lineal:

Figura 2.11 Interpretación gráfica de la regresión no lineal por SVM



$$f(\Phi(x)) = \sum w * \Phi(x) + b \quad (2.37)$$

Siguiendo el mismo procedimiento que en el caso lineal se obtiene:

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 \\ \text{con } & \begin{cases} y_i - \langle w, \Phi(x_i) \rangle - b \leq \varepsilon \\ \langle w, \Phi(x) \rangle + b - y_i \leq \varepsilon \end{cases} \end{aligned} \quad (2.38)$$

Aplicando el concepto de margen suave

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ \text{con } & \begin{cases} y_i - \langle w, \Phi(x_i) \rangle - b \leq \varepsilon + \xi_i \\ \langle w, \Phi(x) \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (2.39)$$

En (2.40) es posible apreciar que el problema a resolver es igual al caso de la regresión lineal, con la particularidad que ya no depende del conjunto de datos original sino de una transformación de los mismos. Por consiguiente, a continuación se formula el problema de optimización con variables duales (A. Smola et al., 1996):

$$\max \begin{cases} -\frac{1}{2} \sum_{j,i=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, x_j) - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i(\alpha_i - \alpha_i^*) \\ \text{con } \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \text{ y } \alpha_i, \alpha_i^* \in [0, C] \end{cases} \quad (2.40)$$

En (2.40) la expresión se conserva igual en su mayoría, pero ya no está en términos del producto punto sino del kernel. Dicho kernel puede ser cualquier función que cumpla con la condición de Mercer (A. Smola et al., 1996).

De lo anterior, se obtiene la expresión para $f(x)$ en términos del kernel:

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*)\Phi(x_i) \quad (2.41)$$

$$\text{Entonces } f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*)k(x, x_i) + b \quad (2.42)$$

La expresión (2.42) es más general que la (2.27) ya que como se dijo en la sección anterior el producto punto corresponde al kernel lineal, por lo tanto, si se está resolviendo un caso de regresión lineal basta con aplicar en la expresión (2.42) un kernel de este tipo.

Para finalizar, (2.40) nuevamente es un problema de optimización de variable dual, el cual puede ser solucionado por diferentes métodos, temática que será abordada en la siguiente sección.

2.3.4 Entrenamiento

La solución del problema de optimización de (2.40) o (2.25), produce como resultado el cálculo de los parámetros α_i y α_i^* los cuales definirán la función $f(x)$ que aproximará a los datos del conjunto de entrenamiento. En otras palabras, la solución de (2.40) constituye el entrenamiento de la SVM.

- **Algoritmos de entrenamiento**

Los problemas de optimización de variable dual revisten tal complejidad que requieren algoritmos de una alta abstracción analítica o un alto gasto computacional, razón por la cual en la literatura se encuentran reportados múltiples trabajos que abordan este tema en el marco del entrenamiento de las SVM.

- (QP³⁰) Métodos numéricos de programación cuadrática

Es uno de los métodos más conocidos, debido a su fácil implementación, sin embargo, en su proceso de cálculo requiere la conformación de matrices en el espacio de Hilbert que resultan conteniendo una gran cantidad de términos, aproximadamente el cuadrado del tamaño del conjunto de datos de entrada, por lo consiguiente ante conjuntos de entrenamiento grandes el gasto computacional es demasiado alto. Entre otras causas, esta es una de las cuales no permitió el avance de las máquinas de vectores de soporte desde que se reportaron en (Vapnik & Lerner, 1963).

- Algoritmo “Chunking”

En (Boser, Guyon, & Vapnik, 1992) se reportó por primera vez un algoritmo de base analítica, donde lo que se pretende es fragmentar series de pequeños problemas de variable dual y aplica QP para solucionarlos. El algoritmo “Chunking³¹” evalúa en cada paso todos los multiplicadores de Lagrange que resultaron ser 0 (cero) en el anterior paso y los M patrones que violan las condiciones KKT³², al final el algoritmo descarta todas las entradas cuyos multiplicadores de Lagrange sean 0 (cero) y produce una matriz que contiene todos los patrones cuyos multiplicadores de Lagrange son diferentes de 0 (cero). El algoritmo “Chunking” reduce el tamaño de las matrices bastante con respecto a los métodos QP completos, del número de muestras del conjunto de datos al cuadrado a la cantidad de multiplicadores de Lagrange iguales a 0 (cero) al cuadrado, sin embargo, continúan resultando matrices con gran cantidad de términos y los costos computacionales ante conjuntos de datos grandes hacen ineficiente este algoritmo.

- Algoritmo “Osuna”

Nombrado debido a su autor, en (Boser et al., 1992) se reportó una mejora al algoritmo “Chunking”, donde nuevamente se divide el problema QP en pequeños sub problemas, los cuales están sujetos a un tamaño fijo de matriz y en cada paso evalúa una muestra de datos y descarta una muestra de datos hasta detectar aquellas muestras que violen las condiciones KKT. Claramente este algoritmo resulta ineficiente porque es necesario recorrer paso a paso todas las muestras, sin embargo, en la práctica se agregan y descartan grupos de varias muestras para agilizar el tiempo de ejecución del algoritmo.

³⁰ QP por sus siglas en inglés Quadratic Programming Algorithms

³¹ Palabra en idioma inglés que traduce pedazos

³² Condiciones Karush-Kuhn-Tucker para solución n de problemas de variable doble

- Algoritmo de optimización secuencial mínima SMO³³

Siguiendo con la línea de los dos algoritmos anteriores, en (Platt, 1999), se reportó un nuevo algoritmo para la solución de este tipo de problemas, por medio de la división del problema general en sub-problemas, sin embargo, este algoritmo recurre a solucionar el sub-problema más simple posible, esto es encontrar solución para dos multiplicadores de Lagrange en cada paso. Este tipo de solución es posible deducirla analíticamente por lo cual es factible programarla en unas pocas líneas de código y no es necesario recurrir a ningún método numérico. Dicha característica es lo que hace a este algoritmo contar con un desempeño alto ya que no debe conformar matrices auxiliares ni ejecutar una gran cantidad de cálculos, lo cual explica que en la literatura sea el algoritmo más recomendado para el entrenamiento de SVM. En contraste, posee una gran desventaja la cual es que la implementación total del algoritmo requiere un conocimiento avanzado en la solución de multiplicadores de Lagrange y la programación de algoritmos matemáticos, razón por la cual no hay suficiente evidencia de implementación en la literatura. En (Platt, 1999) se reportó solución al problema de clasificación por SVM, sin embargo, en (Alex J Smola et al., 2004) se presentó una versión de SMO para el problema de regresión.

- Algoritmo Kernel-Adatron

En (Campbell & Cristianini, 1999) se reportó un algoritmo que presentaba un cambio en el enfoque para dar la solución al problema de optimización de las SVM, donde se aplica una rutina de gradiente ascendente para maximizar la expresión de (2.40) en el problema de clasificación por medio de SVM, el cual presenta una fácil implementación (pocas líneas de código) y garantiza la convergencia además de facilitar el cálculo de los parámetros del kernel y la tasa de aprendizaje (η). En (Veropoulos, 2001) se presenta la versión del algoritmo Kernel-Adatron para el problema de regresión el cual se detallará a continuación.

De (2.40) es posible escribir el Lagrangiano con las variables duales

$$L = -\frac{1}{2} \sum_{j,i=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, x_j) - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i(\alpha_i - \alpha_i^*) + \lambda \sum_{i=1}^l (\alpha_i^* - \alpha_i) \quad (2.43)$$

Donde el último término garantiza la condición $\sum_i \alpha_i = \sum_i \alpha_i^*$

Como el objetivo es maximizar L en (2.43) por el método de gradiente ascendente se aplica:

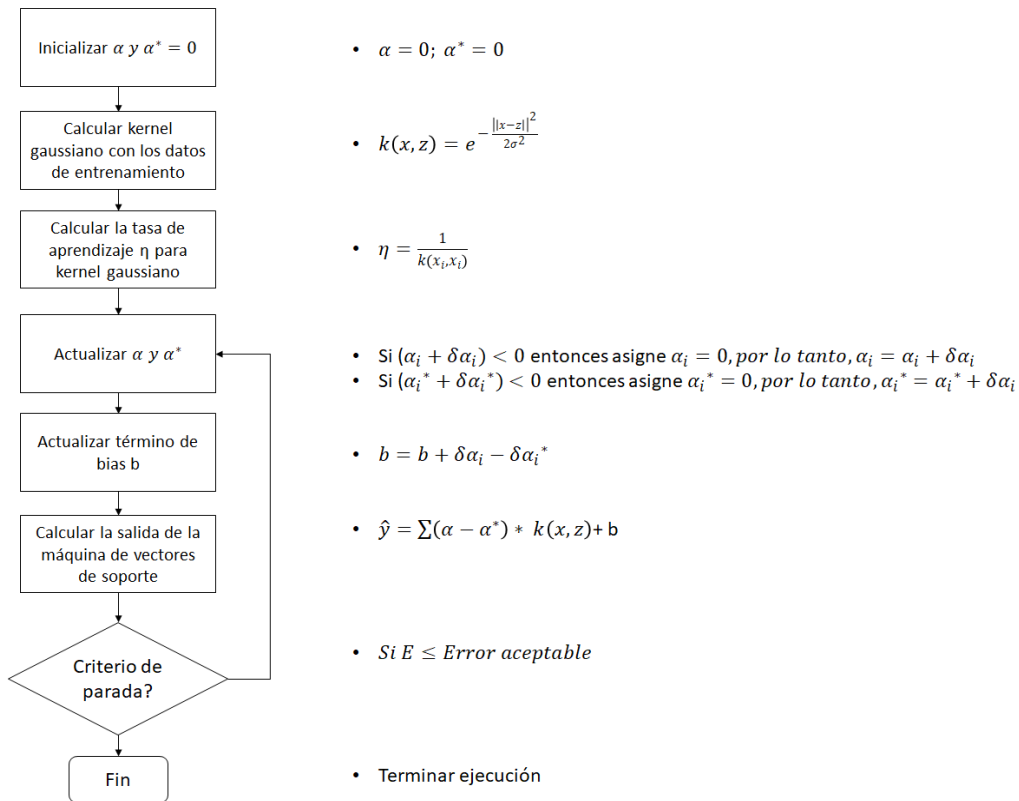
³³ SMO por sus siglas en inglés Sequential minimal optimization

$$\delta\alpha_i = \eta \frac{\partial}{\partial\alpha_i} L = \eta \left(- \sum_{i=1}^l (\alpha_i - \alpha_i^*) k(x_j, x_i) + y_i - \varepsilon - \lambda \right) \quad (2.44)$$

$$\delta\alpha_i^* = \eta \frac{\partial}{\partial\alpha_i^*} L = \eta \left(\sum_{i=1}^l (\alpha_i - \alpha_i^*) k(x_j, x_i) - y_i - \varepsilon + \lambda \right) \quad (2.45)$$

En la figura 2.12 se ilustra el procedimiento de entrenamiento por medio de un diagrama de flujo.

Figura 2.12 Diagrama de flujo para entrenamiento de máquinas de vectores de soporte por algoritmo Kernel-Adatron



Los multiplicadores de Lagrange se actualizan en cada iteración hasta encontrar la solución utilizando directamente la definición de gradiente ascendente presentada en (2.46) y (2.47).

$$\alpha_i(nuevo) = \alpha_i(anterior) + \eta\delta\alpha_i \quad (2.46)$$

$$\alpha_i^*(nuevo) = \alpha_i^*(anterior) + \eta\delta\alpha_i^* \quad (2.47)$$

El término de bias b se calcula a partir de los δ calculados en cada iteración así:

$$b(\text{nuevo}) = b(\text{anterior}) + \delta\alpha_i + \delta\alpha_i^* \quad (2.48)$$

Para calcular la tasa de aprendizaje (η) se aplica lo siguiente

$$\eta = \frac{1}{k(x_i, x_i)} \quad (2.49)$$

Nótese que el η es directamente dependiente del kernel seleccionado, por ejemplo:

- Kernel Polinómico

$$\eta_i = \frac{1}{(|x_i|^2 + 1)^p} ; \text{ con } p \text{ como el orden del polinomio}$$

Por último, a continuación, se presenta el algoritmo regresión por Kernel-Adatron

-
1. Asignar el valor de ε e inicializar $\alpha_i = 0$ y $\alpha_i^* = 0 \forall i$ y $b = 0$
 2. Para todos los patrones $i=1, 2, \dots, l$ realizar los pasos 3 y 4
 3. Calcular η y entonces evaluar:
 - a) $\delta\alpha_i = \eta(-\sum_{j=1}^l (\alpha_j - \alpha_j^*) k(x_j, x_i) + y_i - \varepsilon - \lambda)$

Si $(\alpha_i + \delta\alpha_i) < 0$ entonces asigne $\alpha_i = 0$, por lo tanto, $\alpha_i = \alpha_i + \delta\alpha_i$

- b) $\delta\alpha_i^* = \eta(\sum_{j=1}^l (\alpha_j - \alpha_j^*) k(x_j, x_i) - y_i - \varepsilon + \lambda)$

Si $(\alpha_i^* + \delta\alpha_i^*) < 0$ entonces asigne $\alpha_i^* = 0$, por lo tanto, $\alpha_i^* = \alpha_i^* + \delta\alpha_i^*$

- c) $b = b + \delta\alpha_i - \delta\alpha_i^*$

4. Repetir desde el paso 2 hasta que el criterio de parada se cumpla o el número de iteraciones se complete
-

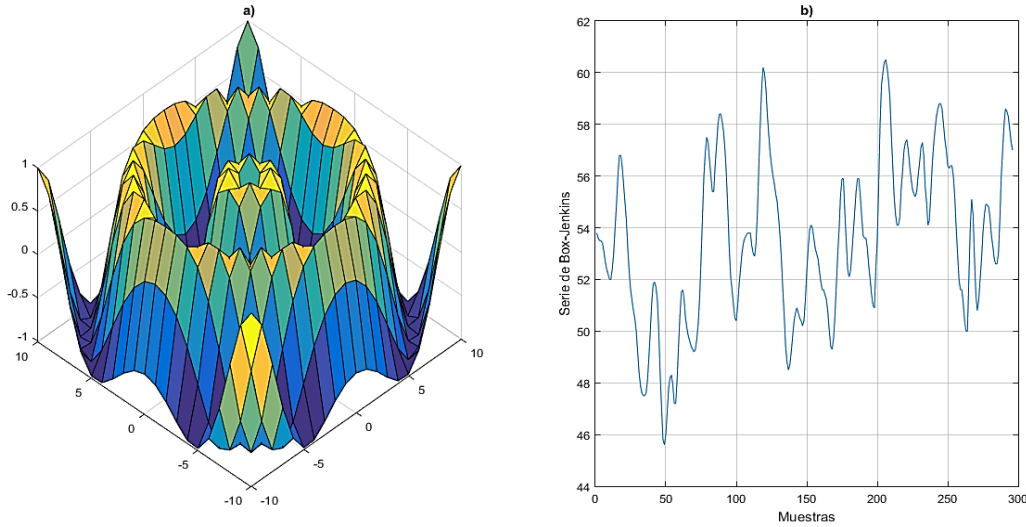
Algoritmo 2.3 Algoritmo de regresión por Kernel-Adatron. Tomado de (Veropoulos, 2001).

2.4 Análisis Comparativo

Con el propósito de establecer las características, alcances, limitaciones, ventajas y desventajas de las diferentes técnicas de aprendizaje de máquina, a continuación, se realiza un comparativo entre las ANN, SVM y sistemas ANFIS pretendiendo conocer el desempeño de cada técnica con respecto a algunos aspectos como la calidad de los datos, la arquitectura seleccionada, entrenamiento, entre otros.

Dicho análisis se realiza a partir de 2 conjuntos de datos representativos, como primera medida, un sistema estático representado por una superficie en 3 dimensiones (ver figura 2.13 a)) y por otro lado un sistema dinámico altamente referenciado en la literatura científica denominado serie de Box-Jenkins, ampliamente utilizado en la identificación de sistemas dinámicos (ver figura 2.13 b)).

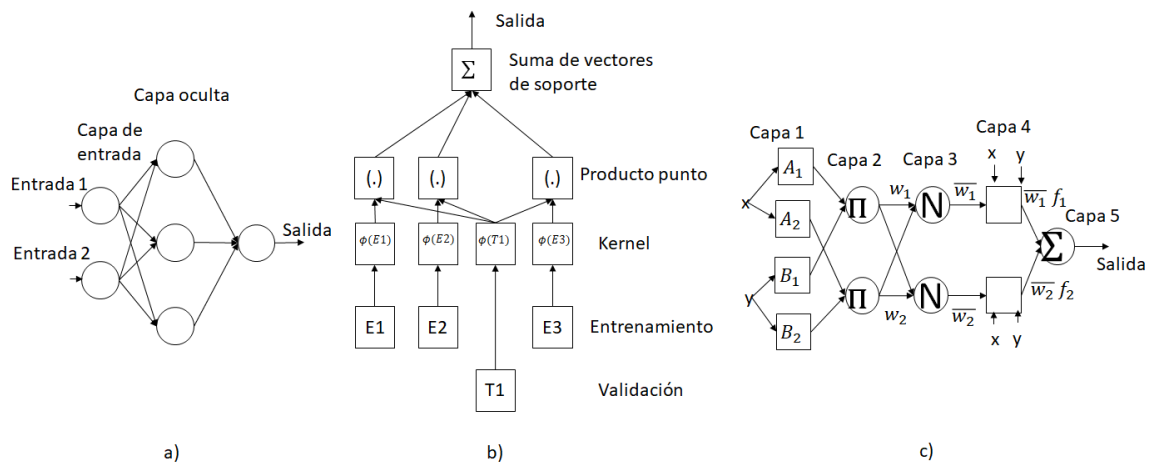
Figura 2.13 Conjuntos de datos de entrenamiento a) Sistema estático (Superficie) b) Sistema dinámico (Serie de Box-Jenkins)



2.4.1 Arquitectura

Las técnicas de aprendizaje seleccionadas, es decir, ANN, SVM y sistemas ANFIS, poseen una arquitectura de red de múltiples capas (ver figura 2.14), donde cada nodo lleva a cabo operaciones matemáticas determinadas y cada interconexión es cuantificada por medio de un valor que representa el peso de la conexión dentro del sistema. Teniendo en cuenta esto, el desempeño de cada una de las técnicas definidas se ve afectado por la selección de los parámetros propios de la arquitectura de cada técnica, entre los que se encuentran la cantidad de nodos, las funciones de activación seleccionadas, tasa de aprendizaje, etc.

Figura 2.14 Arquitectura a) ANN, b) SVM y c) ANFIS. Tomado de (Bishop, 2006; J. S. R. Jang, 1993; Alexander J Smola & Schölkopf, 2004)



▪ **Nodos de Procesamiento**

Este es uno de los parámetros más influyentes en las técnicas de la figura 2.14, ya que afectan tanto el proceso de entrenamiento como el desempeño en la predicción. En el caso de las ANN el diseñador define el número de neuronos en la capa oculta por medio del método de validación cruzada buscando el caso de mejor desempeño. Para los sistemas ANFIS el número de nodos de procesamiento corresponde al número de reglas definido para el FIS, esta cantidad no la define directamente el diseñador, sino que es calculada a través del algoritmo de clustering, por consiguiente, el diseñador selecciona el parámetro de radio de cluster (r_a) adecuado; también es posible utilizar validación cruzada para lograr el mejor desempeño posible. Por último, en las SVM el diseñador no define la cantidad de nodos ya que estos dependen del conjunto de datos, sin embargo, es necesario definir el tipo de kernel que se utilizará en dichos nodos. Las arquitecturas definidas para ANN y ANFIS establecen función sigmoidea y función gaussiana respectivamente, razón por la cual, no es necesario seleccionar funciones de activación por parte del diseñador.

Tabla 2.2 MSE de entrenamiento y validación para ANN y ANFIS frente a variación de cantidad de neuronos y reglas respectivamente. Sistema estático.

Sistema Estático					
	ANN			ANFIS	
Cantidad nodos	MSE Entrenamiento	MSE Validación	Cantidad reglas	MSE Entrenamiento	MSE Validación
2	0.114	0.126	5	0.050	0.054
4	0.027	0.061	6	0.039	0.039
6	0.013	0.043	11	0.012	0.017
8	0.001	0.008	23	0.001	0.007
10	2.66e-4	0.006	90	2.09e-4	0.006
20	1.61e-4	0.006	182	2.81e-5	0.008

Tanto en la tabla 2.2 como en la tabla 2.3 se observa que al aumentar los neuronos o las reglas, el error de entrenamiento disminuye progresivamente; así mismo, el error de validación tiende a disminuir hasta llegar al punto de sobre ajuste, a partir del cual empieza a aumentar. Por ejemplo, en la tabla 2.2 se observa que con 90 reglas se obtiene un error de validación de 0.006 pero al aumentar a 182 reglas el MSE de validación aumenta a 0.008; lo anterior indica que hubo sobreajuste del modelo y por lo tanto la predicción se degrada.

Existen criterios definidos por múltiples autores para establecer el número apropiado de neurodos en la capa oculta, no obstante, dichos criterios no son absolutos y dependen de diferentes condiciones aclaradas respectivamente en (Bishop, 2006). En consecuencia, lo recomendable es utilizar el método de validación cruzada buscando la cantidad de neurodos que ofrezcan el mejor desempeño tanto en entrenamiento como en predicción.

Tabla 2.3 MSE de entrenamiento y validación para ANN y ANFIS frente a variación de cantidad de neurodos y reglas respectivamente. Sistema dinámico.

Sistema Dinámico					
	ANN			ANFIS	
Cantidad	MSE Entrenamiento	MSE Validación	Cantidad Reglas	MSE Entrenamiento	MSE Validación
2	0.0015	0.040	2	1.66e-6	0.350
4	7.47e-4	0.040	3	2.66e-7	4.09e4
6	5.51e-4	0.039	6	5.64e-7	0.354
8	5.21e-4	0.039	8	7.32e-6	0.352
10	4.87e-4	0.039	39	5.58e-6	0.350
20	3.52e-4	0.038	48	1.27e-30	0.345

En cuanto a las SVM, la teoría reportada en (Alex J Smola et al., 2004) indica que el único criterio para la selección del kernel es que la función de transformación cumpla con las condiciones de Mercer (Alex J Smola et al., 2004), debido a que con esto se garantiza que el espacio de características obtenido sea apropiado para la solución del problema de optimización. No obstante, en los trabajos de implementación reportados, el kernel más usado es el gaussiano, fundamentalmente por dos razones, primero por el bajo costo computacional en comparación con otros kernel como el polinómico, y segundo porque el kernel gaussiano permite la aproximación de todo tipo de no linealidades, por lo consiguiente es denominado el “aproximador universal” (Alpaydın, 2010).

▪ **Tasa de Aprendizaje**

Como se describió en subsecciones anteriores, se definieron los algoritmos de entrenamiento más difundidos de cada una de las técnicas y como método de optimización se utilizan métodos de gradiente, por lo tanto, las tres técnicas dependen del parámetro η , también denominado tasa de aprendizaje, en su proceso de entrenamiento.

Figura 2.15 Variación parámetro η en ANN a) Sistema estático, b) Sistema dinámico

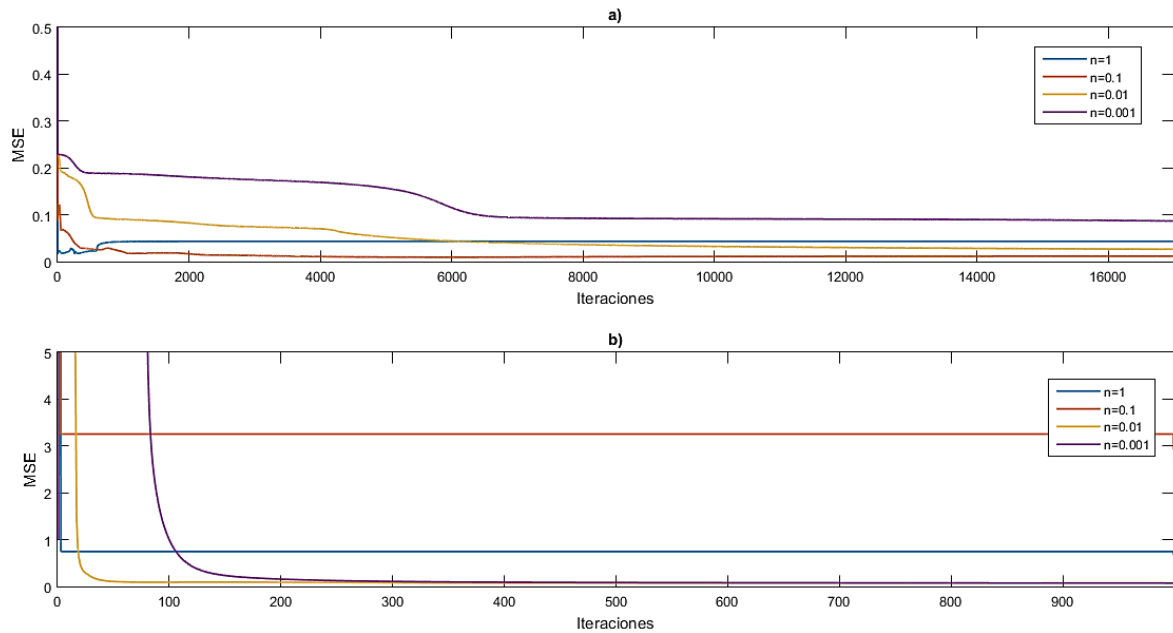
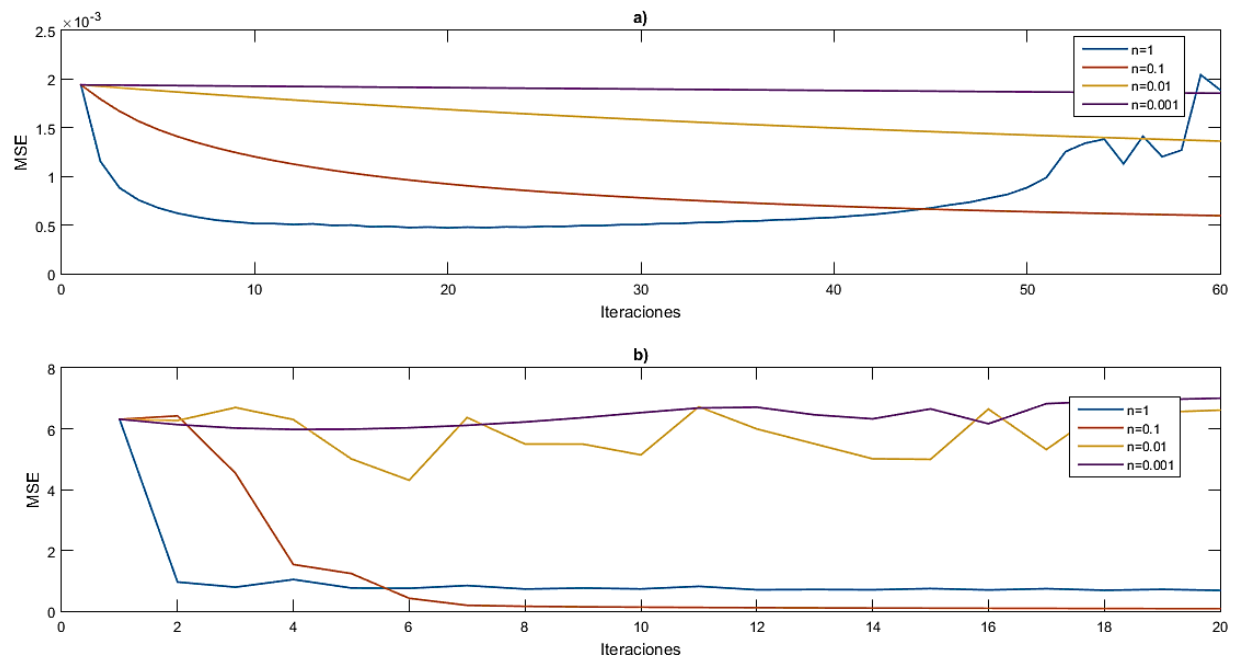


Figura 2.16 Variación parámetro η en ANFIS a) Sistema estático, b) Sistema dinámico

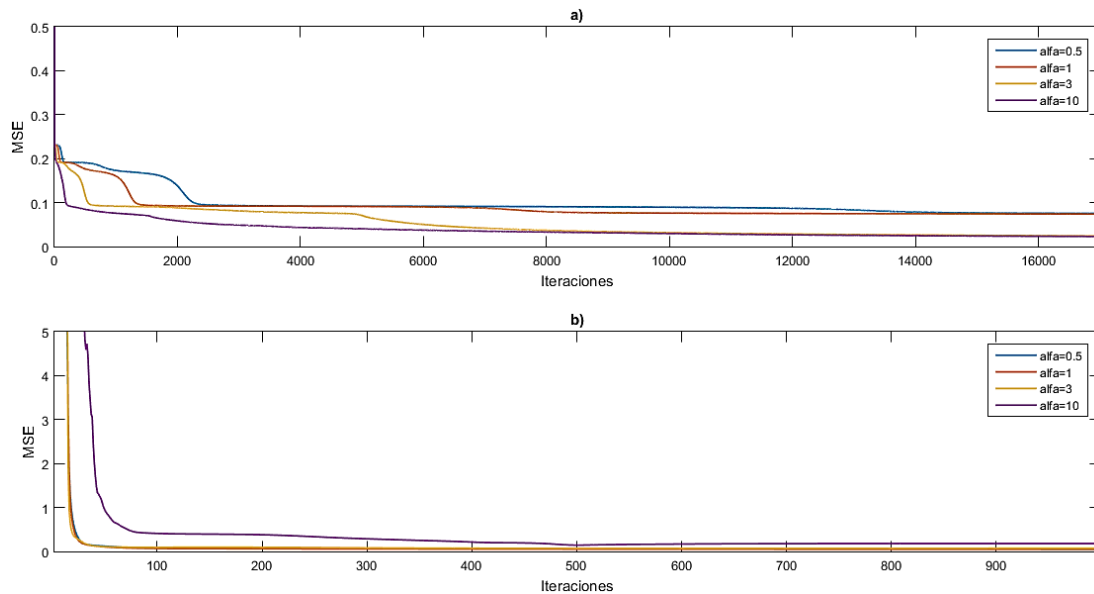


La tasa de aprendizaje tiene un efecto directo en la cantidad de iteraciones que necesitan los algoritmos de entrenamiento para ANN y ANFIS, sin embargo, es importante que este parámetro no se defina con un valor muy alto debido a que esto puede llevar al algoritmo a no converger e incluso a que el error aumente, como se observa en la figura 2.15 donde para $\eta=1$ el algoritmo no reduce el error, por el contrario, tiende a mantenerlo constante. En contraste, si se selecciona un valor muy pequeño para la tasa de aprendizaje, los algoritmos necesitarán un número mayor de iteraciones para lograr el objetivo de error, caso que se puede observar en las figuras 2.15 y 2.16 con $\eta=0.001$, por consiguiente, el entrenamiento tomará más tiempo de ejecución.

Mientras que para ANN y ANFIS el parámetro η es definido por el diseñador y se recurre a la validación cruzada para seleccionar el valor más adecuado, en las SVM la tasa de aprendizaje es dependiente directamente de los datos de entrenamiento, de acuerdo al criterio descrito en (Veropoulos, 2001) donde se establece que el parámetro η se puede calcular a partir del kernel utilizado en la SVM.

▪ **Parámetros de las funciones de procesamiento**

Cada una de las técnicas de aprendizaje, contienen en sus respectivas arquitecturas, nodos de procesamiento donde los conjuntos de datos se aplican a una función matemática que de manera general tienen como objetivo llevar dichos datos a un espacio no lineal. Cada una de estas funciones tiene unos parámetros de ajuste que modifican sus características, por ejemplo, en el caso de las ANN encontramos el parámetro alfa (α) de la función sigmoideal que varía la velocidad de cambio de la función. En cuanto a los sistemas ANFIS se encuentran las funciones de pertenencia correspondientes a los antecedentes de cada regla, las cuales dependen de dos parámetros, el centro o media μ de los datos y el ancho o desviación estándar de los datos σ . Por último, entre los parámetros de las SVM se encuentra p en el caso del kernel polinómico, que es el grado del polinomio y por otro lado se encuentra el ancho de la campana de Gauss (σ) utilizada para el kernel Gaussiano. De acuerdo a los algoritmos de entrenamiento descritos en las secciones anteriores, se infiere que las ANN y las SVM requieren que el diseñador asigne valores adecuados a estos parámetros utilizando validación cruzada para encontrar el mejor desempeño; en contraste, los sistemas ANFIS en su etapa de clustering establecen los valores iniciales para las funciones de pertenencia y a través del proceso de entrenamiento se llega a los valores definitivos

Figura 2.17 Variación parámetro α en ANN a) Sistema estático b) Sistema Dinámico

El parámetro alfa (α), en la función sigmoideal utilizada en las ANN, afecta principalmente el número de iteraciones necesarias para obtener el objetivo de ajuste de los datos, es decir, si se selecciona un valor muy pequeño tomará muchas iteraciones lograr el entrenamiento, tal como se observa en la figura 2.17 de modo diferente, si se selecciona un valor muy alto entonces el algoritmo entrenará la red con menos iteraciones, sin embargo, el valor no puede ser excesivamente alto debido a que esto puede llevar al algoritmo a un bajo desempeño en la validación o incluso a que no converja el entrenamiento.

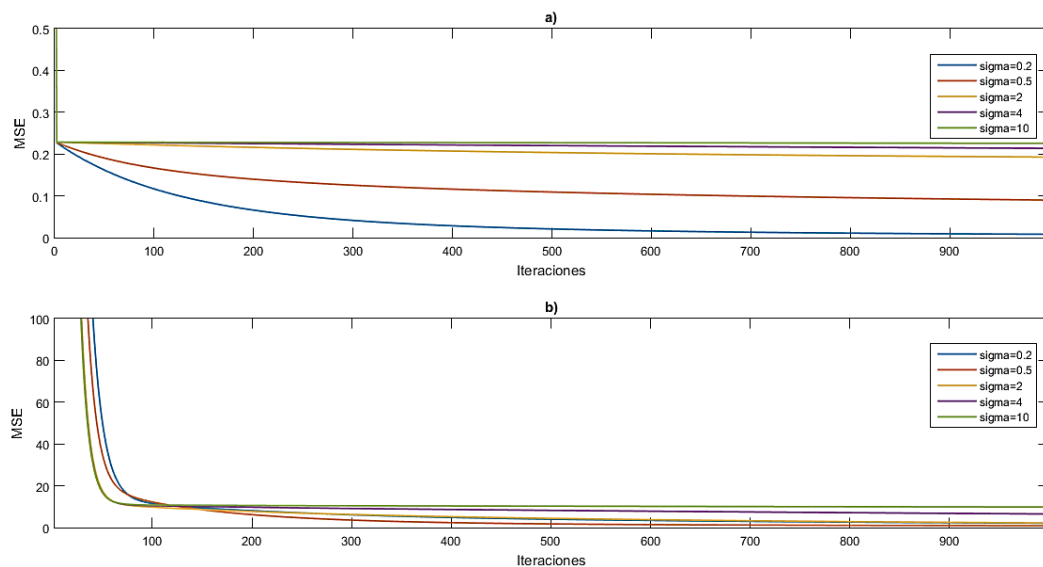
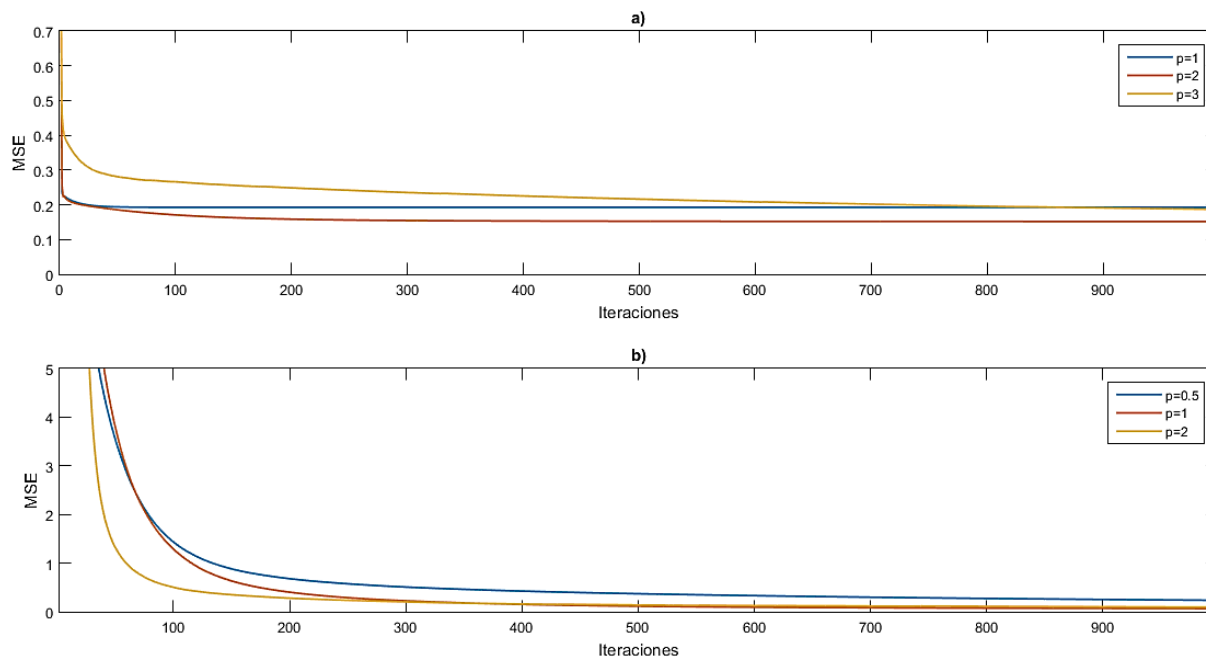
Figura 2.18 Variación parámetro σ en kernel SVM a) Sistema estático b) Sistema dinámico

Figura 2.19 Variación parámetro p kernel polinómico SVM a) Sistema Estático b) Sistema Dinámico

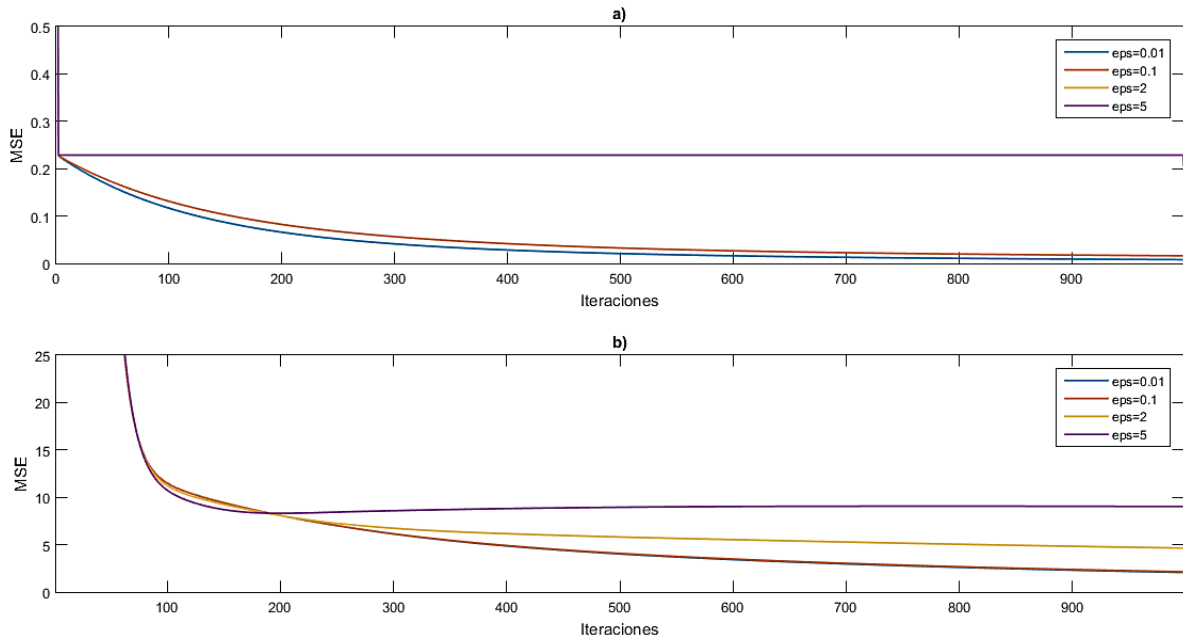


Tanto el parámetro p como el parámetro σ en los respectivos kernel de las SVM, afectan el error final de entrenamiento y validación ya que en valores extremos producen sobre-entrenamiento o sub-entrenamiento según sea el caso, como se observa en las figuras 2.18 y 2.19. Para resumir los parámetros α , σ y/o p producen el mismo efecto sobre las ANN y las SVM respectivamente, con influencia directa en la velocidad con que converge el algoritmo. Así mismo es un parámetro importante cuando hay datos con características como datos fuera de tendencia o ruido porque su ajuste adecuado permite que el algoritmo converja apropiadamente.

▪ Parámetros propios de cada técnica

Además de los parámetros anteriormente tratados, cada una de las técnicas posee parámetros propios que afectan el desempeño de aprendizaje, a continuación, se especifica la influencia de dichos parámetros. En primera instancia, las SVM cuentan con el parámetro C , que corresponde al “margen suave” definido en la sección 2.3, el cual define la penalización del modelo ante datos que queden por fuera de la función de pérdidas. Su impacto es posible verlo en conjuntos con una desviación estándar mayor a 0, razón por la cual con los conjuntos de datos seleccionados la influencia del parámetro C no se puede apreciar.

Figura 2.20 Variación parámetro máxima desviación ε en SVM a) Sistema Estático b) Sistema Dinámico



Por otro lado, las SVM tienen otro parámetro definido por el diseñador, denominado ε , el cual define la máxima desviación que pueden tener los datos y por ende afecta totalmente los vectores de soporte que son solución de la regresión. De lo anterior, ε está directamente relacionado con el error y por consiguiente el valor debe ser el menor posible, sin embargo, no puede ser excesivamente cercano a 0 (cero) ya que la solución del problema de variables duales puede llegar a ser inviable. En las figuras 2.20 a) y 2.20 b) se presenta el resultado para el entrenamiento con $\varepsilon=5$ el cual converge a un error mayor que los casos de $\varepsilon=0.01$, $\varepsilon=0.1$ y $\varepsilon=2$.

Para finalizar, el algoritmo de clustering substractivo tiene un par de parámetros denominados $\bar{\varepsilon}$ y $\underline{\varepsilon}$, los cuales afectan la cantidad de clusters aceptados por el algoritmo. En (J. S. R. Jang, 1993), el autor recomienda $\bar{\varepsilon} = 0.5$ y $\underline{\varepsilon} = 0.15$ en múltiples pruebas realizadas a conjuntos de datos con diferentes comportamientos no lineales. En los sistemas ANFIS, estos parámetros afectan la cantidad de reglas de la base de conocimiento, sin embargo, ya se estableció anteriormente cómo reaccionan las redes ANFIS ante diferentes tamaños de la base de reglas. En general, se dejarán fijos estos parámetros en los valores sugeridos por el autor porque afectan el resultado del clustering más no el resultado del entrenamiento en el sistema ANFIS.

2.4.2 Entrenamiento y Convergencia

Cada una de las técnicas descritas en el presente documento, muestran unas características ante el proceso de entrenamiento, las cuales se analizan utilizando los conjuntos de datos presentados en la sección 2.4, al igual que en el análisis anterior, se

corren los algoritmos tanto con una superficie estática con el objetivo de identificar el comportamiento de cada una de las técnicas en cuanto a convergencia, teniendo en cuenta que todas utilizan un método de solución basado en gradiente.

▪ **Iteraciones**

Debido a que todas las técnicas consideradas, tienen una solución basada en gradiente, es posible comparar su comportamiento frente a la cantidad de iteraciones necesaria para obtener convergencia con un error medio cuadrático máximo de 0.01, en otras palabras, el procedimiento para poder comparar consiste en correr cada uno de los algoritmos la cantidad de veces necesaria para obtener un $MSE \leq 0.01$.

Tabla 2.4 MSE de entrenamiento para ANN, SVM y ANFIS frente a número de iteraciones necesarias

	ANN		SVM		ANFIS	
Sistema	Iteraciones	MSE Entrenamiento	Iteraciones	MSE Entrenamiento	Iteraciones	MSE Entrenamiento
Estático	17000	0.0085	2000	0.0129	15	0.0038
Dinámico	2000	0.0069	50000	0.0155	15	7.01e-7

De la tabla 2.4, es posible observar directamente que en el conjunto de datos que corresponde al sistema estático las ANN requieren más iteraciones para lograr la convergencia, en contraste las SVM requirieron menos iteraciones con respecto a las ANN. Sin embargo, el resultado más notable es el de los sistemas ANFIS los cuales lograron el objetivo de error en tan solo 15 iteraciones; esto debido al entrenamiento híbrido descrito en la sección anterior. Por otro lado, en los sistemas dinámicos, las ANN y ANFIS mejoran su comportamiento ya que el objetivo se consigue en menos iteraciones, lo cual implica un gasto computacional menor.

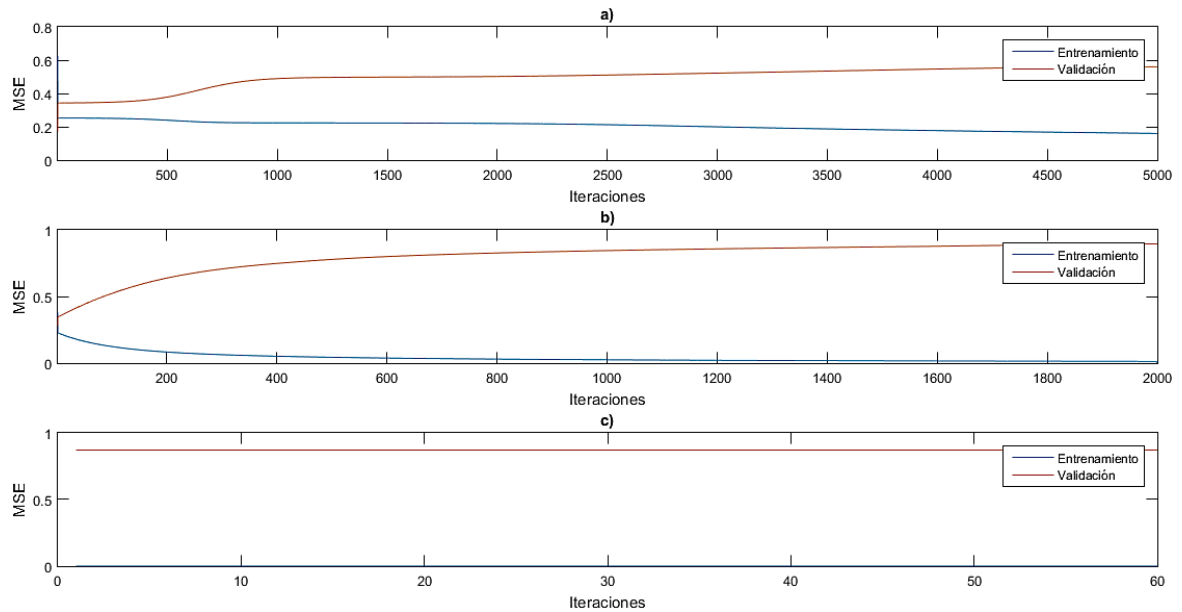
▪ **Interpolación**

Todas las técnicas de aprendizaje de máquina y en particular aquellos basados en aprendizaje supervisado, poseen una capacidad de predicción para datos de entrada que no hacen parte del conjunto de datos de entrenamiento, pero están dentro del mismo espacio. A este concepto se le denomina interpolación y es posible establecer que, si el error de validación es alto con respecto al error de entrenamiento, el algoritmo tiene baja capacidad de interpolación; el algoritmo tiene una alta capacidad de interpolación si se presenta el caso contrario. Con lo anteriormente expuesto se procede a realizar el análisis

de la capacidad de interpolación de cada una de las técnicas establecidas ante los conjuntos de datos ya mencionados en la sección 2.4.

Teniendo en cuenta que la capacidad de interpolación es una característica inherente a la técnica de aprendizaje y no depende de los datos de entrenamiento, el análisis se realiza únicamente con los datos de la superficie estática.

Figura 2.21 Entrenamiento y validación para análisis de interpolación a) ANN, b) SVM, c) ANFIS



Las tres técnicas de aprendizaje presentan una capacidad de interpolación similar ante el conjunto de datos que corresponde a la superficie estática, ver figura 2.21.a). Con el aumento de las iteraciones en la etapa de entrenamiento las técnicas presentan tendencia a perder capacidad de generalización. Así mismo, el entrenamiento y validación para esta prueba se realizó con 30 muestras del conjunto de datos estático, lo cual sugiere que se requieren más datos para mejorar el desempeño en la predicción de las técnicas

▪ Gasto Computacional

Uno de los aspectos de interés en el uso de las técnicas de aprendizaje es el gasto computacional de cada uno de los algoritmos, en razón a que de esto depende directamente el tiempo de entrenamiento de cada una de las técnicas. Las pruebas realizadas a las tres técnicas ya establecidas se ejecutan en un procesador Pentium Dual Core@2.20 GHz con 4GB de memoria RAM, y consisten en tomar los tiempos de ejecución de los algoritmos para una iteración y para un proceso de entrenamiento hasta conseguir un error medio cuadrático menor o igual a 0.001.

Tabla 2.5 Tiempo de ejecución de algoritmos entrenamiento ANN, SVM y ANFIS. Sistema estático

Sistema Estático						
	ANN		SVM		ANFIS	
	Tiempo(s)	MSE Entrenamiento	Tiempo(s)	MSE Entrenamiento	Tiempo(s)	MSE Entrenamiento
1 iteración	0.040	0.268	0.696	0.246	0.707	0.001
Todo entrenamiento	358.458	9.99e-4	3.17	9.98e-4	0.707	0.001

Tabla 2.6 Tiempo de ejecución de algoritmos entrenamiento ANN, SVM y ANFIS. Sistema dinámico

Sistema Dinámico						
	ANN		SVM		ANFIS	
	Tiempo(s)	MSE Entrenamiento	Tiempo(s)	MSE Entrenamiento	Tiempo(s)	MSE Entrenamiento
1 iteración	0.005	2.85	0.012	4.43	0.334	7.01e-7
Todo entrenamiento	15.658	9.99e-4	1.574	9.99e-4	0.334	7.01e-7

En general, de las tablas 2.5 y 2.6 se observa que el algoritmo que toma más tiempo en realizar cada iteración son los sistemas ANFIS, tanto para el conjunto de datos estático como para el dinámico, esto se debe a que en su arquitectura los sistemas ANFIS requieren de 5 capas en las cuales se realizan cálculos sobre los datos, a diferencia de las ANN y SVM que tienen 3 capas y solamente en 2 de ellas se realizan cálculos. En consecuencia, respecto de los tiempos de ejecución, solamente se recomienda el uso de sistemas ANFIS en situaciones donde se tenga sistemas de cómputo lo suficientemente potentes para que los tiempos de cálculo no resulten en una desventaja. Así mismo, las SVM exhiben el mejor desempeño en tiempos de entrenamiento ya que puede ejecutar 5000 iteraciones en 3.17 segundos lo cual supera por mucho las ANN y los sistemas ANFIS en caso de necesitar la misma cantidad de repeticiones en su entrenamiento. Esta velocidad de ejecución se debe principalmente a que las SVM realizan los cálculos en el espacio de características, los cuales resultan más ligeros a nivel de procesamiento con respecto a las ANN y ANFIS.

2.4.3 Calidad de datos

Se realizan pruebas de desempeño a cada una de las técnicas ante diferentes condiciones de degradación de los datos de entrada, con el fin de establecer el comportamiento ante estas características que son típicas en la recolección de datos para aplicaciones como los sensores inferenciales.

- **Patrones Insuficientes en el conjunto de datos**

En la industria no todos los sensores tienen la misma tasa de muestreo, por lo cual es usual que se seleccionen los datos con la tasa de muestreo más baja con el fin de construir el conjunto de datos. Teniendo en cuenta esto, es posible encontrar datos de proceso con tasas de recolección muy bajas, lo cual puede incidir en el desempeño de las técnicas de aprendizaje. A continuación, se presentan los resultados de simular el entrenamiento y validación de las técnicas con los conjuntos de datos seleccionados, retirando muestras del conjunto para emular tasas de muestreo insuficientes.

Tabla 2.7 MSE de entrenamiento y validación para ANN, SVM y ANFIS frente a conjuntos con muestreo insuficiente. Sistema estático.

Sistema estático						
ANN			SVM		ANFIS	
Muestreo	MSE Entrenamiento	MSE validación	MSE Entrenamiento	MSE validación	MSE Entrenamiento	MSE validación
100%	0.0021	0.0079	2.14e-4	0.002	0.0014	0.0075
80%	0.0032	0.0098	2.41e-4	0.0034	0.0023	0.0086
60%	0.0026	0.0095	2.93e-4	0.0043	0.0019	0.0082
30%	0.0205	0.0240	5.26e-4	0.0085	0.0043	0.0107
10%	0.0422	0.0621	5.85e-4	0.0899	8.39e-31	0.0601

Tabla 2.8 MSE de entrenamiento y validación para ANN, SVM y ANFIS frente a conjuntos con muestreo insuficiente. Sistema dinámico.

Sistema dinámico						
ANN			SVM		ANFIS	
Muestreo	MSE Entrenamiento	MSE validación	MSE Entrenamiento	MSE validación	MSE Entrenamiento	MSE validación
100%	5.21e-4	0.040	0.0127	0.0417	4.11e-6	0.3511
80%	9.19e-4	0.096	0.013	0.0604	5.73e-4	0.6409
60%	7.06e-4	0.102	0.0123	0.1648	0.0017	0.2945
30%	2.13e-4	0.048	0.0131	0.5141	4.12e-28	1.78e3
10%	0.0049	0.131	0.0256	2.4969	4.62e-31	0.084

Las SVM solucionan el problema de regresión utilizando la función de pérdidas ϵ -intensiva, lo cual implica que la regresión se realiza sobre una región de ancho 2ϵ , tal como se observa en la figura 2.9, por consiguiente, las SVM presentan robustez ante muestreos insuficientes, tal como se observa en la tabla 2.7, donde el error de validación solamente aumenta significativamente cuando se ha eliminado más del 70% de las muestras.

En cuanto a las redes ANN y ANFIS, la regresión se realiza ajustando punto a punto el error por lo cual, ante conjuntos de datos con insuficientes muestras el error de validación crece tal como se observa en las tablas 2.7 y 2.8.

En contraste, las figuras 2.22 y 2.23 presentan el resultado de simular entrenamiento y validación de los dos conjuntos de datos utilizando desde unos pocos patrones del conjunto hasta el conjunto completo de entrenamiento, con el objetivo de emular el comportamiento con tiempos insuficientes de recolección de datos. Esta característica de las técnicas de aprendizaje también es denominada extrapolación. En la figura 2.22 se observa que un tamaño de conjunto mínimo para lograr convergencia esta alrededor de 300 muestras, mientras que en el caso de los sistemas dinámicos el número mínimo de muestras para lograr convergencia está en las cercanías de 200 patrones.

Figura 2.22 Entrenamiento y validación sistema estático para análisis mínima cantidad de datos
a) ANN, b) SVM, c) ANFIS

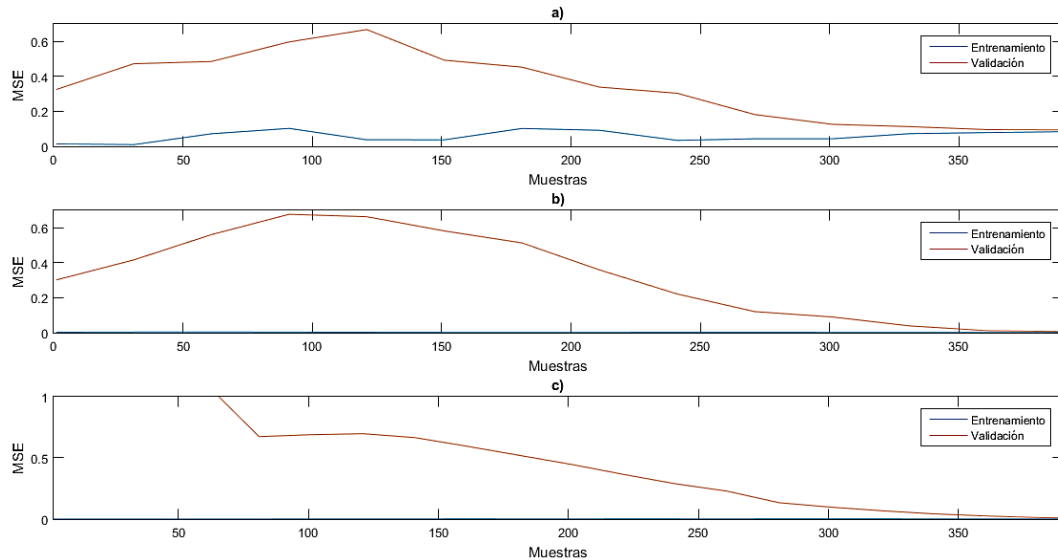
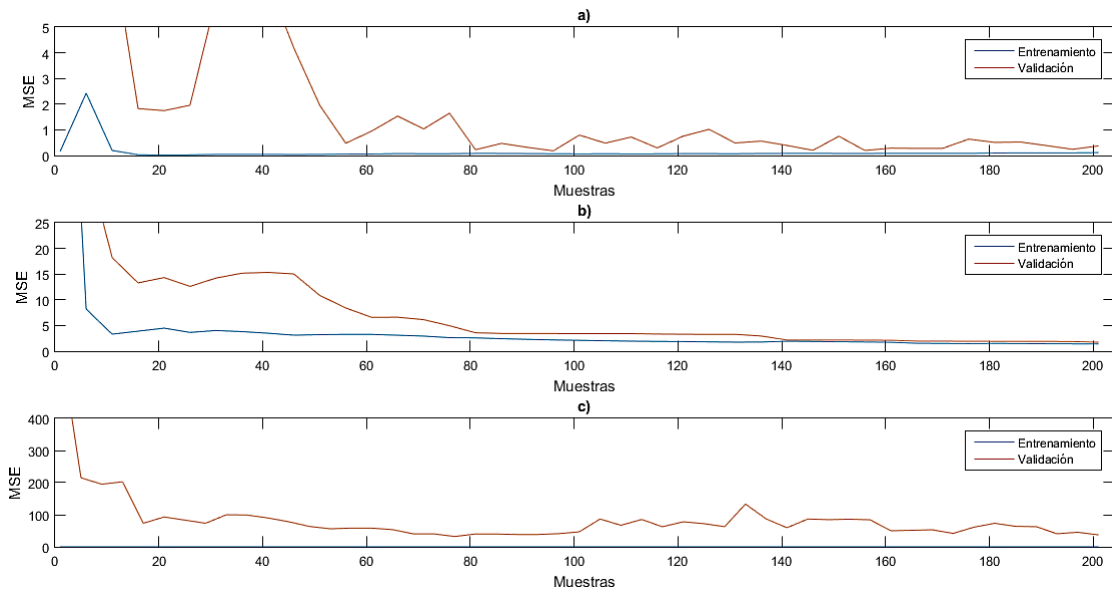


Figura 2.23 Entrenamiento y validación sistema dinámico para análisis mínima cantidad de datos
a) ANN, b) SVM, c) ANFIS



- **Datos fuera de tendencia en el conjunto de datos**

Los datos fuera de tendencia son muestras que no representan el comportamiento que tiene el conjunto de datos en su forma general, y que por el contrario son valores que resultan aislados. Este tipo de datos afectan tanto el entrenamiento como la validación ya que están en los extremos del espacio del conjunto de datos.

Para evaluar el desempeño de las técnicas propuestas, los conjuntos de datos utilizados en las pruebas anteriores se han modificado agregando datos fuera de tendencia de manera aleatoria en diferentes cantidades y amplitudes. Solamente se muestran resultados con el conjunto de datos del sistema estático para evitar duplicar pruebas puesto que los “outliers” son defectos en el conjunto de datos y afectarán de la misma forma ya sea con los datos de la superficie estática o del sistema dinámico.

Tabla 2.9 MSE de validación para conjuntos con datos fuera de tendencia. Sistema estático.

Técnica	Cantidad	0%	10%	20%	30%
	Amplitud				
ANN	2 σ	0.0079	1.6499	2.8231	2.6250
	3 σ	0.0079	2.4570	7.8702	4.7143
	4 σ	0.0079	6.9858	3.3067	4.4576
SVM	2 σ	0.0027	0.6771	0.9304	1.3374
	3 σ	0.0027	0.9019	1.3605	2.5910
	4 σ	0.0027	1.1736	1.8008	2.8668
ANFIS	2 σ	0.0096	2.8299	2.7102	2.9431
	3 σ	0.0096	1.4278	3.9205	8.8950
	4 σ	0.0096	2.7998	4.7893	5.4230

Los datos fuera de tendencia en los conjuntos de entrenamiento degradan el error de entrenamiento en las 3 técnicas tal como se observa en la tabla 2.9, sin embargo, es necesario aclarar algunos aspectos respecto a esto. Como primera medida, se presentan los resultados de validación, ya que corresponde al error medio cuadrático de predicción de la técnica; esta medida indica el desempeño que tiene la técnica frente a conjuntos con datos fuera de tendencia. Por otro lado, se observa de manera clara que al aumentar ya sea la amplitud o la cantidad de “outliers”, el desempeño en la predicción de todas las técnicas se degrada drásticamente. No obstante, entre las 3 técnicas, las SVM presentan mayor robustez frente a datos fuera de tendencia puesto que el error mínimo obtenido es 0.6711 y el máximo es 2.8688 los cuales son mucho menores que para los casos de ANN y ANFIS. Todo lo anterior debido al concepto de “margen suave” que se involucra en las SVM.

Para concluir, se recomienda aplicar algún método para corregir los datos fuera de tendencia y evitar afectaciones en el desempeño de cada una de las técnicas, entre los que se puede encontrar eliminación de datos fuera de tendencia, reemplazarlos por la media o realizar una regresión local. Algunos de estos aspectos se discutirán en capítulos siguientes. En el escenario de tener un conjunto de datos de muy baja calidad respecto a

datos fuera de tendencia y no se pueda filtrar, la técnica que presenta mejor desempeño es las SVM.

▪ Pérdida de datos

Se considera un conjunto de datos con pérdida, aquel donde un intervalo de datos en la entrada esta fuera del rango propio del proceso. En ambientes industriales se presenta cuando hay fallas o averías en alguno de los instrumentos que entrega información del proceso.

Los resultados de las simulaciones con la superficie estática son los siguientes:

Tabla 2.10 MSE de entrenamiento y validación frente a pérdida de datos en el conjunto. Sistema estático.

	ANN		SVM		ANFIS	
	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación
0%	0.0116	0.0161	2.14e-4	0.0027	0.0013	0.0074
10%	0.0792	0.1763	2.07e-4	0.0037	0.0034	0.1025
40%	0.0051	0.0917	2.09e-4	0.0281	0.0014	0.0981

Como es de esperar, técnicas de aprendizaje basadas en datos son afectadas por pérdidas en el conjunto de datos de entrenamiento. Los errores de entrenamiento muestran convergencia en niveles aceptables, sin embargo, cuantos más datos se pierdan, mayor MSE de validación se obtiene.

De la misma manera, se realizaron simulaciones para el sistema dinámico de la columna de destilación

Tabla 2.11 MSE de entrenamiento y validación para ANN frente a pérdida de datos en el conjunto. Sistema dinámico.

	ANN		SVM		ANFIS	
	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación
0%	7e-4	0.0405	0.0202	0.0461	2.68e-6	0.3511
10%	4.03e-4	0.0405	0.0134	0.0090	1.82e-6	0.2915
40%	4.07e-4	0.0402	0.0134	0.0090	1.14e05	0.3534

Para este caso, todas las técnicas presentan un buen desempeño lo que permite intuir que asumir el conjunto de datos como un sistema dinámico, donde se utilizan estados anteriores para predecir el siguiente, puede mejorar el desempeño de las 3 técnicas de aprendizaje frente a conjuntos de entrenamiento y validación con pérdidas de datos.

▪ Conjuntos de datos con ruido

En sistemas de instrumentación industrial es ampliamente conocida la presencia de ruido en las señales de lectura de los sensores, lo cual se ve reflejado en los datos registrados en las bases de datos industriales o los DCS. Para el desarrollo de las siguientes simulaciones se agrega ruido gaussiano a los conjuntos de datos propuestos para evaluar el desempeño de las técnicas.

Tabla 2.12 MSE de entrenamiento y validación frente a conjuntos de datos con ruido. Sistema estático.

Ampl	ANN		SVM		ANFIS	
	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación
0	0.0011	0.0069	2.14e-4	0.0027	0.0013	0.0074
5%	0.0504	0.0383	0.0447	0.0081	0.0551	0.0246
10%	0.1133	0.0567	0.1166	0.0112	0.1167	0.0480
15%	0.1368	0.0774	0.1433	0.0116	0.1685	0.0567

Las tres técnicas presentan un buen comportamiento ante amplitudes de ruido pequeñas que no modifiquen de una manera tan notoria la dispersión del conjunto de datos. En contraste, ante amplitudes altas de ruido, las tres técnicas empiezan a degradar su desempeño tal como se observa en la tabla 2.12, sin embargo, es notorio el desempeño de las SVM frente al ruido en el conjunto de datos, ya que el error de validación sin ruido es de 0.0027 frente a un error de validación con un 15% de ruido de 0.0116, la cuál es una variación inferior a la encontrada en las otras 2 técnicas seleccionadas.

Así mismo, se realizan simulaciones con el conjunto de datos de Box-Jenkins

Tabla 2.13 MSE de entrenamiento y validación frente a conjuntos de datos con ruido. Sistema dinámico.

Ampl	ANN		SVM		ANFIS	
	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación
0	0.0013	0.0411	0.0202	0.0461	2.68e-6	0.3511
5%	0.0013	0.0423	0.0204	0.0510	0.5511	11.780
10%	0.0012	0.0423	0.0221	0.0626	0.5963	13.415
15%	0.0013	0.0421	0.0220	0.0963	0.5214	15.093

En el conjunto de datos correspondiente al sistema dinámico, las técnicas mejoran su desempeño frente a conjuntos de datos con ruido, sin embargo, los sistemas ANFIS presentan errores de validación mayores que en el caso de las ANN Y SVM, por

consiguiente, en los sistemas ANFIS es indispensable agregar estrategias de filtrado o corrección de datos con ruido. Las otras dos técnicas tienen un nivel aceptable de robustez frente al ruido para este caso.

2.5 Conclusiones

Después de realizar la implementación de los algoritmos correspondientes a cada una de las técnicas de aprendizaje de máquina ya definidas, es decir ANN, SVM y ANFIS, y de haber realizado pruebas con el fin de evaluar el desempeño de las mismas ante diferentes aspectos y variaciones de parámetros fundamentales en su funcionamiento se concluye lo siguiente:

- Cada una de las técnicas establecidas y descritas en este capítulo fueron implementadas utilizando los algoritmos más clásicos y de más amplio uso, en dicho proceso de implementación se encontró que el algoritmo de entrenamiento para los sistemas ANFIS representa mayor dificultad con respecto a las otras dos técnicas, principalmente por la falta de documentación disponible en el cálculo de los parámetros Takagi-Sugeno y el algoritmo de clustering substractivo. En contraste, tanto ANN como las SVM requieren de una menor cantidad de líneas de código y existe documentación suficiente para llevar a cabo su implementación. En el capítulo 8 de anexos se presenta el código implementado en cada una de las técnicas
- En general, las tres técnicas presentan un mejor desempeño cuando el conjunto de datos de entrenamiento corresponde a sistemas dinámicos donde se predice la salida futura a partir del estado anterior. En este caso la técnica de aprendizaje adquiere características de robustez al ruido, pérdida de datos e incluso una mejor capacidad de interpolación
- En cuanto a los sistemas estáticos, después de realizar las respectivas pruebas y análisis de desempeño es posible concluir que, frente a problemas en el conjunto de datos, tales como, datos fuera de tendencia y ruido el mejor desempeño lo exhibe las SVM, debido a la aplicación del margen suave en su método de solución al problema de regresión. Las ANN presentan un aceptable desempeño ante datos con ruido, pero tiene inconvenientes ante datos fuera de tendencia. Teniendo en cuenta esto, ANFIS requiere de unos datos de muy buena calidad para asegurar un desempeño adecuado.
- En cuanto a la arquitectura es posible concluir que las SVM requieren menos intervención por parte del diseñador en la afinación de parámetros debido a que solamente interviene el tipo de kernel, el cual, por su amplio uso en la literatura reportada, se recomienda el kernel gaussiano y el número de iteraciones el cual es fácilmente ajustable por prueba y error. Tanto ANFIS como ANN requieren de una

intervención más exhaustiva por parte del diseñador ya que debe aplicar estrategias como la validación cruzada en múltiples oportunidades para ajustar los parámetros adecuados en cada una de las redes.

Para resumir, ANN, SVM y ANFIS presentan un alto desempeño en la regresión de conjuntos de datos correspondientes a comportamientos no lineales, sin embargo, la decisión de uso de una por encima de las otras requiere de un análisis de los objetivos de entrenamiento de la técnica con respecto al tiempo, calidad del conjunto de datos, costo y dificultad en el proceso de entrenamiento de las mismas.

3. Generación de datos – Columna de Destilación Binaria

Son numerosos los trabajos reportados con aplicaciones de sensores inferenciales en diferentes sectores de la industria, por ejemplo, procesos de tipo químico o físico. El presente capítulo pretende realizar un acercamiento a un proceso de separación de etanol-agua, mediante una columna de destilación binaria. Se desarrolló código (Ver Anexos, Capítulo 8) para la solución de un modelo matemático con el fin de generar datos de proceso de variables como temperatura, presión, flujo y concentración de producto destilado y de fondo; el modelamiento se realizará a partir de ecuaciones de tipo fenomenológico y los datos que se generen serán utilizados para el entrenamiento de los sensores inferenciales que se desarrollarán en capítulos siguientes. Adicionalmente, se presenta una serie de experimentos con el fin de obtener datos en diferentes puntos de operación y ante diferentes perturbaciones típicas en las columnas de destilación con el objetivo de obtener un conjunto de datos que contenga una dinámica muy completa del comportamiento del proceso.

Proceso de destilación

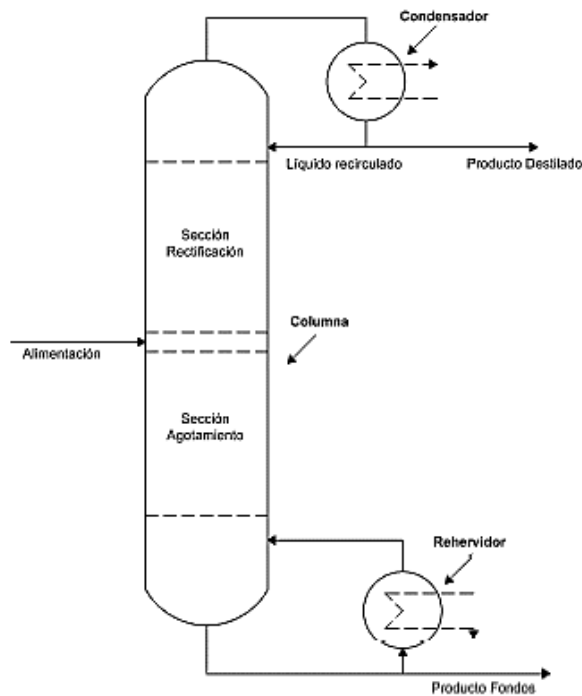
El proceso de destilación tiene como objetivo la separación de dos o más componentes haciendo uso de las características de volatilidad que tiene cada uno de los componentes del compuesto. Mediante la transferencia apropiada de energía a los componentes que se desea separar es posible cambiar su estado a gaseoso o líquido. Los componentes más volátiles o “ligeros” se llevarán a fase gaseosa y los componentes menos volátiles permanecerán en fase líquida produciéndose así una separación (Luyben, 1999).

Los procesos de destilación son ampliamente usados en diferentes sectores industriales como por ejemplo industria de alimentos, hidrocarburos, bio-combustibles, etc.; en todas estas industrias es de interés obtener concentraciones lo más puras posible, que es el índice de calidad de los productos obtenidos por medio de estos procesos, por lo tanto, los procesos de destilación son candidatos adecuados para el diseño de sensores inferenciales.

3.1 Columna de destilación binaria

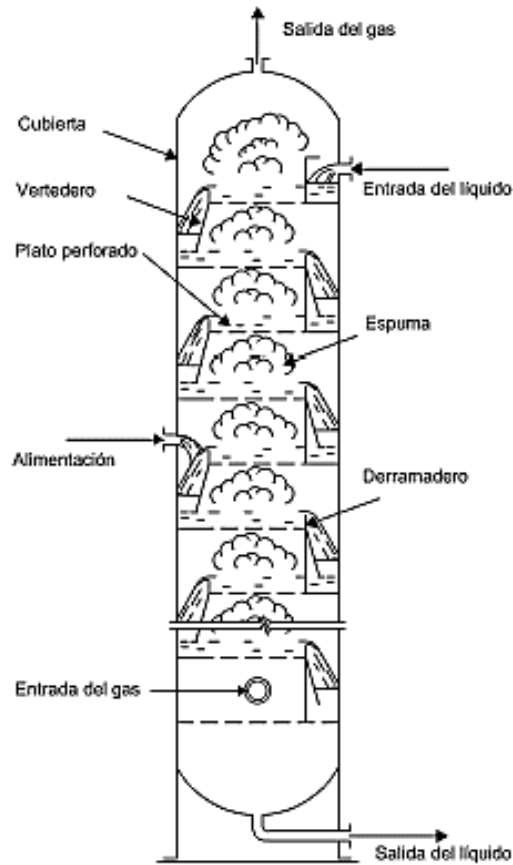
Una columna de destilación es un depósito cilíndrico con un diámetro y altura que dependen de la cantidad de material que se quiere destilar y la pureza que se quiere obtener en el producto. En su interior cuenta con una serie de reboses, comúnmente denominados platos o bandejas a través de los cuales se provoca la circulación de los productos a lo largo de la columna. El material que se desea separar se vierte a la columna a una altura determinada y la energía se agrega al sistema a través de vapor en el fondo de la columna. La figura 3.1 muestra el esquema general de una columna de destilación.

Figura 3.1 Esquema general de columna de destilación binaria. Tomado de (Alzate Ibañez, 2010)



El proceso de separación se lleva a cabo en los platos, en los cuales fluye líquido de manera descendente por acción de la gravedad, mientras que el vapor sube desde la etapa más baja hacia lo alto de la columna atravesando los platos por unos orificios y entrando en contacto con el líquido; durante este contacto se produce un intercambio termodinámico y de balance de masa que permite la separación de los componentes. La figura 3.2 muestra la manera en que se presenta el flujo de líquido y vapor dentro de la columna.

Figura 3.2 Flujo de líquido y vapor en el interior de columna de destilación. (Alzate Ibañez, 2010)



El diseño de las columnas de destilación incluye las dimensiones de diámetro y alto de la columna, número de bandejas, altura de las bandejas, distancia entre bandejas, etc. Todas estas dimensiones dependen de la complejidad de la separación que se va a llevar a cabo y de la pureza que se desea obtener en el proceso, para lo cual hay múltiples procedimientos reportados en la literatura (Luyben, 1999; Roffel & Betlem, 2007; Skogestad, 2010). Sin embargo, debido a que el modelo que se describe en este capítulo tiene el propósito de generar datos, pero no el de diseñar una columna de destilación, se reprodujeron los resultados obtenidos en (Alzate Ibañez, 2010) donde se describe con un detalle suficiente el modelado de una columna de destilación binaria para la separación de agua-etanol.

En (Alzate Ibañez, 2010), se reporta el diseño de una columna de destilación de mezcla agua-etanol con las características consignadas en la tabla 3.1.

Tabla 3.1 Características columna de destilación. Tomado de (Alzate Ibañez, 2010)

Parámetro	Valor
Flujo molar de alimentación (Kgmol/hr)	540
Fracción molar etanol en la corriente de alimentación	0.25
Temperatura (K)	355.18
Presión (bar)	1.01
Fracción de líquido	1
Flujo molar de destilado (Kgmol/hr)	158
Fracción molar de etanol destilado	0.82
Fracción molar etanol en fondo	0.01

Así mismo, presenta las características hidráulicas de la tabla 3.2 para la construcción de la columna de destilación.

Tabla 3.2 Características hidráulicas de la columna de destilación. Tomado de (Alzate Ibañez, 2010)

Parámetro	valor
Número total de platos	12
Plato de alimentación	11
Diámetro de la columna (m)	1.75
Longitud del derramadero (m)	1.225
Espaciamiento entre platos (m)	0.6
Diámetro de los orificios	0.0045
Distancia entre los centros de orificios	0.012
Altura del derramadero (m)	0.06
Área total de plato (m ²)	2.4053
Área de sección transversal vertedero (m ²)	0.21166
Área activa (m ²)	1.76

Por último, reporta las condiciones de estado estable de la columna de destilación diseñada, consignadas en la tabla 3.3.

Tabla 3.3 Condiciones de estado estable de columna de destilación. Tomado de (Alzate Ibañez, 2010)

Etapa	Fracción molar etanol	Masa retenida (Kgmol)
1	0.8060	10
2	0.7758	1.3219
3	0.7473	1.3383
4	0.7192	1.3552
5	0.6905	1.3732
6	0.6600	1.3936
7	0.6262	1.4176
8	0.5871	1.4480
9	0.5394	1.4895
10	0.4771	1.5535
11	0.3891	1.6730
12	0.2574	2.3508
13	0.1025	3.6787
14	0.0159	10

3.1.1 Modelo

En (Alzate Ibañez, 2010) se presentan 3 modelos con diferente rigurosidad, sin embargo, en el presente capítulo se describe un modelo con algunas simplificaciones que no afectan la dinámica de las concentraciones y que da una riqueza de variables suficiente para realizar la estimación por medio de cualquiera de las técnicas de aprendizaje seleccionadas.

- **Balance y Equilibrio de componentes**

Representa la dinámica que existe entre los flujos de vapor y líquido que llegan y abandonan cada uno de los platos, por lo tanto, se debe plantear para cada uno de los platos que componga la columna destilación.

Platos

Balance General

$$\frac{dM_i}{dt} = L_{i-1} - L_i + V_{i+1} - V_i \quad (3.1)$$

Balance por componente

$$\frac{d(M_i x_i)}{dt} = L_{i-1} x_{i-1} - L_i x_i + V_{i+1} y_{i+1} - V_i y_i \quad (3.2)$$

Alimentación

Balance General

$$\frac{dM_i}{dt} = L_{i-1} - L_i + V_{i+1} - V_i + F \quad (3.3)$$

Balance por componente

$$\frac{d(M_i x_i)}{dt} = L_{i-1} x_{i-1} - L_i x_i + V_{i+1} y_{i+1} - V_i y_i + F z_f \quad (3.4)$$

Condensador

Balance General

$$\frac{dM_D}{dt} = L_D + V_2 - D \quad (3.5)$$

Balance por componente

$$\frac{d(M_D x_D)}{dt} = -L_D x_D + V_2 y_2 - D x_D \quad (3.6)$$

Rehervidor*Balance General*

$$\frac{dM_B}{dt} = L_{i-1} - B - V_i \quad (3.7)$$

Balance por componente

$$\frac{d(M_B x_B)}{dt} = L_{i-1} x_{i-1} - V_B y_B - B x_B \quad (3.8)$$

Donde la nomenclatura que se sigue es la siguiente:

- V Flujo molar de vapor
- L Flujo molar de líquido
- D Flujo molar de producto destilado
- B Flujo molar de residuos (Fondo)
- X Concentración molar de líquido
- Y Concentración molar de vapor
- M Masa retenida en los platos o bandejas
- I Numero de la etapa

- **Relación de equilibrio termodinámico**

La formulación de equilibrio termodinámico se realiza por medio de la ley de Raoult con el fin de obtener el comportamiento de la concentración de vapor (y) y la temperatura en cada una de las etapas de la columna de destilación, dentro de la ingeniería química se le denomina perfil de temperatura.

En equilibrio, la temperatura del líquido, su composición y presión están relacionadas, por lo tanto, es posible expresar una en términos de la otra. Teniendo en cuenta esto se aplican las ecuaciones de Dalton Raoult y Antoine (Luyben, 1999; Roffel & Betlem, 2007) para expresar la temperatura y la composición de vapor.

La ecuación de Raoult plantea que la composición del líquido depende de la presión de vapor y la presión del componente puro:

$$P_j = x_j \gamma_j P_j^o \quad j = A, B \quad (3.9)$$

Así mismo, la ecuación de Dalton relaciona la presión por componente y la presión total con la composición de vapor de la siguiente manera:

$$y_j = \frac{P_j}{P} \quad j = A, B \quad (3.10)$$

En el mismo sentido, la ecuación de Antoine define la presión de componente puro dependiente de la presión total y la temperatura:

$$P_j^o = 10^{A_j - \frac{B_j}{C_j - T}} \quad j = A, B \quad (3.11)$$

Por último, utilizando la ecuación de Van Laar, se calcula el parámetro γ para completar la ecuación de Raoult:

$$\ln(\gamma_A) = A_{AB} * \left(\frac{A_{BA} * x_B}{A_{AB} * x_A + A_{BA} * x_B} \right)^2 \quad (3.12)$$

$$\ln(\gamma_B) = A_{BA} * \left(\frac{A_{AB} * x_A}{A_{AB} * x_A + A_{BA} * x_B} \right)^2 \quad (3.13)$$

Las ecuaciones (3.9), (3.10) y (3.11) aplican para cualquiera de los dos componentes de la mezcla (A o B) y los coeficientes A_{AB} y A_{BA} corresponden a los coeficientes de interacción entre los componentes los cuales son constantes tomadas de (Perry John H, 1997).

- **Consideraciones y simplificaciones adicionales**

Para completar el modelo y poder implementarlo en código Octave/Matlab, es necesario agregar las siguientes consideraciones con el fin de simplificar algunos aspectos de la dinámica del proceso (Alzate Ibañez, 2010):

1. Todos los flujos de entrada y salida se encuentran en fase líquida
2. Alimentación en un único plato
3. No hay pérdidas de calor, es decir que la columna es adiabática.
4. El condensador es total, por lo tanto, la composición del vapor que abandona la columna por cabeza será la misma que la de la corriente de reflujo y destilado
5. El líquido acumulado en cada plato es incompresible y se encuentra como una mezcla perfecta (la composición es la misma en todos los puntos).
6. Las fases líquida y vapor que abandonan el plato se encuentran en equilibrio térmico
7. Se considera que la presión a lo largo de la columna es constante
8. La acumulación de materia en fase vapor es nula.
9. Cada plato es considerado con una eficiencia del 100%
10. La fase vapor y la fase líquida se consideran no ideales

11. Se desprecia el tiempo muerto en la corriente de vapor que va desde el último plato de la columna hasta el condensador, y también en la corriente de retorno del reflujo a la cabeza de la torre.
12. Los flujos de destilado y fondo se consideran constantes
13. Los flujos de Vapor y líquido son constantes a lo largo de la columna
14. Las variaciones de masa con respecto al tiempo en el condensador y en el rehervidor son iguales a cero.
15. No se considera acumulación de vapor a lo largo del sistema.
16. La variación de masa con respecto al tiempo en los platos se considera cero
17. No se considera balance de energía

3.1.2 Implementación en código

Se realiza la implementación mediante lenguaje Octave/Matlab de las ecuaciones descritas anteriormente de la siguiente manera, teniendo en cuenta que en la tabla 3.1 se establece que la columna está conformada por 12 platos, 1 condensador y 1 rehervidor, para un total de 14 etapas. Del modelo se obtuvieron 32 ecuaciones distribuidas de la siguiente manera:

Tabla 3.4 Complejidad del modelo

	Ecuaciones Diferenciales	Ecuaciones Algebraicas	Relaciones Termodinámicas	Total
Condensador	1	1	1	32
Platos	11	1	11	
Alimentación	1	1	1	
Rehervidor	1	1	1	
Total	14	4	14	

Para la implementación se tuvo en cuenta las pautas de cálculo que indican (Luyben, 1999; Skogestad, 2010) en sus respectivos modelos, adicionalmente se utilizó el método de integración ode15s, disponible tanto en Matlab como en Octave.

3.1.3 Resultados Modelo

Después de realizar la implementación del modelo matemático se obtuvieron resultados muy aproximados a los reportados en el documento de referencia (Alzate Ibañez, 2010), los cuales se presentan a continuación.

Por un lado, el perfil de temperatura que se obtuvo tiene una diferencia menor a 1 K lo cual es una muy buena aproximación tal como se muestra en la tabla 3.5 y la figura 3.3.

Figura 3.3 Perfil de temperatura obtenido

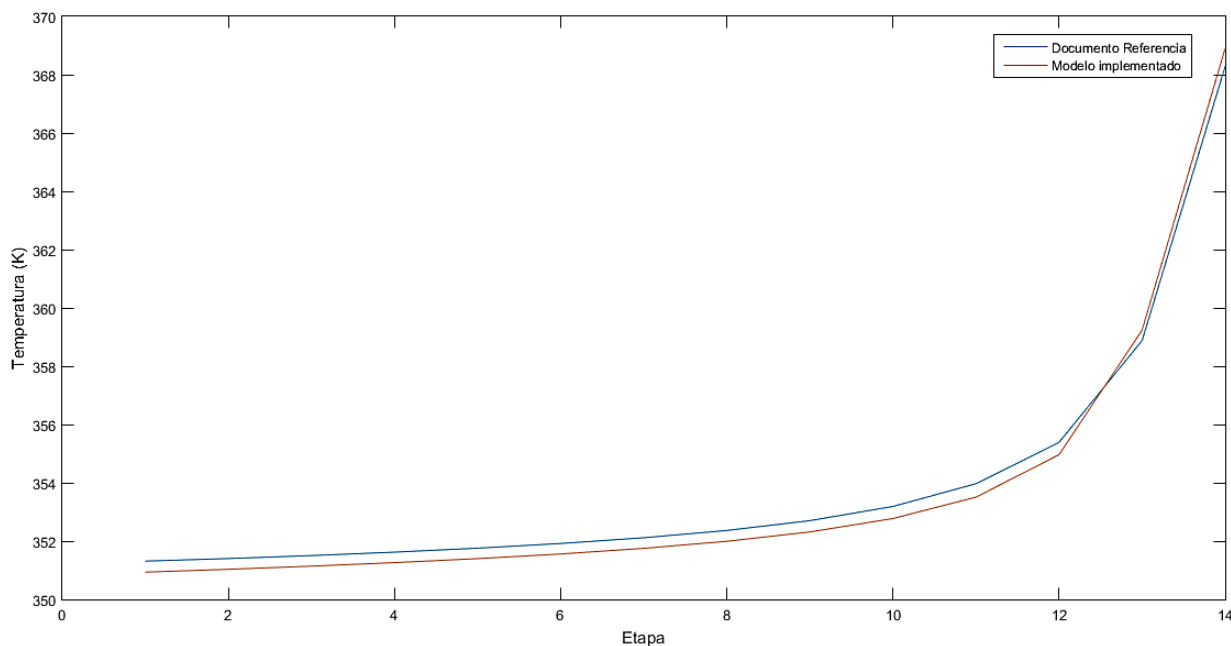


Tabla 3.5 Resultados perfil de temperatura modelo columna de destilación

Etapa	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Temp referencia	351.3	351.4	351.5	351.6	351.7	351.9	352.1	352.6	352.7	353.1	353.9	355.3	358.8	368.2
Temp simulación	350.9	351.0	351.1	351.2	351.3	351.5	351.7	351.9	352.3	352.7	353.5	354.9	359.2	368.9

Así mismo, la composición de destilado y fondo resultó muy cercana con diferencia de 0.001, tal como se aprecia en la tabla 3.6, con lo cual se concluye que los resultados fueron reproducidos satisfactoriamente y el modelo está en capacidad de generar datos de flujo, temperatura, presión y composición a lo largo de la columna de destilación.

Figura 3.4 a) Concentración molar de etanol en cima. b) Concentración molar de etanol en fondo

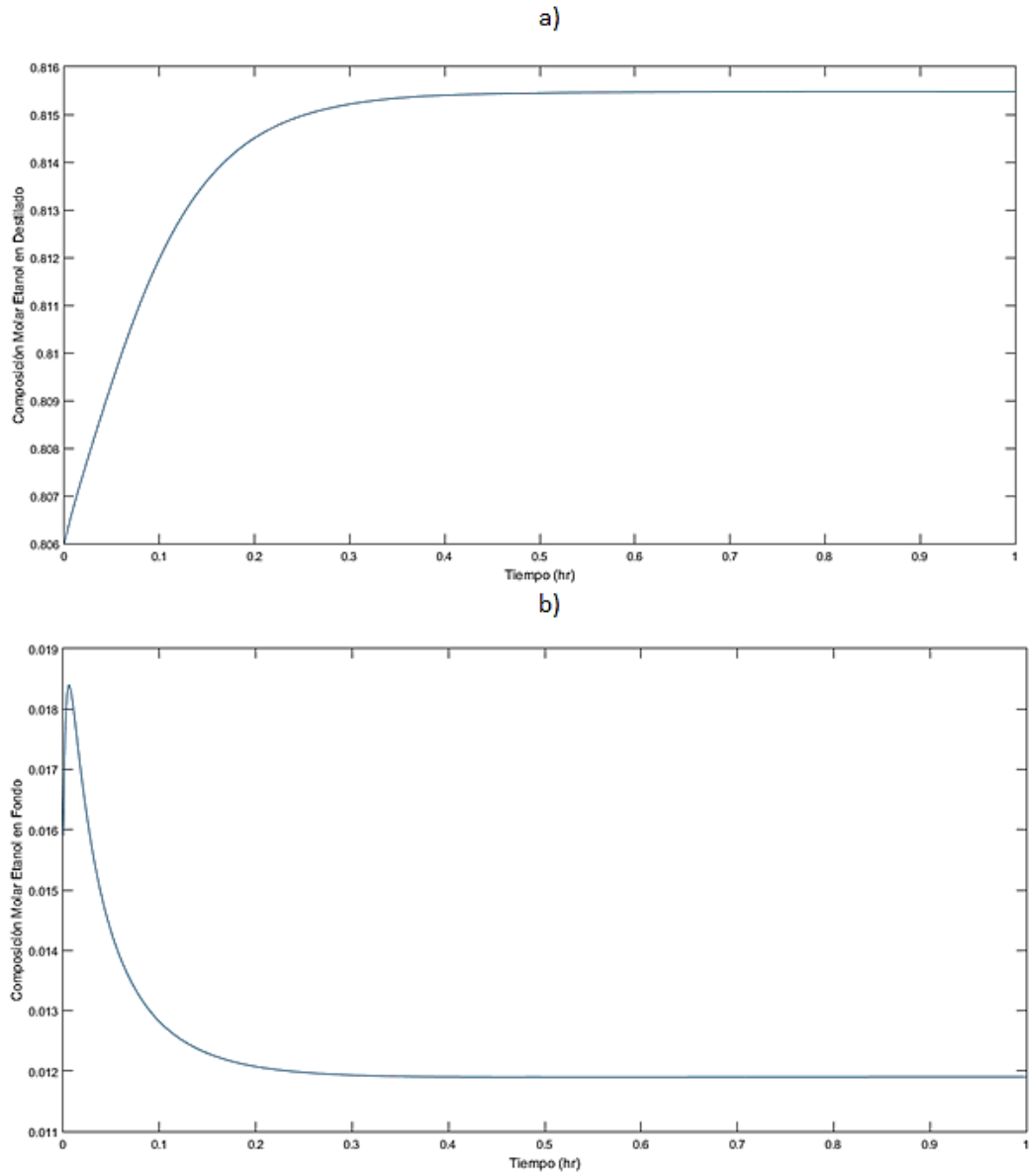


Tabla 3.6 Valores de estado estable para concentración molar de etanol

	Implementación	Documento referencia
$x_{condensador}$	0.8155	0.817
x_{fondo}	0.0119	0.011

Para concluir, con el modelo implementado y todas las consideraciones tenidas en cuenta en (Alzate Ibañez, 2010) es posible obtener datos que capturen una dinámica suficientemente rica en el comportamiento de la composición de los productos de destilado y fondos para realizar su estimación por medio de cualquiera de las técnicas de aprendizaje de máquina seleccionadas y descritas en el capítulo 2.

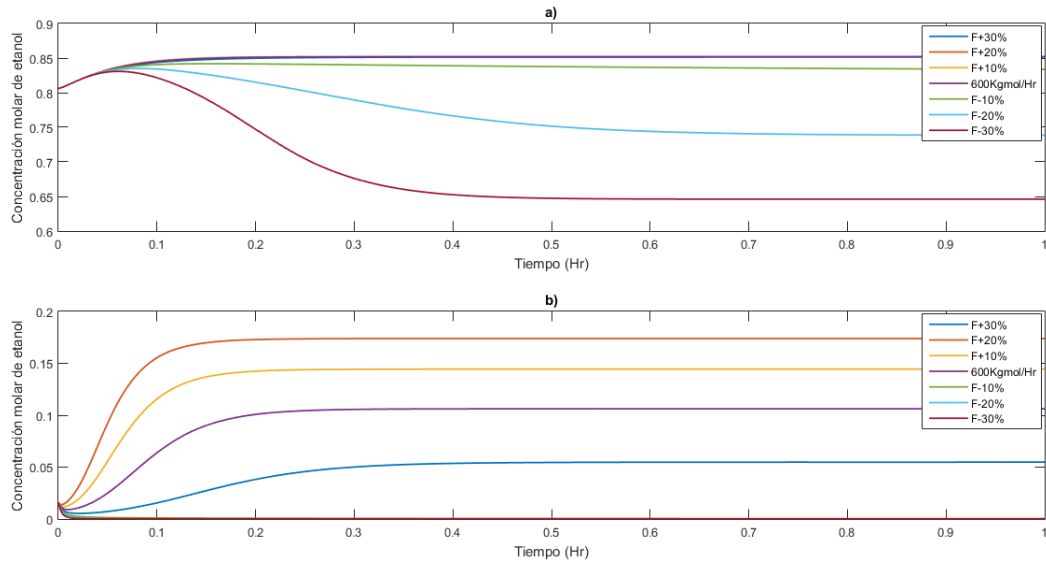
3.2 Generación de conjunto de datos

A continuación, se desarrollan múltiples experimentos para obtener un conjunto de datos que contengan una dinámica suficientemente completa, para obtener una estimación de la concentración de etanol en diferentes escenarios y ante distintas perturbaciones que provocan cambios en la concentración de etanol y por lo tanto influyen en la calidad del proceso de destilación.

Desde el punto de vista de funcionamiento de la columna de destilación binaria, es posible manipular la relación de reflujo y el flujo destilado con el fin de controlar la concentración del producto destilado, por lo tanto, este es uno de los parámetros a variar para obtener diferentes condiciones en la columna de destilación. Así mismo, las columnas de destilación binarias se afectan por perturbaciones tales como variaciones en el flujo de alimentación y composición de alimentación, las cuales afectan la composición que se obtiene en el producto destilado. Teniendo en cuenta lo anterior, en la figura 3.4 se presenta el comportamiento de la composición de destilado, composición de fondo y perfil de temperatura para una composición de alimentación $z_f = 0.4$, $F = 600 \text{ Kgmol/hr}$, $D = 260 \text{ Kgmol/hr}$ y $R = 3$.

En la figura 3.5 se aprecia la concentración de destilado y de fondo, ante variaciones desde 10% hasta 30% en el flujo molar de alimentación. Se aprecia que ante flujos de alimentación bajos la concentración del producto destilado cae, debido a que el flujo molar de destilado se mantiene constante por lo tanto el flujo de reflujo dentro de la columna disminuye. Así mismo, flujos muy altos provocan un aumento en la concentración de etanol en el flujo de fondo, situación que no es deseable ya que se está perdiendo una porción de etanol en los residuos. Así mismo, en la figura 3.6 se observa un comportamiento similar ante variaciones en la concentración de alimentación.

Figura 3.5 Variación paramétrica de concentración molar de etanol a través de flujo molar de alimentación. a) Cima, b) Fondo.



En las figuras 3.7 y 3.8 se evidencia la variabilidad que tiene la concentración molar de etanol ante cambios en la relación de reflujo (R) y flujo molar de destilado (D) respectivamente.

El comportamiento presentado en la figura 3.7, evidencia relación directa entre concentración molar de etanol y relación de reflujo; esto coincide con las situaciones reales de planta donde la calidad de producto se controla con la relación de reflujo.

Figura 3.6 Variación paramétrica de concentración molar de etanol a través de concentración molar de alimentación. a) Cima, b) Fondo.

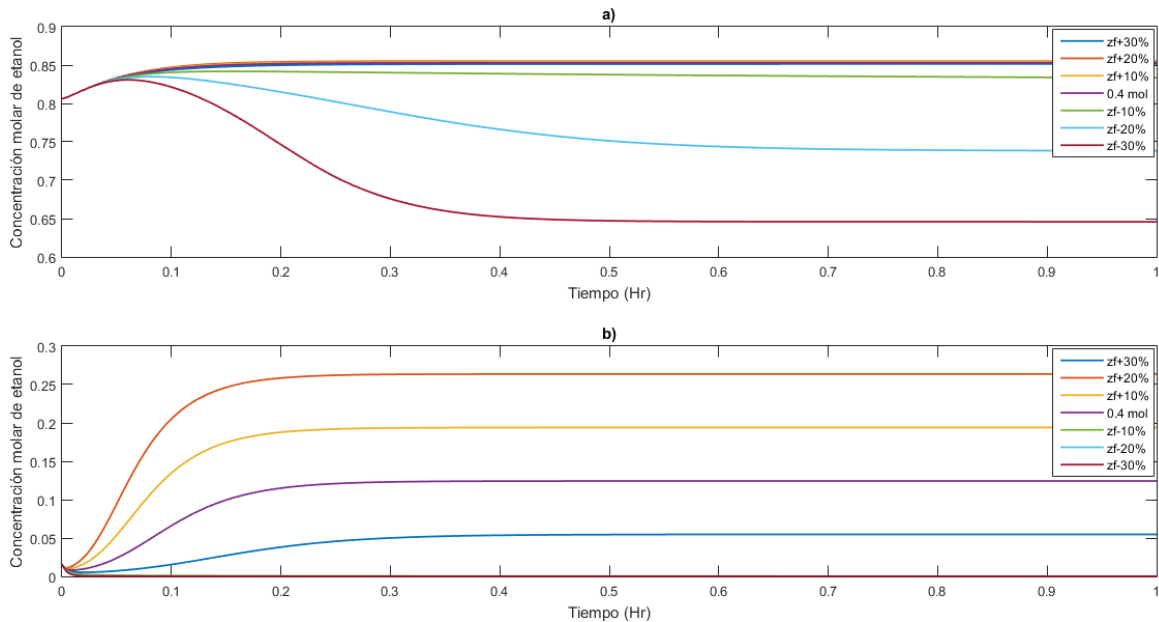
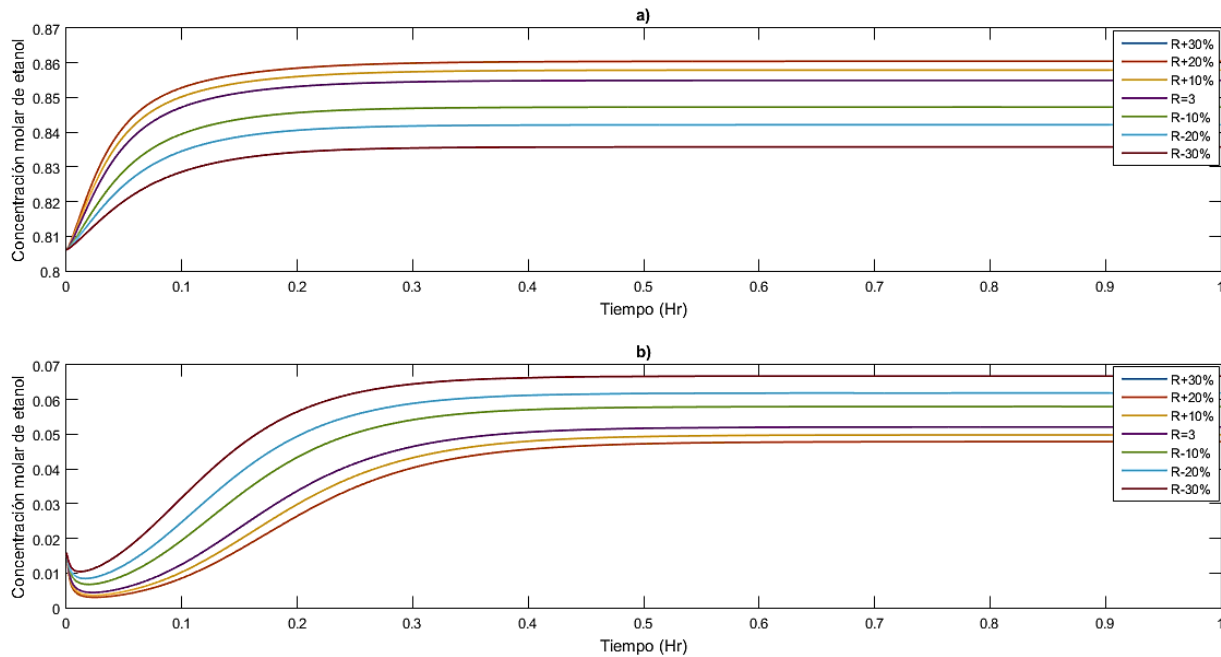
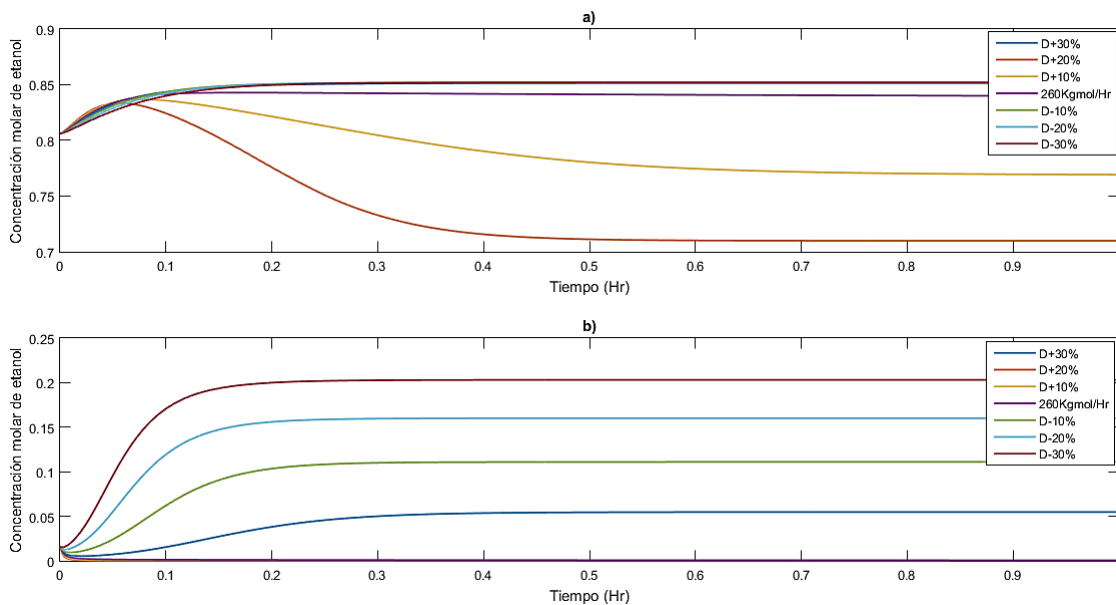


Figura 3.7 Variación paramétrica de concentración molar de etanol a través de relación de reflujo (R). a) Cima, b) Fondo.



Así mismo, en la figura 3.8 se presentan variaciones significativas de la concentración molar de etanol frente a cambios en el flujo de destilado, esto se explica en la situación presentada arriba ya que el flujo de destilado y la relación de reflujo están relacionados matemáticamente.

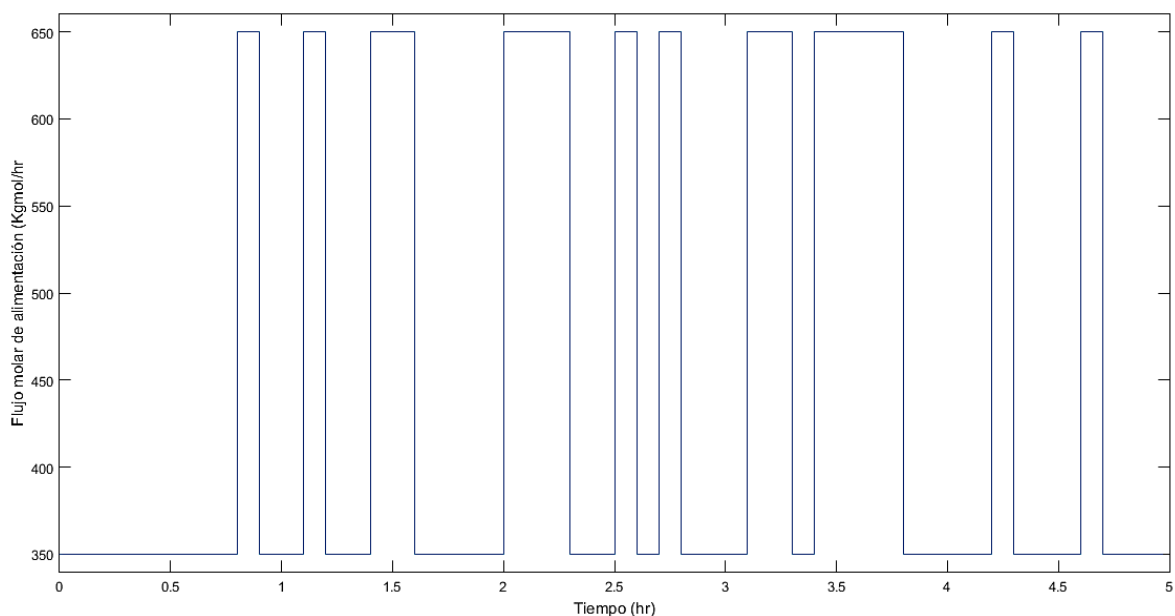
Figura 3.8 Variación paramétrica de concentración molar de etanol a través de flujo de destilado (D). (arriba) Cima, (abajo) Fondo.



En las figuras anteriores se evidencio que el modelo responde satisfactoriamente ante cambios en los parámetros de alimentación y destilado y la calidad del producto (etanol) se afecta ante las variaciones realizadas en dichos parámetros. Sin embargo, para el entrenamiento de las técnicas de aprendizaje de máquina es deseable obtener series de tiempo de cada una de las variables, donde se recolecte la mayor cantidad de información sobre el comportamiento del proceso; en otras palabras, es necesario establecer un procedimiento para obtener un conjunto de datos con las características antes descritas.

Para este objetivo, se utilizará una de las herramientas más ampliamente reportadas en el ámbito de la identificación y modelamiento de sistemas, denominada secuencias binarias pseudo-aleatorias (PRBS³⁴), las cuales consisten en un tren de pulsos con amplitud determinada y anchos de pulso aleatorios que al aplicarlas al sistema se obtiene una operación en múltiples puntos y una recolección de información abundante a cerca de la dinámica del sistema (Ljung, 1987). En la figura 3.9 se observa una de las secuencias utilizadas para la variación del flujo molar de alimentación de la columna de destilación.

Figura 3.9 Secuencia pseudo-aleatoria aleatoria (PRBS)



- **Serie de tiempo**

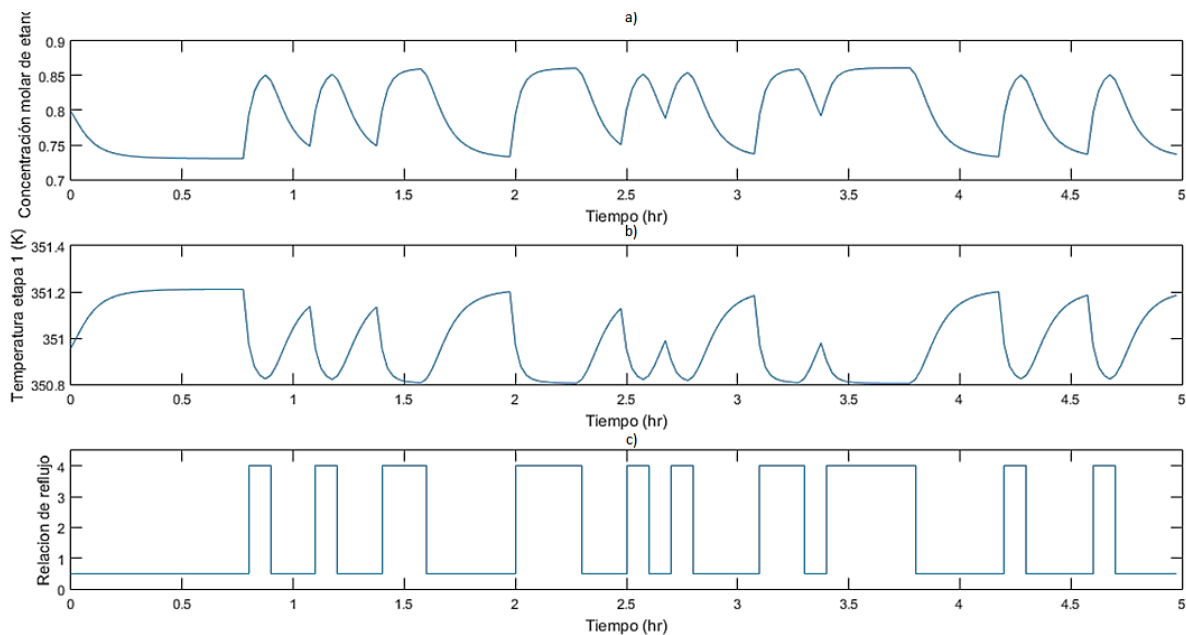
Como se dijo anteriormente, la generación de las series de tiempo se realizó mediante la variación de los parámetros F, D y R a través de secuencias PRBS, en las cuales se tuvo en cuenta lo siguiente para que los datos obtenidos representen con suficiencia la dinámica del proceso de destilación de etanol:

³⁴ PRBS Pseudo-Random Binary Sequence

- La amplitud de la secuencia se debe seleccionar de acuerdo con el contexto de la variable y con el rango de variación del proceso. Por ejemplo, en la figura 3.9 se realizan variaciones de flujo de alimentación entre 350 y 650 Kgmol/hr, lo cual está dentro del rango de operación de la columna de destilación.
- El tiempo de bit debe ser seleccionado adecuadamente con respecto al número de muestras que se desea obtener y el tiempo total de la secuencia. Para el caso de la columna de destilación se establecieron 4000 bits y un tiempo de bit de 360 segundos.
- Por último, es necesario garantizar que, en las zonas de pulso más ancho, el proceso llegue a estado estable, por lo tanto, el tiempo de bit y el tiempo de estado estabilización de la variable se relacionan. Para el caso de la columna de destilación se tomó el tiempo de estabilización como 0.4 horas.

La generación del conjunto de datos se realizó aplicando las secuencias a cada una de las 3 variables indicadas (F, D, R), primero de manera individual (una sola secuencia a la vez) y luego de manera combinada por parejas, considerando todas las combinaciones. En la figura 3.10 se presenta un segmento de la serie de tiempo generada para la concentración de etanol destilado frente a la variación de la relación de reflujo.

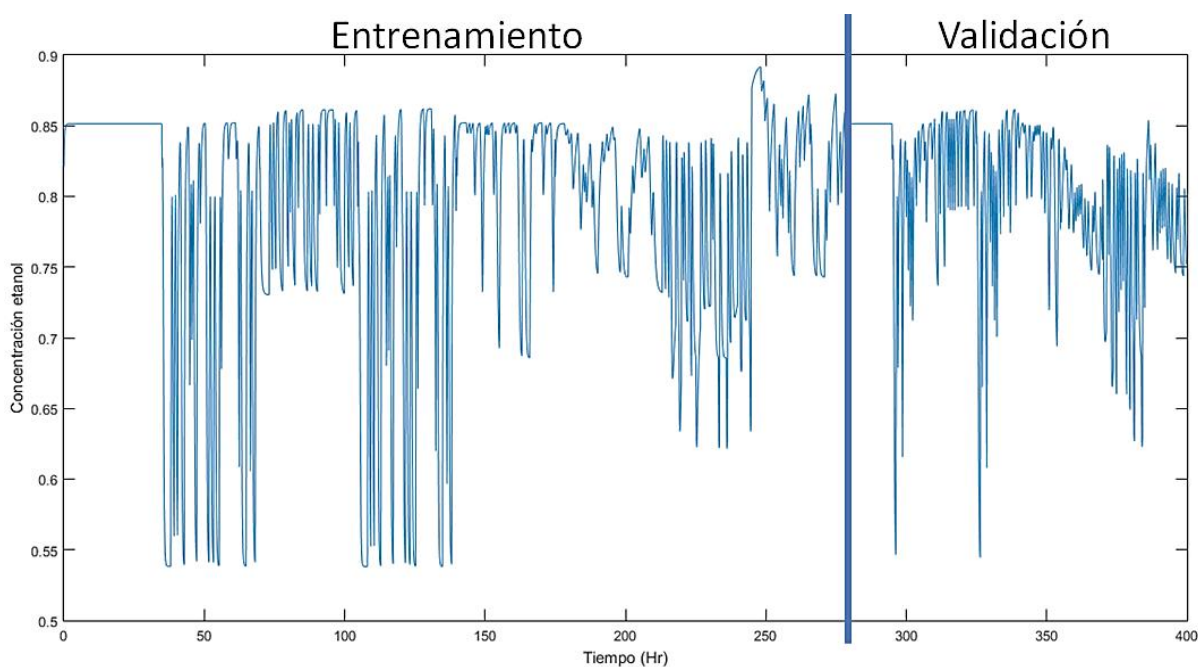
Figura 3.10 Fragmento de serie de tiempo generada a) concentración molar etanol, b) Temperatura etapa 1 c) Relación de reflujo variada a través de PRBS



En las series de tiempo presentadas en la figura 3.10, se observa el funcionamiento del proceso de destilación en diferentes tipos de operación y un rango de variación que resulta útil para comprender el comportamiento del proceso. En la figura 3.11 se presenta la serie de tiempo de la composición de etanol resultante de los experimentos desarrollados con

el modelo matemático de la columna de destilación y los segmentos utilizados para el entrenamiento y validación de las técnicas de aprendizaje de máquina, los cuales corresponden al 70% de los datos para entrenamiento y el 30% de los datos para llevar a cabo la validación del modelo. Teniendo en cuenta los datos anteriores, la serie de tiempo corresponde a 400 horas de funcionamiento con un tiempo de muestreo de 0.1 horas se obtuvieron 4000 muestras que cubren las variaciones de las 3 variables realizadas en la entrada.

Figura 3.11 Serie completa de tiempo para la concentración de etanol del modelo matemático de la columna de destilación binaria



3.3 Conclusiones

Después de realizar la implementación del modelo matemático del proceso de destilación de una mezcla agua-etanol y de realizar múltiples experimentos para la generación de series de tiempo de cada una de las variables de proceso que juntas constituyen el conjunto de datos de entrenamiento y validación, se llegó a las siguientes conclusiones:

- Desde la perspectiva de generación de datos sintéticos de procesos industriales, el proceso de una columna de destilación binaria es altamente accesible a nivel de modelo matemático, y aunque es necesario establecer ciertos conceptos de áreas de la ingeniería química, el proceso es entendible con una formación básica de ingeniería, lo cual hace del proceso de destilación un muy buen candidato para la generación de datos sintéticos de proceso.

- A partir del modelo matemático y variaciones paramétricas del proceso, es posible establecer cierto nivel de conocimiento acerca del proceso, como dependencia de variables, instrumentación que tiene una columna de destilación real, entre otros; sin embargo, no es posible dejar de lado el conocimiento experto que concatena el comportamiento matemático con los fenómenos físicos y químicos que ocurren dentro de la columna de destilación, lo cual es importante para la interpretación del modelo y la validación del mismo.
- Es necesario establecer un conjunto de datos que contenga no solamente datos suficientes sino también información suficiente que permita la inferencia de las variables de interés, esto es posible lograrlo mediante el uso de las secuencias PRBS que llevarán a las variables a operar en un rango determinado, emulando de alguna manera las condiciones típicas de una columna de destilación.
- Aunque lo deseable en el diseño de sensores inferenciales es contar con conjuntos de datos obtenidos directamente del sistema de instrumentación de la planta de proceso, los datos sintéticos obtenidos de un modelo matemático y la capacidad de variabilidad que ofrece un sistema de simulación son convenientes en el establecimiento de una metodología para el diseño y la implementación de sensores inferenciales.

Para resumir, se implementó el modelo de una columna de destilación binaria para una mezcla de agua-etanol a partir de sus ecuaciones fenomenológicas. En el mismo sentido, a través del modelo implementado se obtuvo un conjunto de datos de todas las variables de proceso mediante la variación de los parámetros que influye en la calidad del producto desde el proceso o como perturbaciones.

4.Pre - procesamiento de Datos

Los conjuntos de datos obtenidos de entornos industriales tales como bases de datos o DCS (Distributed Control System) presentan características tales como ruido, datos fuera de tendencia y pérdida de datos que afectarán el desempeño de cualquier técnica de regresión que se utilice para realizar un modelo o inferir variables; por lo cual, es necesario la implementación de estrategias para obtener conjuntos de datos convenientes para el entrenamiento de una técnica de aprendizaje de máquina. Así mismo, los procesos industriales realizan mediciones de una gran cantidad de variables, las cuales pueden ser de interés para la implementación de sistemas de control en el proceso, sin embargo, desde el punto de vista del comportamiento del proceso no todas estas variables son relevantes y por ende es necesario establecer procedimientos que permitan la selección o reducción de todas las variables medidas a un sub – conjunto que sea representativo para la dinámica del proceso que se está modelando o la variable que se está estimando (Guyon, Elisseeff, & De, 2003). En suma, el presente capítulo establece un análisis comparativo de diferentes técnicas y estrategias para el pre – procesamiento de los conjuntos de datos que se utilicen en el entrenamiento de técnicas de aprendizaje de máquina.

4.1 Datos Fuera de tendencia “Outliers”

Como se dijo en la subsección 2.4.3, los datos fuera de tendencia son característicos de datos provenientes de procesos industriales y se deben a condiciones ambientales alrededor de la instrumentación del proceso y a fallas esporádicas de los sensores. Como se estableció en el capítulo 2, este tipo de datos afectan drásticamente el desempeño de las técnicas de aprendizaje de máquinas seleccionadas para el presente trabajo, por lo cual es necesario implementar estrategias de filtrado que permitan identificar cuales datos no corresponden con la tendencia del comportamiento en las variables.

4.1.1 Detección de Datos fuera de Tendencia

Existen múltiples técnicas para la detección y corrección de datos fuera de tendencia, todas basadas en formas de filtrar los datos para identificar cuales datos están desviados de la tendencia de los mismos y de esta manera reemplazarlos por algún valor representativo del conjunto de datos, típicamente la media o la mediana. No obstante, todas las técnicas

para la detección de datos fuera de tendencia o “*outliers*”, tienen un margen de error en la detección, el cual se presenta en 2 situaciones:

- **Enmascaramiento de datos (Masking)**

Se presenta cuando la técnica de filtrado identifica algunos datos que están fuera de la tendencia del conjunto como parte del mismo (Ben-gal, 2005).

- **Inundación de datos (Swamping)**

Se presenta cuando la técnica de filtrado identifica como datos fuera de tendencia algunos que son parte del conjunto de datos (Ben-gal, 2005).

A continuación, se realizará una breve descripción de algunas de las técnicas utilizadas con mayor frecuencia en tareas de detección de datos fuera de tendencia, las cuales serán evaluadas en siguientes subsecciones por medio del desempeño de las técnicas de aprendizaje de máquina seleccionadas y descritas en el capítulo 2.

4.1.2 Técnicas de Filtrado

- **Regla de $3\text{-}\sigma$**

Asumiendo que la variable que se va a filtrar se comporta dentro de una distribución Gaussiana con media μ y desviación estándar σ , todos aquellos datos que se encuentren por fuera del rango $[\mu - 3\sigma, \mu + 3\sigma]$ son considerados datos fuera de tendencia (F. A. A. de Souza, 2014). Este criterio está basado en la idea, que para una distribución normal la probabilidad de encontrar datos con una desviación mayor a 3σ con respecto a la media μ es aproximadamente del 0.3% (Pearson, 2002).

- **Identificador de Hampel**

El identificador de Hampel es una extensión de la regla $3\text{-}\sigma$, sin embargo, reemplaza la media μ y la desviación estándar σ , debido a su sensibilidad ante datos fuera de tendencia siendo susceptible a errores de “*masking*”. En contraste el identificador de Hampel utiliza como medidas de tendencia la mediana y la desviación absoluta de la mediana (MAD³⁵), las cuales son más “resistentes” a los datos fuera de tendencia (Pearson, 2001). El criterio de detección consiste en lo siguiente:

³⁵ MAD por sus siglas en ingles Median Absolute Deviation

Para un conjunto de muestras de una variable con mediana x^* la desviación absoluta de la mediana es la siguiente (Pearson, 2002):

$$MAD = 1.4826 * mediana\{|x_k - x^*|\} \quad (4.1)$$

Donde x_k son los datos del conjunto con $k=1, 2, \dots, N$. El factor 1.4826 es seleccionado para que la MAD sea igual a la desviación estándar σ con una distribución gaussiana.

Son considerados datos fuera de tendencia todos aquellos que cumplan con:

$$x_k \pm x^* > t * MAD$$

Donde t es el criterio que define la amplitud del filtro.

▪ Filtro de Hampel

El filtro de Hampel consiste en un algoritmo que divide el conjunto de muestras de la variable en pequeños grupos de datos a los cuales aplica el identificador de Hampel; al tamaño de los subconjuntos de datos se les denomina ventana, de tal manera, que a cada dato que no cumpla con el criterio se le reemplazará por la mediana de la ventana, permitiendo de esta manera una detección por sectores lo cual resulta útil en conjuntos de datos que presentan variaciones en sus muestras por rangos de tiempo (Ben-gal, 2005).

Es así como el filtro de Hampel, posee 2 parámetros de ajuste, por un lado, el ancho de la ventana que como se dijo anteriormente define la cantidad de datos que se analizan en cada ventana, se representa como K que corresponde a la mitad de la ventana y por otro lado t que es heredado del identificador Hampel, el cual define la amplitud del filtro; en (Pearson, 2002) recomiendan $3 < K < 5$ y $2 < t < 5$, por lo tanto, se utilizan entre 7 y 11 datos para analizar en cada ventana.

4.1.3 Comparativo de Técnicas de Filtrado

En el capítulo 2, se realizó un estudio comparativo, sobre la influencia de los datos fuera de tendencia en la predicción de cada una de las técnicas seleccionadas, en forma general se encontró que todas presentan degradación en el error de predicción cuando los conjuntos de datos de entrenamiento tienen datos fuera de tendencia. Por lo tanto, es necesario incluir una etapa de filtrado de los datos con el fin de detectar y dar el adecuado tratamiento a los datos que presentan altas desviaciones con respecto a la dinámica normal de los mismos. A continuación, se presenta un estudio comparativo del comportamiento de cada una de las técnicas de aprendizaje de máquina seleccionadas, frente a datos que han sido sometidos al tratamiento de “outliers” mediante las técnicas descritas anteriormente.

Para esta sección se utilizó un conjunto de datos generados sintéticamente, a partir de un modelo matemático de una columna de destilación binaria para la separación de agua-etanol, que se describe en el capítulo anterior, de la cual se pueden obtener datos de flujo, presión y temperatura en sus diferentes etapas. Para la evaluación de desempeño, se agregaron datos fuera de tendencia al conjunto obtenido, en diferentes cantidades y con diferentes amplitudes para analizarlos con cada una de las técnicas de detección de datos fuera de tendencia.

Desempeño de técnicas de filtrado

A partir de los datos generados con el modelo matemático del proceso de destilación de una mezcla agua-etanol, se agregan datos fuera de tendencia de manera aleatoria, con cantidad y amplitud definidas para identificar la efectividad de cada técnica de filtrado en la detección de los datos fuera de tendencia en términos de cantidad. El conjunto utilizado corresponde a 4000 muestras y 20 variables correspondientes a temperatura y flujo en diferentes etapas de la columna de destilación.

En la tabla 4.1 Se presenta la cantidad de datos fuera de tendencia detectados por cada una de las técnicas, se evidencia que al usar la técnica 3-sigma tiende a ocurrir errores de “*masking*” y que su mejor comportamiento se obtiene cuando hay pocos datos fuera de tendencia y con amplitudes altas. En el mismo sentido al analizar las técnicas de MAD y filtro de Hampel es posible afirmar que presentan su mejor comportamiento ante conjuntos de datos altamente contaminados con “*outliers*” con una pequeña mejora en el desempeño; estas técnicas se caracterizan por el predominio de error de “*swamping*”.

Adicionalmente, para las pruebas se configuraron los parámetros del filtro de Hampel con $K=5$ y $t=4$ y se obtuvo un buen resultado para datos con más del 15% de “*outliers*”, no obstante, para conjuntos con un número menor de datos fuera de tendencia se puede configurar nuevamente con unos parámetros K y t adecuados para evitar el error de “*swamping*” que se presenta; en contraste las otras dos técnicas no permiten esta flexibilidad en el nivel de filtrado.

Tabla 4.1 Detección de datos fuera de tendencia con las técnicas 3-sigma, MAD y filtro de Hampel

cantidad	800	1600	4000	12000	20000	24000	
	1%	2%	5%	15%	25%	30%	
Ampl							
2	1403	2646	2151	1077	3719	1912	3 - σ
2.5	1956	1314	2777	2180	1111	3815	
3	3885	1946	1316	2748	2127	1094	
3.5	1109	3819	1983	1306	2709	2074	
4	2100	1113	3788	1918	1287	2721	
2	19535	24944	23920	23547	22024	20205	MAD
2.5	20707	19520	24918	24076	23218	22038	
3	21983	20203	19539	25020	24346	23511	
3.5	22828	22011	20216	19562	25497	24091	
4	23791	23345	21995	20202	19541	25488	
2	11795	19541	19331	16951	13374	12157	Hampel
2.5	12213	11794	19683	19181	17035	13446	
3	13440	12247	11791	19829	19175	17122	
3.5	16765	13384	12174	11837	19706	19386	
4	19107	16915	13352	12124	11796	19558	

En la tabla 4.1 se presentan los resultados las pruebas realizadas a cada una de las técnicas de filtrado consideradas, y se observa claramente los errores de masking y swamping que predominan en las reglas de 3sigma y MAD respectivamente. En el mismo sentido, la tabla muestra que la técnica más versátil en cuanto a la cantidad y amplitud de datos fuera de tendencia es el filtro Hampel, esto debido a que los parámetros se pueden ajustar para obtener un filtrado adecuado dependiendo de la naturaleza del conjunto de datos

Así mismo, en la tabla 4.2 se presenta el error promedio entre series de datos del conjunto original y las mismas series filtradas después de agregarles datos fuera de tendencia; entre los resultados obtenidos se confirma lo advertido anteriormente, que el filtro Hampel tiene

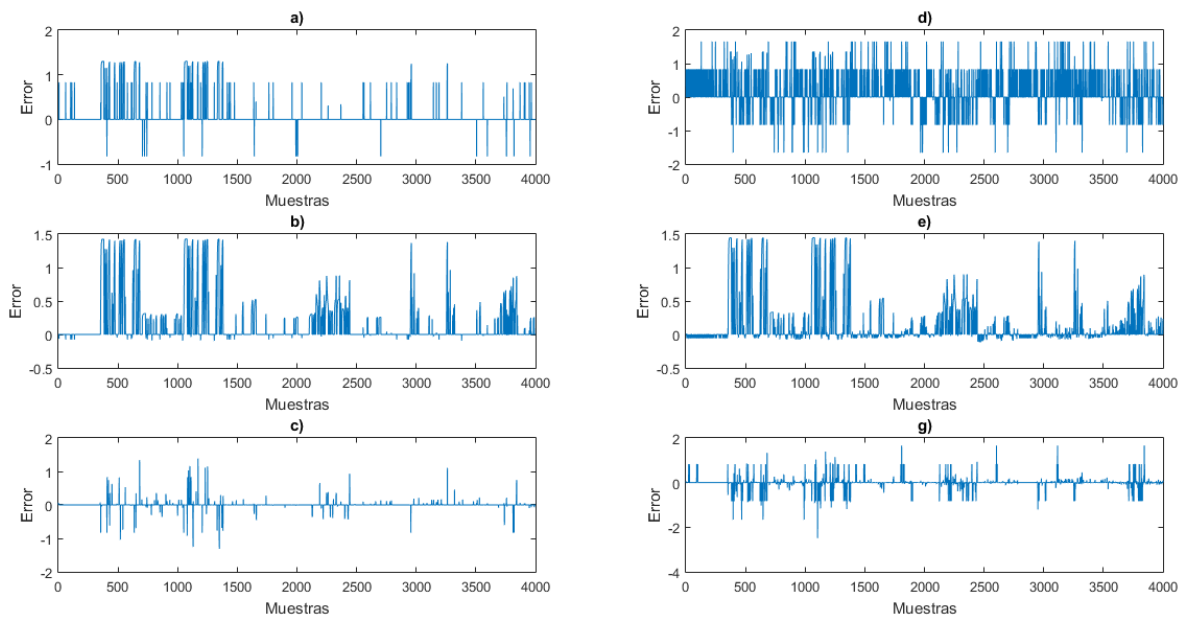
un desempeño superior a la regla $3\text{-}\sigma$ y a la media absoluta MAD, debido a que el filtro Hampel permite ajustar sus parámetros de ancho de ventana y altura de filtro para mejorar la detección de datos fuera de tendencia.

Tabla 4.2 Error promedio (ME) con diferentes filtros

Ampl/ Cantidad	$3\text{-}\sigma$	MAD	Hampel
2.5 σ/2%	0.0588	0.1346	0.0012
3.5 σ/2%	0.0446	0.1352	0.0022
2.5 σ/15%	0.0535	0.1373	0.0016
3.5 σ/15%	0.0698	0.1386	0.0029
2.5 σ/25%	0.0927	0.1282	0.0051
3.5 σ/25%	0.0902	0.1281	0.0019

De la tabla 4.2 también se observa que el filtro MAD presenta mayor error con respecto a $3\text{-}\sigma$, sin embargo, al analizar los datos se detecta que mejora su comportamiento al aumentar la cantidad de “outliers” y adicionalmente presenta una variación muy baja ante “outliers” de amplitud alta; lo cual es deseable en series de datos con datos fuera de tendencia de diferentes amplitudes como los que se presentan en el caso de los sensores inferenciales.

Figura 4.1 Error de datos filtrados: a) regla $3\text{-}\sigma$ con 2% “outliers”, b) MAD con 2% “outliers”, c) Filtro Hampel con 2% “outliers”, d) regla $3\text{-}\sigma$ con 25% “outliers”, e) MAD con 25% “outliers”, f) Filtro Hampel con 25% “outliers”

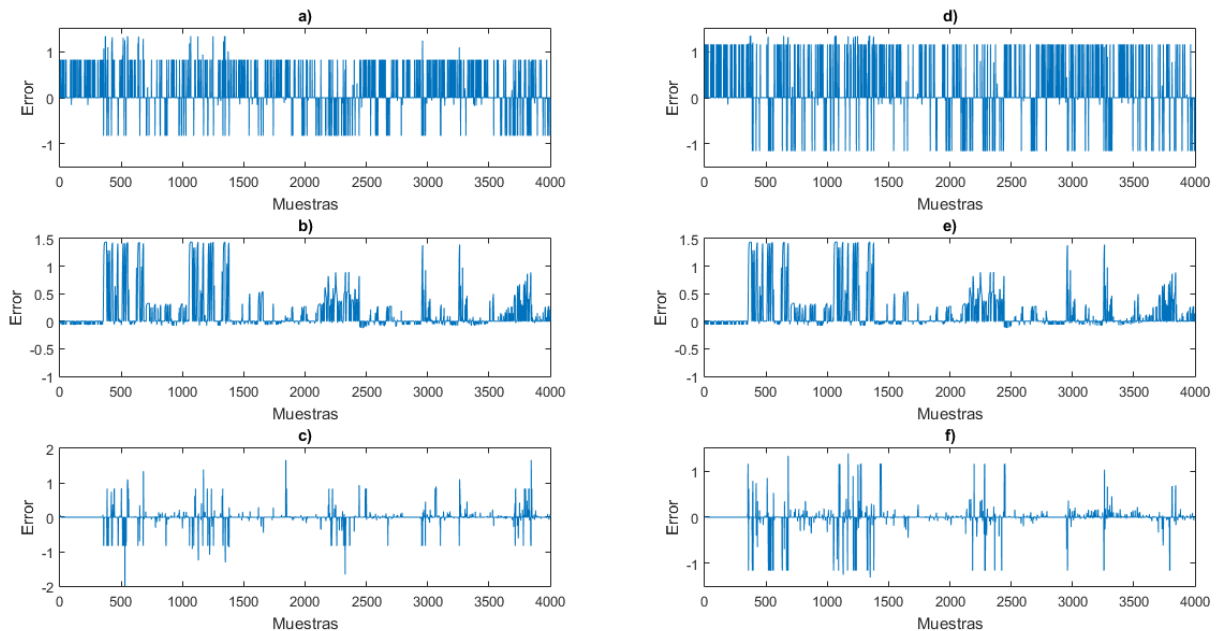


En las figuras 4.1, 4.2, se muestran las figuras de mérito del error entre datos originales del conjunto y datos filtrados que fueron contaminados con “outliers” previamente con cada uno de los filtros descritos en el presente capítulo, en las cuales es posible observar el resultado de la detección que hace cada filtro ante diferentes cantidades y amplitudes de “outliers”.

Particularmente, en la figura 4.1 se muestra el desempeño de los filtros a través de una figura de mérito de error, ante conjuntos con 2% y 25% de datos fuera de tendencia, confirmando que el mejor desempeño corresponde a el filtro de Hampel, el cual detecta de manera más efectiva y más uniforme los “outliers” que contiene el conjunto de datos.

Por otro lado, en la figura 4.2 se presenta el desempeño de cada uno de los filtros descritos, ante conjuntos que contienen “outliers” de diferentes amplitudes, específicamente se muestran las figuras de mérito de error ante datos fuera de tendencia desviados 2.5σ y 3.5σ , donde nuevamente el filtro Hampel tiene un desempeño mucho mejor que los resultados obtenidos con la regla $3-\sigma$ y el coeficiente MAD.

Figura 4.2 Error de datos filtrados: a) regla $3-\sigma$ con 2.5σ de amplitud, b) MAD con 2.5σ de amplitud, c) Filtro Hampel con 2.5σ de amplitud, d) regla $3-\sigma$ con 3.5σ de amplitud, e) MAD con 3.5σ de amplitud, f) Filtro Hampel con 3.5σ de amplitud.



4.2 Reducción dimensión conjunto de datos

Típicamente los sistemas de instrumentación de los entornos industriales almacenan información de una gran cantidad de variables que pertenecen al proceso, sin embargo, no todas estas variables son relevantes o representativas para la dinámica del proceso y en el caso de los sensores inferenciales para la estimación de las variables de interés. Por consiguiente, existen técnicas para la reducción de la dimensionalidad de los conjuntos de datos, lo cual no solo beneficia desde el aspecto de costos, debido a que no es necesaria una gran cantidad de sensores para estimar la variable, sino desde el punto de vista de desempeño de las técnicas de aprendizaje de máquina ya que existe una relación entre el desempeño de la estimación, la cantidad de variables y la cantidad de muestras del conjunto; denominado “curse of dimensionality”³⁶ (Alpaydın, 2010; Bishop, 2006). En otras palabras, si se tiene una gran cantidad de variables y pocas muestras en relación con estas, el desempeño de las técnicas de aprendizaje se degrada ya que se puede presentar sobreajuste del modelo. En consecuencia, existen 2 enfoques para realizar estas tareas de reducción de variables, por un lado, es posible encontrar las variables que son más representativas para la dinámica del modelo o es posible combinar algunas de las variables para obtener un nuevo conjunto de características que es más pequeño e igualmente representativo; ambos casos serán discutidos en las siguientes sub-secciones (Alpaydın, 2010).

4.2.1 Selección de variables o características

El problema de selección de variables o características ha sido ampliamente tratado y reportado en múltiples trabajos, de los cuales, se distinguen 3 enfoques principales que son:

- Métodos de Filtrado

Son métodos que buscan establecer un ranking de cada una de las variables del conjunto de datos, de tal manera que se resalten criterios como la relevancia dentro del conjunto, evitar la co-linearidad y redundancia de información y en algunos casos inclusive mejorar el desempeño. Entre los trabajos más destacados se encuentran (Bella, Fortuna, & Graziani, 2007; Gonzaga et al., 2009) que realizan el filtrado por medio del coeficiente de correlación de Pearson (Guyon et al., 2003), uno de los más utilizados en este enfoque. Así mismo, trabajos como en (Battiti, 1994) presentan un algoritmo denominado MIFS³⁷ que utiliza la teoría de la información para calcular el coeficiente de información mutua, el cual se utiliza como criterio de clasificación de variables buscando que la selección

³⁶ Curse of dimensionality es un término del inglés ampliamente conocido en el ámbito del aprendizaje de máquina

³⁷ MIFS por sus siglas en inglés Mutual Information Feature Selection

contenga máxima relevancia y mínima redundancia; en el mismo sentido en (Peng, Long, & Ding, 2005) se reporta el algoritmo “mRMR” el cual utiliza el coeficiente de información mutua para seleccionar las características con los mismos criterios de (Battiti, 1994), pero con un procedimiento más robusto. Otras variantes de este algoritmo se han reportado buscando mejoras de desempeño y gasto computacional como en (Ludwig, Nunes, Araújo, Schnitman, & Lepikson, 2009; F. Souza, Santos, & Araújo, 2010).

- Métodos “Wrapper”

Son métodos que realizan selección de características a partir de una medida de desempeño del modelo, a diferencia de los métodos de filtrado que solamente utiliza la información presente en el conjunto de datos para generar el ranking de variables. Teniendo en cuenta lo anterior es posible inferir que la selección de variables se realiza teniendo como objetivo principal el mejor desempeño del modelo en cuanto a exactitud en la predicción. Con este enfoque se han reportado varios trabajos, por ejemplo, en (Fortuna et al., 2005) hace la selección de los auto-regresores a través de un análisis de correlación reportado en (Billings & Voon, 1986), donde se establecen criterios de desempeño por la correlación entre características y el error de estimación. Por otro lado, en (Bhartiya & Whiteley, 2001) se utiliza la métrica de desempeño “Cp mallows” la cual es altamente utilizada en el problema de selección de características (C. L. Mallows, 1973). En cualquiera de los casos o métrica de desempeño que se seleccione, la desventaja de estos métodos está en la selección de subconjuntos de variables para probar el desempeño del modelo, es decir si se prueban todas las posibles combinaciones la cantidad de pruebas puede resultar muy grande, por lo cual algoritmos como “hill climbing” reportado en (Kohavi & John, 1997) establece una metodología para evaluar una menor cantidad de subconjuntos de variables.

- Métodos híbridos

Son métodos que combinan los anteriormente mencionados, por ejemplo en (Luigi Fortuna, Salvatore Graziani, 2005) utilizan gráficos de dispersión entre las variables, coeficiente de correlación y coeficiente de ajuste “Cp Mallows” (C. L. Mallows, 1973), donde los gráficos y el coeficiente de correlación se utilizan para hacer un pre-filtrado de las características y posteriormente se realizan pruebas con diferentes subconjuntos hasta encontrar un desempeño aceptable en la predicción del modelo.

A continuación, se presentan algunos métodos de selección de características o variables, a través de métodos de filtrado:

Conocimiento Experto

La primera instancia para obtener una reducción de variables es el conocimiento experto, ya que el personal de planta conoce profundamente la dinámica de proceso y por consiguiente las variables que afectan el proceso y el efecto que producen en el mismo. Aunque, una de las motivaciones del presente proyecto es evitar al máximo el uso de este

conocimiento con el fin de lograr metodologías que en su mayoría sean basadas en datos, siempre es útil la incorporación de conocimiento experto para reducir el conjunto de datos.

Coeficiente de correlación

También denominadas medidas de similitud o dependencia, los coeficientes de correlación son útiles para establecer si una serie de tiempo puede ser útil para predecir otra o si el uso de una determinada serie de tiempo puede mejorar la predicción de otra. De hecho, en esto consiste la tarea de selección de características; en encontrar las variables más relevantes para predecir una variable de interés (De Silva & Leong, 2015).

▪ Coeficiente de Pearson

El coeficiente de Pearson calcula la correlación entre 2 series de tiempo a partir de las medidas de tendencia de los datos que las componen. Esta dada por:

$$R = \frac{\sum(x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum(x_i - \bar{x}_i)^2 \sum(y_i - \bar{y}_i)^2}} = \frac{cov(x, y)}{\sqrt{var(x_i)var(y_i)}} \quad (4.2)$$

El coeficiente de Pearson fue creado para detectar correlación de tipo lineal entre dos series de tiempo.

▪ Coeficiente de Spearman

El coeficiente de correlación de Spearman también calcula la correlación que existe entre los datos de dos series de tiempo, sin embargo, no utiliza directamente los datos de los conjuntos de datos, sino que asigna rangos a cada pareja (x, y) y calcula la medida a partir de dichos rangos, lo cual le da algunas características, como poder evaluar valores de tipo cualitativo y detectar relaciones de tipo no lineal entre los datos. El cálculo del coeficiente de Spearman esta dado por:

$$\rho = 1 - \frac{6 \sum d^2}{n(n^2 - 1)} \quad (4.3)$$

Donde d es la diferencia entre los rangos de cada una de las parejas (x, y) y n es el número total de muestras de las series de tiempo que se están analizando.

Los dos casos anteriores, presentan un resultado en la misma escala, es decir el valor de R y ρ están comprendidos entre $[-1, 1]$ de la siguiente manera:

$$R = \rho < 0; \quad \text{Correlación negativa}$$

$$R = \rho > 0; \quad \text{Correlación positiva}$$

$$R = \rho = 0; \quad \text{No existe correlación}$$

Por lo tanto, si el coeficiente de correlación está cerca de 1 o -1, las series de tiempo están “fuertemente” correlacionadas, pero si está cerca de cero, la relación entre los datos es “débil”.

Criterio teoría de información

Existen diferentes enfoques para la selección de variables utilizando la teoría de la información, tal como se relaciona en diferentes revisiones como las reportadas en (Torkkola, 2003), entre las cuales se puede mencionar el coeficiente de ganancia de información (IG), el coeficiente de información mutua (MI), el coeficiente de máxima información (MIC), entre otros.

Los tres criterios anteriormente mencionados, son calculados a partir de la entropía de los datos. La entropía es la incertidumbre de la fuente de datos (Guyon et al., 2003) y está dada por:

$$H(X) = - \int f(x) \log f(x) dx \quad (4.4)$$

$$H(Y|X) = \iint f(x,y) \log \frac{f(x,y)}{f(x)f(y)} dx dy \quad (4.5)$$

Las ecuaciones 4.4 y 4.5 sirven para calcular la entropía de una variable aleatoria de tipo continuo. En el caso de la ecuación (4.4) es la entropía para una variable considerada de manera independiente y la ecuación (4.5) permite el cálculo de la entropía condicional entre 2 variables aleatorias continuas. En ambos casos las funciones denominadas f son las funciones de densidad de probabilidad de los conjuntos de datos, las cuales son altamente difíciles de estimar (Guyon et al., 2003), por lo tanto, cuando la entropía se estima con el propósito de aplicar algún criterio de selección de características se recomienda estimar una aproximación de estas funciones de densidad de probabilidad, discretizando los datos y calculando la frecuencia de aparición de los datos discretos.

Teniendo en cuenta lo anterior, la entropía para variables discretas es posible calcularla de la siguiente manera (Guyon et al., 2003):

$$H(X) = - \sum_i p(x_i) \log_2 p(x_i) \quad (4.6)$$

$$H(Y|X) = - \sum_j \sum_i p(y_j|x_i) \log_2 p(y_j|x_i) \quad (4.7)$$

Donde x_i y y_j son los valores discretos de los conjuntos de datos, p es la probabilidad de aparición, calculada a partir de sus frecuencias.

A partir del cálculo o estimación de la entropía, es posible obtener diferentes criterios de información tales como:

- **Información Mutua (MI)**

El coeficiente de información mutua mide la dependencia entre dos variables aleatorias. Típicamente se calcula en bits y está dado por:

$$MI(Y, X) = H(Y) + H(X) - H(Y|X)$$

Aplicando las ecuaciones (4.6) y (4.7) se obtiene:

$$MI(Y, X) = - \sum_{i,j} p(y_j, x_i) \log_2 \frac{p(y_j, x_i)}{p(y_j)p(x_i)} \quad (4.8)$$

Entre mayor sea el coeficiente de información mutua, mayor es la dependencia entre las variables que se están analizando, por lo tanto, el criterio para seleccionar variables debe tener como objetivo la maximización de este parámetro.

- **Ganancia de información (IG)**

Es una medida de comparación entre la información existente en la variable dependiente con respecto a la información teniendo en cuenta las características (Torkkola, 2003). Esta dada por:

$$IG(Y, X) = H(Y)H(Y|X)$$

Al igual que con el coeficiente de información mutua, el coeficiente de ganancia de información entre mayor sea, mayor dependencia existe entre las variables.

- **Coficiente de información máxima (MIC)**

Es una medida de dependencia entre dos variables aleatorias cuya relación es lineal o no lineal. Esta dada por:

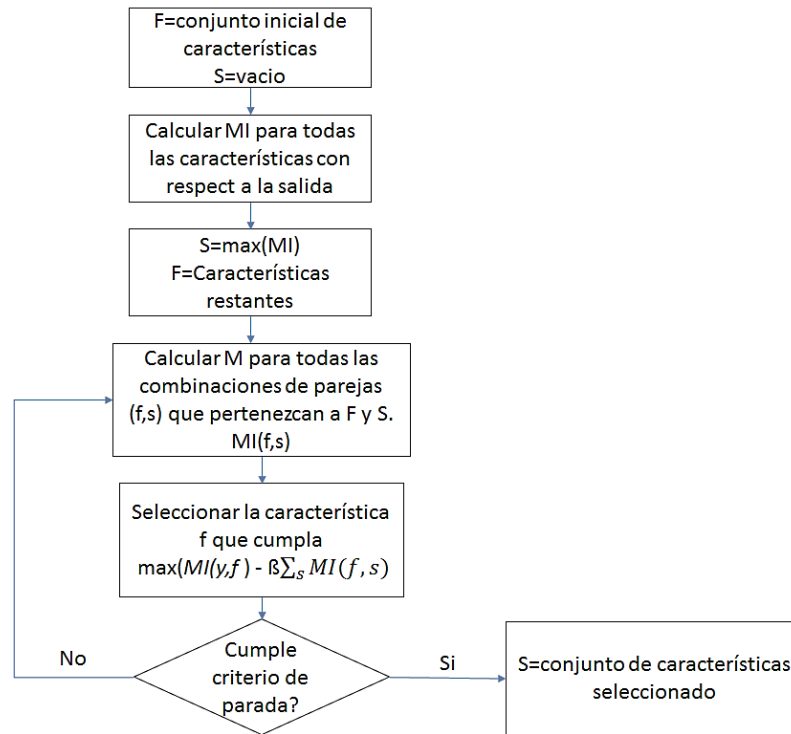
$$MIC(Y, X) = \frac{H(X) - H(Y|X)}{\min\{H(X), H(Y)\}}$$

La interpretación del MIC indica que si $0 < MIC(Y, X) < 1$ existe dependencia entre las variables X y Y; si $MIC(Y, X) = 0$ indica que las variables son independientes.

De lo anterior, y en una revisión de la literatura donde se aplica criterios de teoría de la información para la selección de características, se encontró que en sensores inferenciales el coeficiente más utilizado es el coeficiente de información mutua (MI), el cual se utiliza como criterio para crear un ranking de características identificando las que son mayormente dependientes de la salida (F. A. A. de Souza, 2014). Para llevar a cabo esto, en la literatura se han planteado múltiples propuestas de algoritmos, entre los cuales los más clásicos son el algoritmo de selección de características de información mutua

(MIFS³⁸) y el algoritmo de mínima redundancia máxima relevancia (mRMR³⁹), los cuales son altamente referenciados en trabajos que involucran regresión y diseño de sensores inferenciales. Por consiguiente, el presente trabajo abordará el enfoque de información mutua mediante el algoritmo MIFS propuesto en (Battiti, 1994) y cuyo diagrama de flujo se observa en la figura.

Figura 4.3 Diagrama de flujo algoritmo MIFS. Adaptado de (Battiti, 1994)



Donde F y S corresponden al conjunto inicial de características del conjunto de datos y el conjunto de características seleccionadas. En el mismo sentido f y s corresponden a características que pertenecen a los conjuntos F y S. Por último, β es el peso que se desea dar a la penalización por redundancia entre características, en (Battiti, 1994) se recomienda este valor en el intervalo $[0.7, 1]$; el criterio de parada depende del número de características que se deseen conservar con respecto al conjunto inicial.

▪ Aplicación de técnicas de selección de características

Teniendo en cuenta los 3 criterios presentados en párrafos anteriores, lo siguiente es la aplicación de los mismos para seleccionar un conjunto de variables adecuado que

³⁸ Por sus siglas en inglés Mutual Information Feature Selection

³⁹ Por sus siglas en inglés Minimal Redundancy Maximal Relevance

contribuya con el mejoramiento del desempeño de las técnicas de aprendizaje de máquina descritas en el capítulo 2. El conjunto que se utiliza está compuesto por 4000 muestras o patrones y 60 características de entrada que se compararán frente a la salida que es la concentración de etanol. Estos datos fueron obtenidos a partir de la simulación desarrollada en el capítulo 3 y la cantidad de variables del conjunto de datos se presenta en la tabla 4.3

Tabla 4.3 Variables conjunto de datos inicial

Variable	Presión	Flujos de líquido	Flujos de vapor	Temperatura	Masa retenida	Flujo destilado	Flujo fondo	Flujo alimentación	Total
Cantidad	14	14	1	14	14	1	1	1	60

- **Aplicación del conocimiento experto**

Aunque el presente trabajo busca obtener información a partir de los datos, eludiendo al máximo el conocimiento experto del proceso y el modelo matemático, el uso de éste en la mayoría de los casos resulta un excelente complemento y un buen punto de partida para iniciar con la eliminación de características, que por experiencia ya es conocido que no aportan a la estimación de la variable deseada. En este sentido expertos en procesos y modelos reportados en (Luyben, 1999; Roffel & Betlem, 2007) indican que la caída de presión a lo largo de la columna es casi constante por lo que es válido utilizar solamente una medida de presión para toda la columna. Lo mismo aplica para el flujo de reflujo y el flujo de vapor, ya que en el diseño de la columna de destilación se busca que estos flujos sean constantes a lo largo de la misma. Por último, la misma idea se aplica para las masas retenidas en cada bandeja, las cuales son iguales en toda la columna de destilación. Después de realizar este análisis utilizando como base el conocimiento del proceso y del modelo matemático se llega a un conjunto de 20 variables y 4000 muestras los cuales se especifican en la tabla 4.3.

- **Aplicación de los coeficientes de correlación**

A continuación, se presenta el cálculo de los coeficientes de correlación y la selección de características para el conjunto de datos obtenido del modelo matemático del proceso de destilación presentado en el capítulo 3 y cuyas variables se definen en la tabla 4.4.

Tabla 4.4 Variables conjunto de datos reducido por conocimiento experto

Variable	Presión	Flujos de líquido	Flujos de vapor	Temperatura	Flujo destilado	Flujo fondo	Flujo alimentación	Total
Cantidad	1	1	1	14	1	1	1	20

Se aplican los coeficientes de correlación de Pearson y de Spearman al conjunto de la tabla 4.4, utilizando cada una de las 20 características con respecto a la salida (concentración molar de etanol). Los resultados se presentan en la tabla 4.5.

De la tabla 4.5 y siguiendo con el criterio de coeficientes de correlación, es posible observar que existe una relación “nula” entre la concentración molar de etanol (salida) y la presión, así mismo, el flujo molar de líquido y vapor (L y V), el flujo molar de destilado y fondo (D y B) y el flujo molar de alimentación (F) exhiben una relación “débil” con respecto a la concentración de etanol. Sin embargo, es necesario acotar lo siguiente:

- La presión presenta los valores más bajos de correlación en ambos casos, prácticamente cero, por lo cual se descarta. Esto es compatible con el comportamiento del modelo matemático el cual se soluciona por medio de la presunción de $P=0$.
- Los flujos molares L, V, D, B y F presentan valores relativamente bajos de correlación, para ambos casos, sin embargo, del modelo matemático es conocido, por ejemplo, que L y D por si solos no afectan drásticamente la concentración de etanol sin embargo su relación (Relación de reflujo) es la variable que controla la concentración de destilado; teniendo en cuenta lo anterior, L, D, y F no se eliminan del conjunto. En contraste se eliminan B y D los cuales por información que suministra el modelo matemático es conocido que no afecta la concentración de etanol.

Tabla 4.5 Coeficientes de correlación para conjunto de 20 variables

Variable	Pearson	Spearman	Promedio
P	5.1032e-15	0	2.6511e-15
T1	-0.9807	-1	-0.9903
T2	-0.9389	-0.9928	-0.9658
T3	-0.9464	-0.9825	-0.9645
T4	-0.9265	-0.9706	-0.9486
T5	-0.8800	-0.9561	-0.9180
T6	-0.8371	-0.9374	-0.8873
T7	-0.7917	-0.9138	-0.8528
T8	-0.7446	-0.8822	-0.8134
T9	-0.6996	-0.8432	-0.7714
T10	-0.6538	-0.7959	-0.7249
T11	-0.6049	-0.7368	-0.6709
T12	-0.5531	-0.6465	-0.5998
T13	-0.4877	-0.5282	-0.5079
T14	-0.3952	-0.4987	-0.4470
L	0.0424	0.1829	0.1127
V	0.0344	0.1791	0.1067
D	-0.0260	0.0554	0.0147
B	0.4399	0.1309	0.2854
F	0.5003	0.1844	0.3424

- Por último, las temperaturas de la etapa 11 a la 14 (T11, ..., T14), son menores de 0.7, no obstante, la correlación no nos da certeza suficiente para eliminarlas, por consiguiente, se definirá su eliminación a través de pruebas de medición del error

medio cuadrático (MSE) de entrenamiento y validación con las técnicas de aprendizaje de máquina.

En la tabla 4.6 se presentan los errores de entrenamiento y validación para cada una de las técnicas de aprendizaje frente a la eliminación progresiva de las temperaturas T11 a la T14.

Tabla 4.6 MSE de entrenamiento y validación para ANN, SVM y ANFIS frente a eliminación de variables

Reducción	ANN		SVM		ANFIS	
	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación
T14	4.0411e-5	4.0973e-4	1.0363e-4	1.0251e-4	4.7888e-5	1.1466e-4
T14, T13	4.2588e-5	3.8531e-4	9.8100e-5	9.9161e-5	7.7729e-5	1.1818e-4
T14, T13, T12	4.3756e-5	4.4539e-4	9.4070e-5	9.8724e-5	6.6271e-5	1.1491e-4
T14, T13, T12, T11	4.5737e-5	4.0910e-4	8.5845e-5	1.0057e-4	7.0365e-5	1.1794e-4
Ninguna	4.5208e-5	4.1716e-4	1.1702e-4	1.0012e-4	4.4087e-5	1.2687e-4

De la tabla 4.6 es posible observar que los errores de entrenamiento y validación con cada una de las técnicas se mantiene prácticamente invariante ante la eliminación de las variables de temperatura, en la tabla 4.7 se expresa el porcentaje de variación con respecto a la situación donde no se elimina ninguna de las variables de temperatura, por lo cual con el criterio de correlación y adicionando un análisis de error en la técnica de aprendizaje es posible establecer la eliminación de algunas variables adicionales.

Después de realizar el proceso de entrenamiento, eliminando progresivamente las variables candidatas, es posible concluir que las medidas de temperatura T11 a T14 no son relevantes para la predicción de la concentración molar de etanol y por consiguiente se pueden eliminar del conjunto de datos, reduciendo a un total de 15 variables.

Tabla 4.7 Porcentaje de variación de MSE frente a la eliminación de variables candidatas. Para ANN, SVM y ANFIS

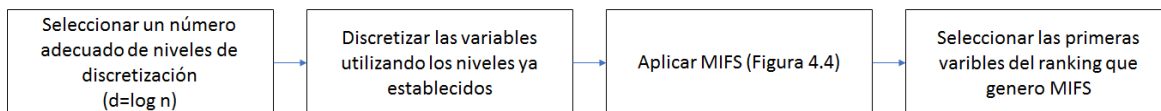
Reducción	ANN	SVM	ANFIS
Ninguna	4.1716e-4	1.0012e-4	1.2687e-4
T14, T13, T12, T11	1.9%	0.4%	7%

Tal como se aprecia, la metodología para la selección de variables a través de la correlación entre las variables medidas y la variable estimada no es certera, es necesario múltiples entrenamientos para probar la relevancia de algunas variables, además requiere del apoyo del conocimiento del proceso y/o del modelo matemático, para tomar decisiones de eliminar variables candidatas.

- **Aplicación de Selección de características por información mutua (MIFS)**

Como se dijo con anterioridad, el uso del coeficiente de información mutua (MI) es ampliamente reportado en la selección de características, por lo cual, se aplicará el algoritmo MIFS descrito en la figura 4.4 para buscar las características más relevantes con respecto a la salida y evitar redundancia entre las características seleccionadas, con el fin de seleccionar las “mejores” variables para estimar la concentración de etanol. El algoritmo ya mencionado se correrá con los datos correspondientes al conjunto descrito en la tabla 4.3 que posee 20 características.

Figura 4.4 Proceso aplicación MIFS



Para implementar el algoritmo MIFS es necesario calcular el MI, el cuál puede ser estimado mediante una aproximación convirtiendo en variables discretas el conjunto de la tabla 4.4. En (Guyon et al., 2003) se indica que esta aproximación es suficientemente exacta para llevar a cabo la selección de características. El proceso para implementar el MIFS a los datos se ilustra en la figura 4.4.

Después de aplicar el algoritmo de selección de características por información mutua (MIFS), se obtuvo el ranking que se presenta en la tabla 4.8, donde solamente se presentan las primeras 10 características mejor calificada por el algoritmo.

Tabla 4.8 Primeras variables del ranking de MIFS

Variable	T1	T2	Flujo destilado	Flujo alimentación	T6	T12	Flujo líquido	T4	Flujo Fondo	T7
MI(Y,X)	3.42	2.47	0.17	0.23	1.52	0.98	0.69	2	0.57	1.38

En la tabla 4.8 es posible observar que el ranking que arrojó el algoritmo MIFS, corresponde plenamente con lo esperado a partir del conocimiento experto del proceso y el modelo matemático de la columna binaria de destilación.

Adicionalmente, se observa que el algoritmo busca garantizar la máxima relevancia de las características seleccionadas, sin embargo, esto no implica que todas las variables tengan el máximo coeficiente de información mutua (MI), ya que tiene en cuenta el otro criterio que es garantizar mínima redundancia entre las características seleccionadas. En otras palabras, pueden existir 2 variables con altos MI, pero desde el punto de vista del modelo entregan la misma información, denominada también información redundante, por lo cual tener ambas variables no colabora dando riqueza de información al modelo.

Por último, después de múltiples simulaciones realizadas, se identificó que los modelos obtenidos con ANN, SVM y modelos ANFIS requieren de menos de 10 de las variables presentadas en la tabla 4.8.

En la tabla 4.9, se presenta algunos resultados de variables seleccionadas para cada una de las técnicas de aprendizaje de máquina y se evalúa su desempeño por medio de una métrica muy utilizada en el área de modelamiento basado en datos, denominada coeficiente de ajuste "Cp Mallows", el cual se especifica como medida de desempeño en trabajos referentes a sensores inferenciales como en (Bhartiya & Whiteley, 2001) y se describe matemáticamente en el capítulo siguiente.

Después de realizar múltiples simulaciones con diferentes conjuntos de variables o características, se encontró que las redes neuronales artificiales solamente requieren de las primeras 5 variables del ranking obtenido con el algoritmo MIFS. En el mismo sentido, después de realizar varios entrenamientos con las SVM, la variación del error entre las primeras 5 variables y las 10 primeras variables es de 6×10^{-5} , por lo tanto, se considera aceptable el modelo de las primeras 5 variables.

Tabla 4.9 Coeficiente de ajuste y MSE de validación para diferentes subconjuntos de variables

Modelo	ANN		SVM		ANFIS	
	Cp Mallows	MSE Validación	Cp Mallows	MSE Validación	Cp Mallows	MSE Validación
T1	12654	7.8941e-4	2509	1.7429e-4	5.019e6	0.1911
T1,T2,D	11550	7.6956e-4	2656	1.8422e-4	1.446e6	0.0553
T1,T2,D,F,T6	6783	3.9569e-4	2181	1.5107e-4	6.370e5	0.0242
T1,T2,D,F, T6,T12,L,T4	29779	0.0017	1545	1.0664e-4	3.6378e5	0.0135
T1,T2,D,F,T6,T12,L,T4,B,T7	26574	0.0016	1395	9.5939e-5	4933	1.8564e-4

En el caso de los sistemas ANFIS, se observa el mejor resultado con las 10 primeras variables, sin embargo, después de verificar varias combinaciones de variables se encontró que en el modelo de ANFIS es muy importante considerar la variable “Flujo de fondo”, y verificando los pesos asignados a cada variable en el modelo resultante se encontró que dicha característica tiene un alto peso dentro del modelo. En la tabla 4.10 se indican los resultados y el modelo obtenido para los sistemas ANFIS.

Tabla 4.10 Características obtenidas para modelo con sistema ANFIS

Modelo	Cp Mallows	MSE Validación
T1,T2,D,F,T6,T12,B	6609	2.4910e-4

En la exploración realizada se encontró que las técnicas de selección de características con criterios basados en teoría de la información resultan más certeras al momento de seleccionar las características más adecuadas para el entrenamiento de las técnicas de aprendizaje de máquina.

Aunque también es necesario complementar el algoritmo MIFS con algunas pruebas entrenando las técnicas de aprendizaje, estas pruebas se hacen sobre un conjunto de variables ya seleccionado y con la certeza que son características que aportan significativamente al modelo.

4.2.2 Extracción de características

Siguiendo con la temática tratada en la sección anterior, a continuación, se describe la técnica de extracción de características denominada Análisis de componentes principales, que en el documento será referida como PCA⁴⁰.

▪ Análisis de componentes principales

La técnica de análisis de componentes principales consiste en realizar una transformación del espacio de los datos $\mathbb{R}^n \rightarrow \mathbb{R}^m$ donde, $n > m$, en otras palabras, proyectar los datos desde un espacio con n dimensiones hacia un espacio de menor cantidad de dimensiones con la mínima pérdida de información (Pearson, 2001).

La proyección de los datos en el nuevo espacio se realiza siguiendo la ecuación (4.4)

$$z = w_i^T x \quad (4.4)$$

Se trata de una técnica no supervisada, la cual no utiliza la información de salida, únicamente las entradas. Para lograr la reducción de características, PCA calcula nuevas variables, denominadas componentes principales, que son una composición lineal del conjunto original, tal como lo expresa la ecuación (4.4), donde w_i son los pesos que el algoritmo asigna a cada una de las variables originales de acuerdo a su contenido de información. El algoritmo entrega una matriz de $k \times n$, donde k es el número de muestras del conjunto de datos y n es el número de variables del conjunto original.

Nótese, que la dimensión de la matriz resultante después del cálculo de los componentes principales es igual al conjunto de datos original, por lo tanto es necesario establecer un criterio de reducción que cumpla los siguientes objetivos (Pearson, 2001):

- a) Extraer la mayor cantidad de información del conjunto original de datos
- b) Reducir la cantidad de variables conservando la información más importante
- c) Simplificar la descripción del conjunto de datos
- d) Facilitar el análisis de la estructura de los datos y las variables

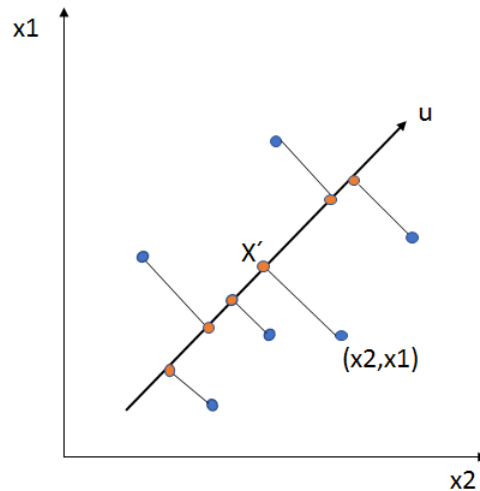
El algoritmo de componentes principales organiza las nuevas variables comenzando por el primer componente principal, el cual, posee la mayor varianza y por ende se asume que tiene la mayor cantidad de información. El segundo componente principal es calculado de tal manera que resulte ortogonal al primer componente. Los componentes principales restantes tienen una varianza mucho menor al primero y también son ortogonales.

Los valores asociados a estas nuevas variables pueden ser interpretados geoméricamente como las proyecciones de los datos en los componentes principales, que

⁴⁰ PCA por sus siglas en inglés Principal Component Analysis

desde el mismo punto de vista pueden ser interpretados como los vectores que conforman el nuevo espacio.

Figura 4.5 Interpretación geométrica PCA



En la figura 4.5 se observa como los datos son proyectados hacia los componentes principales, en este caso solo un componente principal u , reduciendo la dimensionalidad del espacio de 2 a 1 dimensión.

▪ Criterio de reducción

Como se dijo en párrafos anteriores, el conjunto resultante después de realizar las proyecciones es de las mismas dimensiones que el conjunto de variables originales, por consiguiente, es necesario establecer un criterio de reducción para obtener la mínima cantidad de componentes principales con la mayor cantidad de información posible. Para obtener esto se utiliza las varianzas correspondientes a cada uno de los componentes principales, sumándolas una a una hasta obtener la varianza acumulada deseada; cuando se cumpla con el porcentaje de varianza necesario se tendrán los componentes principales suficientes.

El porcentaje de varianza se selecciona dentro del contexto del conjunto de datos que se está tratando ya que dependiendo de la situación que describen los datos existen conjuntos que con un 80% de varianza están suficientemente bien descritos mientras que en otros casos pueden ser necesarias varianzas acumuladas del 99%. Por lo tanto, para el caso de los sensores inferenciales, el conocimiento experto del proceso es útil en la aplicación del análisis de componentes principales para complementar el criterio de reducción de los componentes principales.

Teniendo en cuenta la definición de los componentes principales, su interpretación en el espacio de los datos y los criterios de reducción de los componentes el algoritmo 4.1 presenta el procedimiento para el cálculo y reducción de los componentes principales.

1. Normalizar los datos del conjunto, excepto la salida, para llevar todas las variables o características originales a un espacio comparable
2. Calcular la matriz de covarianza Σ
 - a. $\Sigma = \frac{1}{N} (x^T * x)$; $N = \text{Numero de muestras}; x = \text{conjunto de datos}$
3. Calculo de los valores y vectores propios
4. Calculo de la varianza de cada variable
5. While($i < \text{número de características}$)

$$\text{Var}_{\text{acum}} = \text{Var}_{\text{acum}} + \frac{\text{varianza}}{\text{varianza}_{\text{total}}}$$

If($\text{Var}_{\text{acum}} \leq \text{criterio}$)

$$k = k + 1$$

Else

Terminar

6. Calcular factores de los componentes principales $z = U_i^T * x$

Algoritmo 4.1 Extracción de características por análisis de componentes principales

▪ Aplicación PCA conjunto de datos

Teniendo como conjunto de datos base el presentado en la tabla 4.3, se procede a aplicar el algoritmo de análisis de componentes principales para extraer aquellos que poseen la mayor parte de la información del proceso.

Después de correr el algoritmo se obtiene una matriz de covarianza de dimensiones 20x20. De otro lado, los valores de varianza para cada una de las nuevas características se detallan en la tabla 4.11

Tabla 4.11 Varianza de variables en el espacio de los valores propios

Variable	Varianza σ^2
1	17.4176
2	4.5079
3	0.0730
4	0.0228
5	0.0001
6	0.0001
7	2.8339e-05
8	1.9461e-05
9	7.7898e-06
10	5.2110e-06
11	2.2155e-06
12	1.5490e-06
13	5.5904e-07
14	2.9631e-07
15	1.1665e-07
16	8.2022e-08
17	3.9711e-08
18	1.7236e-13
19	4.7716e-16
20	7.2287e-17

Por último, algunos de los valores para los primeros 5 componentes principales se presentan en la tabla 4.12, junto con el porcentaje de varianza de cada componente y el porcentaje de varianza acumulada con el fin de seleccionar aquellos que contienen la

mayor parte de la información del conjunto. Se establece que los componentes seleccionados deben contener el 99.5% de la varianza total del conjunto por lo cual el algoritmo de PCA nos indica que se deben utilizar los primeros 2 componentes principales.

Tabla 4.12 Porcentaje de varianza acumulada para los primeros 5 componentes del conjunto

Componente	1	2	3	4	5
	-0.2276	-0.1037	0.0076	0.0035	0.0002
	-0.2277	-0.1036	0.0077	0.0035	0.0001
	-0.2278	-0.1036	0.0078	0.0035	-4.831e-05
	⋮	⋮	⋮	⋮	⋮
Varianza	17.4176	4.5079	0.0730	0.0228	0.0001
%Varianza	79.09	20.47	0.33	0.10	895.33e-06
Acumulado	79.09	99.56	99.89	99.99	99.99

Así mismo, Analizando el espacio de valores y vectores propios se establece que los componentes principales seleccionados, tienen mayor peso las variables L, V, F, B, D, T13, T12, T5 y T4, tal como se aprecia en la tabla 4.13.

Después de aplicar PCA, se encontró que el algoritmo seleccionó 2 componentes principales de 20 que se generaron, por lo cual el entrenamiento de las técnicas de aprendizaje de máquina se debería realizar con 2 entradas. Teniendo en cuenta lo anterior, desde el punto de vista metodológico y práctico no es conveniente utilizar el algoritmo de extracción de características PCA, debido a que por la baja cantidad de variables de entrada la predicción puede perder generalización. El algoritmo de PCA es recomendable en casos donde el resultado conserva un número adecuado de entradas para el desempeño deseado de las técnicas de aprendizaje seleccionadas.

Tabla 4.13 Variables de mayor peso dentro de los primeros 5 componentes principales

variable	1	2	3	4	5
Presión	-0.0002	-0.0006	-0.0002	-0.0002	-0.0003
T1	-0.0998	-0.2121	-0.0798	-0.0836	-0.1548
T2	-0.0999	-0.2124	-0.0828	-0.0844	-0.2015
T3	-0.1001	-0.2129	-0.0887	-0.0851	-0.2828
T4	-0.1003	-0.2133	-0.0926	-0.0837	-0.3259
T5	-0.1005	-0.2136	-0.0945	-0.0799	-0.3157
T6	-0.1008	-0.2140	-0.0958	-0.0759	-0.2573
T7	-0.1011	-0.2143	-0.0974	-0.0729	-0.1562
T8	-0.1014	-0.2146	-0.0991	-0.0709	-0.0295
T9	-0.1016	-0.2150	-0.1006	-0.0695	0.0963
T10	-0.1019	-0.2153	-0.1020	-0.0680	0.2024
T11	-0.1021	-0.2156	-0.1031	-0.0664	0.2801
T12	-0.1024	-0.2160	-0.1035	-0.0650	0.3175
T13	-0.1035	-0.2183	-0.1168	-0.0607	0.4791
T14	-0.1045	-0.2205	-0.1118	-0.0594	0.3116
L	-0.6026	0.3732	0.0623	-0.3441	0.0119
V	-0.6752	0.2323	-0.0595	0.3335	-0.0129
D	-0.0726	-0.1409	-0.1218	0.6776	-0.0249
B	-0.0835	-0.1846	0.7070	-0.2878	0.0278
F	-0.1562	-0.3256	0.5852	0.3898	0.0029

4.3 Conclusiones

Después de aplicar las técnicas de selección y extracción de características al conjunto de datos generado a partir del modelo matemático de una columna de destilación binaria se llegó a las siguientes conclusiones:

- Una de las técnicas más extensamente usada para detectar “*outliers*” es la regla 3σ , por su fácil implementación. No obstante, es recomendable el uso de una técnica más robusta para el filtrado de los datos ya que, este aspecto tiene una influencia directa en el desempeño de la predicción, por tal motivo y con las pruebas realizadas se concluyó que el filtro de Hampel es una excelente alternativa para la detección de datos fuera de tendencia debido a que es posible ajustar sus parámetros para detectar datos fuera de tendencia en conjuntos con abundantes o pocos “*outliers*”.
- De las 3 estrategias de selección de características exploradas, conocimiento experto, coeficientes de correlación y coeficiente de información mutua, se encontró que ninguna de las 3 tienen una interpretación absoluta, y requieren de llevar a cabo un análisis con el fin de establecer indicios que permitan seleccionar las variables más relevantes. No obstante, los criterios basados en teoría de la información presentan directrices más exactas para seleccionar las variables más relevantes para el modelo
- Aunque el objetivo del presente trabajo es eliminar al máximo el conocimiento experto, se encontró que en la selección de características es necesario apoyarse en este, ya que los modelos dependientes de datos pueden presentar inconsistencias en algunos casos.
- En cuanto a la extracción de características, es un método reportado en muchos trabajos de sensores inferenciales, sin embargo, para el proceso que se está utilizando en las pruebas de desempeño, el algoritmo PCA solamente extrae 2 componentes principales lo que desde el punto de vista metodológico y práctico no es adecuado. Por esta razón no se utilizó en el diseño de los sensores basados en ANN, SVM y ANFIS.
- La aplicación de PCA en modelos no lineales, puede llevar a errores en la conformación de los componentes principales, debido a que es una técnica para datos con comportamiento lineal, por lo tanto, es recomendable, utilizar versiones más robustas de PCA que consideren no linealidades, por ejemplo, PCA con kernel.

5. Selección de Modelo

Dentro del desarrollo de los sensores inferenciales, se han reportado algunos trabajos como (Acuña, Curilem, & Cubillos, 2014), el cual describe el uso de una estructura en los datos con el fin de mejorar la exactitud en la predicción que está realizando el sensor. Dicha estructura consiste de manera general, en utilizar valores anteriores de las variables de entrada y salida, denominados auto-regresores, con el fin de insertar mayor información al conjunto de datos sin necesidad de aumentar las variables medidas. El presente capítulo se propone explorar la mejora del desempeño mediante la utilización de una estructura de modelo, se consideran dos de los modelos más reportados en la literatura los cuales son conocidos como NARX⁴¹ y NARMAX⁴². En el mismo sentido, en el capítulo se aplica una métrica de desempeño para realizar la selección de los auto-regresores que son más relevantes para realizar la predicción. El análisis se realiza sobre el conjunto de datos del modelo de la columna de destilación binaria desarrollada en el capítulo 3 con la selección de variables reportada en las tablas [4.9](#) y [4.10](#).

5.1 Estructuras de modelos dinámicos

Las estructuras de modelos dinámicos son herramientas utilizadas principalmente para la identificación de sistemas de los cuales solamente se conoce las entradas y las salidas, pero su comportamiento interno no es conocido. Estos modelos son denominados de caja negra y representan la situación que se tiene en el diseño de los sensores inferenciales, ya que la única información disponible es el conjunto de datos, que está compuesto de variables de entrada y salida, pero el conjunto de datos no tiene una estructura que permita identificar aspectos como la relación entre las variables de entrada ni la relación entre las variables de entrada con respecto a las de salida (Ljung, 1987).

Para el presente documento se consideran dos estructuras que están ampliamente reportadas y que han sido extensamente utilizadas para el desarrollo de sensores inferenciales debido a que los procesos industriales son de naturaleza dinámica no lineal y por lo tanto es correcto emplear una estructura que sea capaz de modelar este tipo de comportamientos. Por un lado, se tienen los sistemas no lineales auto-regresivos con

⁴¹ Non-linear auto regressive with exogenous input, NARX

⁴² Non-linear auto regressive movable average with exogenous input, NARMAX

entrada exógena denominados NARX, por sus siglas en inglés; de otro lado se tienen los sistemas no lineales auto-regresivos de promedio móvil y entrada exógena conocidos como NARMAX, por sus siglas en inglés.

5.1.1 NARX

Corresponde a la extensión no lineal de los sistemas denominados ARX⁴³ (Ljung, 1987), y se trata de un modelo que depende de sus entradas y salidas pasadas y de una entrada adicional denominada error de predicción la cual se puede interpretar como la incertidumbre del modelo. En la ecuación (5.1) se presenta la estructura matemática de los sistemas NARX (Acuña et al., 2014):

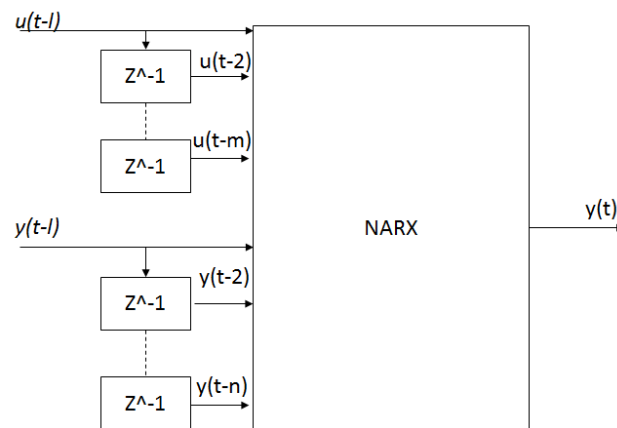
$$y(k) = f(y(k-1), \dots, y(k-n), \dots, u(k), \dots, u(k-m), \dots) + e(k) \quad (5.1)$$

El predictor asociado a este tipo de modelos tiene la siguiente forma:

$$\hat{y}(k) = \Psi(y(k-1), \dots, y(k-n), \dots, u(k-1), \dots, u(k-m), \dots) \quad (5.2)$$

Donde $\hat{y}(k)$ corresponde a la predicción de la variable “y” y depende de los valores anteriores de ella misma y de la variable exógena “u” en estados anteriores, los cuales son conocidos como auto-regresores.

Figura 5.1 Esquema sistema NARX. Tomado de (Acuña et al., 2014).



⁴³ Auto regressive with exogenous input

En la figura 5.1 se presenta el esquema de los sistemas NARX, del cual es posible interpretar como modificar el conjunto de datos para llevar a cabo el entrenamiento de las técnicas de aprendizaje seleccionadas siguiendo la estructura NARX.

- Entradas: A las entradas existentes del conjunto de datos se debe agregar las mismas entradas con los respectivos retardos duplicando de esta manera el número de entradas para el entrenamiento. Adicionalmente se debe agregar la salida con sus respectivos retardos (Acuña et al., 2014).
- Salidas: La salida corresponde a la predicción de la variable en un tiempo posterior (Acuña et al., 2014).

5.1.2 NARMAX

Corresponde a la extensión no lineal de los sistemas denominados ARMAX⁴⁴ (Ljung, 1987), y se trata de un modelo que depende de sus entradas y salidas pasadas y de los valores pasados del error de predicción. En la ecuación (5.3) se presenta la estructura matemática de los sistemas NARMAX (Acuña et al., 2014):

$$y(k) = f(y(k-1), \dots, y(k-n), \dots, u(k), \dots, u(k-m), e(k-1), \dots, e(k-p)) \quad (5.3)$$

El predictor asociado a este tipo de modelos tiene la siguiente forma:

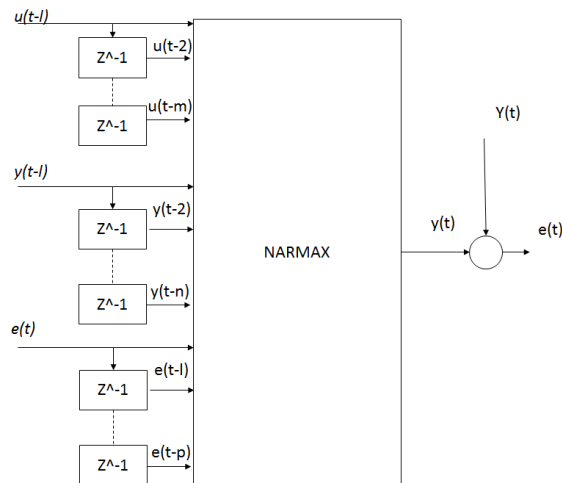
$$\hat{y}(k) = \Psi(y(k-1), \dots, y(k-n), \dots, u(k-1), \dots, u(k-m), e(k-1), \dots, e(k-p)) \quad (5.4)$$

Donde $\hat{y}(k)$ corresponde a la predicción de la variable “y” y depende de los valores anteriores de ella misma, de la variable exógena “u” en estados anteriores, y de los errores de predicción anteriores. De 5.4 es posible observar que el modelo NARMAX trae consigo mayor complejidad debido al uso de error en instantes anteriores a la predicción.

En la figura 5.2 se presenta el esquema de los sistemas NARMAX, para el caso de la columna de destilación binaria descrita en el capítulo 3 se utilizará la metodología de entrenamiento propuesta en (Acuña et al., 2014) a través de 2 sistemas NARX.

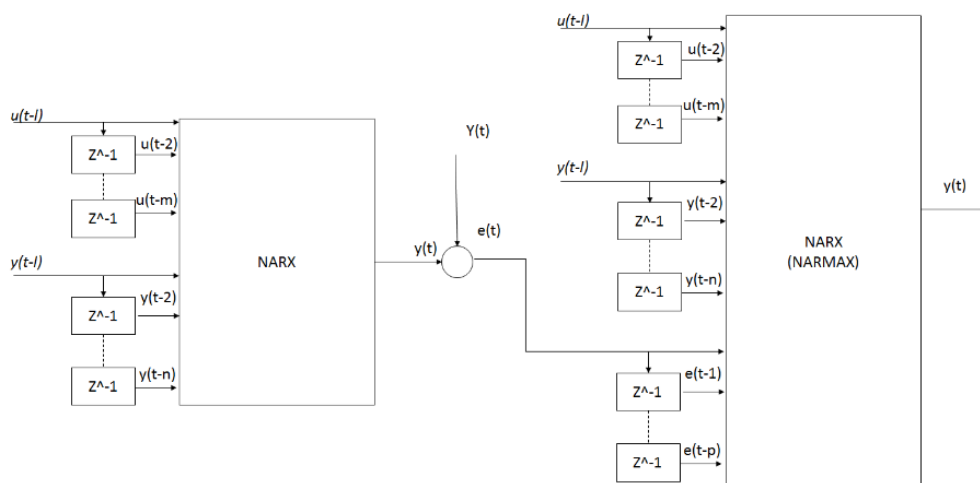
⁴⁴ Auto regressive with exogenous input

Figura 5.2 Esquema sistema NARMAX. Tomado de (Acuña et al., 2014).



- Entradas: A las entradas existentes del conjunto de datos se debe agregar las mismas entradas con los respectivos retardos duplicando de esta manera el número de entradas para el entrenamiento. Adicionalmente se debe agregar la salida con sus respectivos retardos y por último el error de estimación de una etapa NARX. En la figura 5.3 se ilustra el entrenamiento según se plantea en (Acuña et al., 2014).
- Salidas: La salida corresponde a la predicción de la variable (Acuña et al., 2014)

Figura 5.3 Metodología de entrenamiento sistema NARMAX. Tomado de (Acuña et al., 2014)



De acuerdo a lo expresado en la sección anterior, el conjunto de datos para entrenamiento sufre un incremento en el número de características, al punto de duplicar o triplicarlas según sea el orden de regresión del modelo. Teniendo esto en cuenta y su relación con el fenómeno conocido como “curse of dimensionality”, es necesario realizar una nueva selección de características con el fin de identificar cuáles de los auto-regresores son relevantes para la predicción de la variable de interés.

Por lo tanto, a continuación, se aplicará una metodología reportada en (Kohavi & John, 1997), que tiene como objetivo la selección de los auto-regresores que son más relevantes para el modelo NARX o NARMAX.

5.1.3 Selección de auto-regresores

Para realizar la selección de auto-regresores es necesario emplear nuevamente algún método para selección de características, donde se escogen aquellos que resulten más relevantes para el modelo de predicción. En la literatura revisada se encontró que trabajos como (Bhartiya & Whiteley, 2001; Fortuna et al., 2005) utilizan métodos de selección de características del tipo “wrapper”, descritos en el capítulo anterior, en parte porque se busca garantizar el mejor desempeño del modelo, por eso lo más conveniente puede ser seleccionar los auto-regresores en función del mejor desempeño con cada una de las técnicas de aprendizaje de máquina, ya que como se observó en el capítulo 4, en el sistema ANFIS se obtuvo un conjunto de características diferente a los que se consiguieron con las técnicas ANN y SVM.

En consecuencia, se realizan múltiples pruebas con diferentes combinaciones de variables, siguiendo tanto la estructura NARX como NARMAX para cada una de las técnicas de aprendizaje de máquina seleccionadas. Debido a que el presente capítulo realiza una exploración de la selección de modelo, solamente se presentan resultados con sistemas NARX y NARMAX de orden 1, en otras palabras, para el desarrollo del modelo se utilizan valores hasta un instante anterior a la predicción. No obstante, el procedimiento de selección es igual para un orden n ($n=1,2, 3,\dots$) de auto-regresión, con la diferencia que el conjunto de variables será mayor.

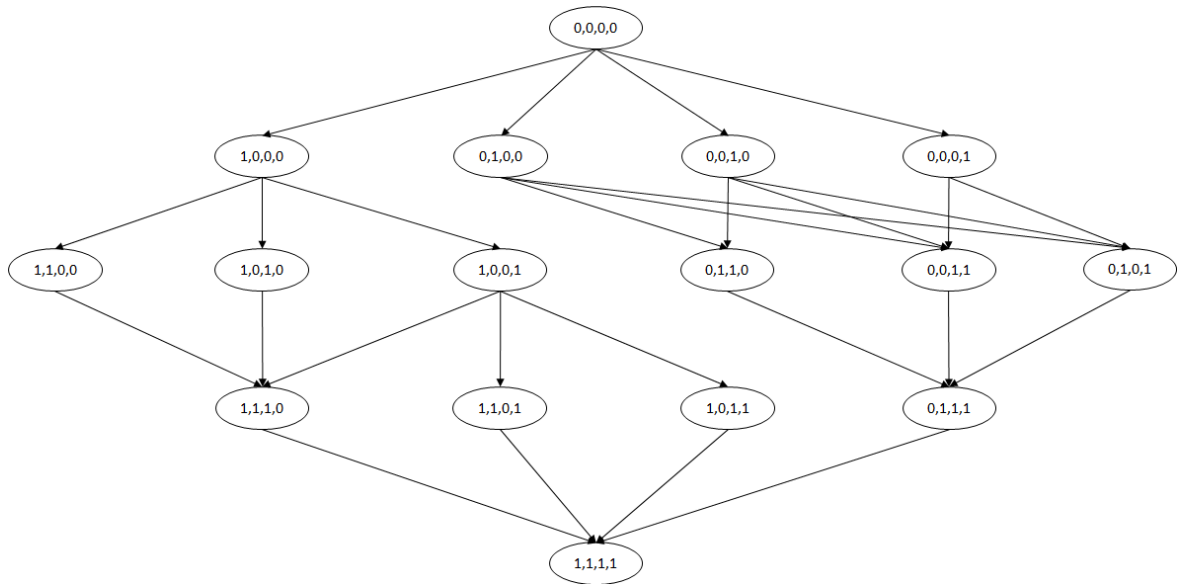
Siguiendo con la notación utilizada en las figuras 5.1 y 5.2, en la tabla 5.1 se presenta la correspondencia entre las variables físicas y los nombres de variable procedentes de la notación de sistemas auto-regresivos. En el mismo sentido, de la primera selección de características obtenida en el capítulo 4, se estableció que para el sistema ANFIS eran necesarias 2 características adicionales con respecto a lo obtenido para las redes neuronales artificiales y las máquinas de vectores de soporte, situación que esta evidenciada en la tabla 5.1.

Tabla 5.1 Correspondencia de variables y notación de sistemas auto-regresivos

Variable	ANN	SVM	ANFIS
T1(k)	$u_1(k)$	$u_1(k)$	$u_1(k)$
T2(k)	$u_2(k)$	$u_2(k)$	$u_2(k)$
D(k)	$u_3(k)$	$u_3(k)$	$u_3(k)$
F(k)	$u_4(k)$	$u_4(k)$	$u_4(k)$
T6(k)	$u_5(k)$	$u_5(k)$	$u_5(k)$
T12(k)	_____	_____	$u_6(k)$
B(k)	_____	_____	$u_7(k)$
Y(k)	$y(k)$	$y(k)$	$y(k)$

Para llevar a cabo la selección de auto-regresores, se aplica la metodología denominada “hill climbing”, reportado en (Kohavi & John, 1997), la cual inicia con un conjunto de características vacío y se agregan variables de a una y de manera sucesiva hasta conseguir el mejor desempeño posible en cuanto a la predicción. En la figura 5.4 se observa el procedimiento para ir incluyendo y probando las nuevas variables, de tal manera, que se asigna un “1” a las variables que están presentes en el conjunto de datos y un “0” a aquellas que no están presentes. Al utilizar este tipo de metodologías, es posible reducir el número de entrenamientos necesarios para encontrar un subconjunto de variables adecuado. En el mismo sentido, existen múltiples métodos para la exploración de estos sub-conjuntos de variables, buscando minimizar el número de pruebas, sin embargo, los algoritmos disponibles implican bastante complejidad y capacidad de computo, razón por la cual, se aplicará la técnica más clásica reportada en la literatura, debido a la facilidad en su interpretación e implementación. Por otro lado, para evaluar el desempeño de cada subconjunto de variables, en lo referente a la exactitud en la predicción, se utiliza el coeficiente de ajuste “Cp Mallows” reportado en (C. L. Mallows, 1973) y ampliamente utilizado en trabajos de modelamiento como los sensores inferenciales (Bhartiya & Whiteley, 2001; Park & Han, 2000; Warne et al., 2004) como medida de desempeño de la predicción del modelo.

Figura 5.4 Metodología Hill Climbing. Tomado de (Kohavi & John, 1997).



En las tablas 5.2 y 5.3 se presentan algunos resultados representativos de todo el proceso de selección, tanto para sistemas NARX como para los sistemas NARMAX, respectivamente, detallando el subconjunto utilizado en cada prueba, el coeficiente “Cp Mallows” para evaluar la calidad de la predicción y la correlación entre la predicción y los datos de validación como una medida de referencia para asegurar que se tiene una predicción “fuertemente” correlacionada con los datos.

De los resultados obtenidos, se observa que, para los tres casos, el conjunto completo de variables y auto-regresores no permite obtener el mejor desempeño en la predicción y, por consiguiente, es necesario eliminar algunos regresores que afectan dicho desempeño. En el mismo sentido los errores medios cuadráticos de validación (MSE) son aceptables y al examinarlos junto con los coeficientes “Cp Mallows” se detecta que existe una relación entre los dos.

El coeficiente de correlación muestra en todos los casos que existe una relación “fuerte” entre la predicción y los datos, lo cual por sí sólo, no constituye una medida de desempeño y se utiliza como referencia para garantizar que la predicción tiene cierta “calidad”, en otras palabras, si alguno de los coeficientes de correlación hubiera sido menor a 0.7, es un indicador de que el subconjunto correspondiente no puede modelar la salida del sensor inferencial por lo cual es necesario probar con otro subconjunto.

Tabla 5.2 Subconjuntos obtenidos para estructura NARX en metodología "Hill climbing" y métricas de desempeño

Técnica	Modelo	MSE Validación	Pearson	Cp Mallows
ANN	$f(u_1(k))$	3.267e-4	0.9804	8235
	$f(u_1(k), u_3(k), u_2(k-1), u_1(k-1))$	7.7492e-5	0.9908	2286
	$f(u_1(k), u_3(k), u_2(k-1), u_1(k-1), u_3(k-1), u_4(k-1), y(k-1))$	3.4965e-5	0.9938	850
	$f(u_1(k), u_3(k), u_5(k), u_1(k-1), \dots, u_5(k-1), y(k-1))$	5.3961e-5	0.9937	1750
	$f(u_1(k), \dots, u_5(k), u_1(k-1), \dots, u_5(k-1), y(k-1))$	8.8218e-4	0.9932	21130
SVM	$f(u_1(k))$	2.0533e-4	0.9841	2706
	$f(u_1(k), u_2(k), u_5(k-1), u_2(k-1))$	1.4140e-4	0.9876	1895
	$f(u_1(k), u_2(k), u_4(k), u_5(k-1), u_2(k-1))$	1.3839e-4	0.9827	1829
	$f(u_1(k), \dots, u_5(k), u_1(k-1), u_3(k-1), u_5(k-1), u_2(k-1))$	1.6178e-4	0.9799	2142
	$(u_1(k), \dots, u_5(k), u_1(k-1), \dots, u_5(k-1), y(k-1))$	1.9802e-4	0.9813	2629
ANFIS	$f(y(k-1))$	0.0064	0.7886	172530
	$f(u_1(k), u_3(k), u_7(k), u_5(k-1), y(k-1))$	2.4913e-4	0.9892	6696
	$f(u_1(k) \dots u_4(k), u_6(k), u_7(k), u_3(k-1), u_5(k-1), \dots, u_7(k-1), y(k-1))$	1.9212e-4	0.9962	5505
	$f(u_1(k) \dots u_7(k), u_2(k-1), \dots, u_7(k-1), y(k-1))$	3.1556e-4	0.9975	8498
	$f(u_1(k) \dots u_7(k), u_1(k-1), \dots, u_7(k-1), y(k-1))$	4.7178e-4	0.9988	12700

Por último, el objetivo de la metodología "Hill climbing" para la selección de características, es utilizar el subconjunto que ofrezca el mejor desempeño en cuanto a predicción; para este caso, como ya se dijo, la métrica de desempeño seleccionada es el coeficiente de ajuste "CP Mallows" y entre más pequeño sea éste mejor ajuste existe entre la predicción y los datos.

Tabla 5.3 Subconjuntos para estructura NARMAX en metodología “Hill climbing” y métricas de desempeño

Técnica	Modelo	MSE Validación	Pearson	Cp Mallows
ANN	$f(u_5(k), e(k-1))$	5.6741e-5	0.9970	85
	$f(u_5(k), u_4(k), e(k-1))$	8.3375e-5	0.9999	121
	$f(u_4(k), u_5(k), u_5(k-1), e(k-1))$	1.9401e5	0.9999	19
	$f(u_2(k), u_4(k), u_5(k), u_1(k-1), \dots, u_5(k-1), e(k-1))$	2.0791e-4	0.9943	354
	$f(u_1(k), \dots, u_5(k), u_1(k-1), \dots, u_5(k-1), y(k-1), e(k-1))$	0.0018	0.9967	2679
SVM	$f(u_3(k-1), e(k-1))$	1.7425e-5	0.9966	31
	$f(u_1(k), u_3(k), u_3(k-1), u_5(k-1), e(k-1))$	1.0487e-4	0.9878	233
	$f(u_1(k), u_3(k), u_2(k-1), \dots, u_4(k-1), e(k-1))$	8.6823e-5	0.9952	196
	$f(u_1(k), \dots, u_5(k), u_2(k-1), \dots, u_5(k-1), e(k-1))$	1.5674e-4	0.9968	359
	$f(u_1(k), \dots, u_5(k), u_1(k-1), \dots, u_5(k-1), y(k-1), e(k-1))$	7.1874e-4	0.9850	1625
ANFIS	$f(u_3(k))$	0.0249	0.9626	-1.94
	$f(u_3(k), u_6(k), \dots, u_8(k), u_2(k-1), u_3(k-1), u_5(k-1), u_{10}(k-1), y(k-1), e(k-1))$	0.0019	0.9191	39
	$f(u_1(k), \dots, u_3(k), u_5(k), \dots, u_8(k), u_{10}(k-1), u_1(k-1), \dots, u_3(k-1), u_5(k-1), u_{10}(k-1), y(k-1), e(k-1))$	9.5473e-4	0.9542	16
	$f(u_1(k) \dots u_{10}(k), u_1(k-1), \dots, u_3(k-1), u_5(k-1), \dots, u_{10}(k-1), y(k-1), e(k-1))$	2.2467e-3	0.2326	3419
	$f(u_1(k) \dots u_{10}(k), u_1(k-1), \dots, u_{10}(k-1), y(k-1), e(k-1))$	1.9185e-3	0.2372	3615

5.1.4 Modelos obtenidos

Después de realizar todo el proceso de selección de auto-regresores se obtuvieron los siguientes modelos. La tabla 5.4 muestra la representación matemática de los sistemas NARX obtenidos con sus respectivos auto-regresores.

Tabla 5.4 Variables obtenidas para cada técnica de aprendizaje de máquina en estructura NARX

Técnica	Modelo	Cantidad
ANN	$f(u_1(k), u_3(k), u_5(k), u_1(k-1), \dots, u_5(k-1), y(k-1))$	9
SVM	$f(u_1(k), \dots, u_5(k), u_1(k-1), \dots, u_3(k-1), u_5(k-1))$	9
ANFS	$f(u_1(k) \dots u_4(k), u_6(k), u_7(k), u_3(k-1), u_5(k-1), \dots, u_7(k-1), y(k-1))$	11

Tabla 5.5 Variables obtenidas para cada técnica de aprendizaje de máquina en estructura NARMAX

Técnica	Modelo	Cantidad
ANN	$f(u_2(k), u_4(k), u_5(k), u_1(k-1), \dots, u_5(k-1), e(k-1))$	9
SVM	$f(u_1(k), u_3(k), u_2(k-1), \dots, u_4(k-1), e(k-1))$	6
ANFS	$f(u_1(k), \dots, u_3(k), u_5(k), \dots, u_8(k), u_{10}(k), u_1(k-1), \dots, u_3(k-1), u_5(k-1), u_{10}(k-1), y(k-1), e(k-1))$	15

En contraste la tabla 5.5, muestra las variables que representan el sistema NARMAX obtenido con sus respectivos auto-regresores para cada caso.

5.2 Comparativo de modelos

Las tablas 5.4 y 5.5 contienen el subconjunto de variables y auto-regresores que componen los modelos respectivos para cada una de las técnicas de aprendizaje de máquina, tanto para la estructura NARX como para la estructura de sistemas NARMAX. En la presente sección se establecerá un comparativo entre los sistemas obtenidos, con el objetivo de verificar cuál de las estructuras ofrece un mejor desempeño en cada caso. Tal como se estableció al comienzo de este capítulo, por tratarse de una exploración solamente se tiene en cuenta sistemas de orden 1, no obstante, los resultados podrían variar frente a sistemas de orden 2 o 3, aunque no significativamente.

Despeño modelos

Cada uno de los modelos presenta diferentes características que pueden resultar adecuadas o inadecuadas frente al desarrollo de sensores inferenciales basados en datos. Para establecer algunas de estas características se entrenarán las 3 técnicas con cada una de las estructuras, NARX y NARMAX.

Como primera medida, de acuerdo a lo establecido en secciones anteriores, el procedimiento de entrenamiento de los sistemas NARX es más simple que para el sistema NARMAX, debido a la necesidad de incluir el error de predicción en los datos de entrenamiento. Existen múltiples metodologías para llevar a cabo esto, para el entrenamiento se siguió el método descrito en (Acuña et al., 2014) el cual se adecúa a la manera en que se han venido preparando los conjuntos de datos.

Tabla 5.6 MSE de validación con técnicas ANN, SVM y ANFIS bajo estructura NARX y NARMAX

Estructura	ANN		SVM		ANFIS	
	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación	MSE Entrenamiento	MSE Validación
NARX	2.0297e-5	2.6534e-4	1.4277e-4	1.4216e-4	2.3327e-5	2.0745e-4
NARMAX	3.2099e-5	1.9597e-4	5.2733e-5	2.3308e-4	1.6917e-4	0.0012

De la tabla 5.6 se observa que no existe una diferencia significativa entre usar la estructura NARX o NARMAX de orden 1, en lo referente al desempeño de la predicción. Sumando a lo anterior, la dificultad en la implementación del entrenamiento en sistemas NARMAX, para el conjunto de datos utilizado es recomendable utilizar la estructura NARX que ofrece una adecuada exactitud en la predicción y adicionalmente una implementación más directa, implicando menor esfuerzo y tiempo en el desarrollo; factores que son de bastante importancia en entornos industriales.

Teniendo en cuenta, que el presente capítulo es una exploración en la selección de modelo auto-regresivo, no es posible afirmar que una u otra estructura es más adecuada en el desarrollo de sensores inferenciales, debido a que el conjunto de datos es uno de los factores que influirá en dicha decisión, es decir, en algunos conjuntos de datos o procesos pueden resultar con mejor desempeño los sistemas NARMAX y en otros casos los NARX. Por ejemplo, en el campo de los sensores inferenciales se encuentran reportados en (Chitrlekha & Shah, 2010a; Fortuna et al., 2005), el uso de estructuras NARMAX para mejorar el desempeño de sensores inferenciales en diferentes procesos industriales, con lo cual, se obtiene un mejor desempeño en la predicción de los mismos.

Para resumir, dependiendo del proceso es necesario tomar la decisión de cuál estructura utilizar (sistemas estáticos, NARX, NARMAX u otro); realizando pruebas de validación, siguiendo una metodología que reduzca la cantidad de entrenamientos, y seleccionando métricas de desempeño apropiadas, es posible establecer cuál es la estructura más conveniente y el orden de regresión más adecuado para obtener el mejor desempeño en la predicción del sensor inferencial.

5.4 Conclusiones

Después de realizar la exploración de los 2 modelos dinámicos seleccionados, conocidos como NARX y NARMAX, y de realizar pruebas de desempeño en la implementación de sensores inferenciales basados en ANN, SVM y ANFIS se llegó a las siguientes conclusiones:

- Aunque el uso de estructuras como NARX y NARMAX implica un esfuerzo adicional en tiempo y desarrollo de los sensores inferenciales, el desempeño de los mismos tiene una influencia en la mejora del desempeño de la predicción, en la mayor parte de los procesos industriales.
- La selección de auto-regresores puede ser realizada por técnicas de filtrado como la correlación o “wrapper” como por ejemplo “hill climbing”, no obstante, es más adecuado el uso de la segunda metodología debido a que está directamente relacionada con el desempeño de la predicción; y la razón para usar estructuras NARX o NARMAX en el desarrollo de los sensores inferenciales es justamente mejorar el desempeño de los mismos.
- Existen múltiples metodologías para el uso de NARMAX, en el entrenamiento de sensores inferenciales basados en datos y su implementación requiere no solo del conocimiento de la estructura, sino también de la metodología de entrenamiento que conlleva más complejidad al diseño.
- El uso de NARX es muy conveniente en el diseño de sensores inferenciales, puesto que el entrenamiento de técnicas basadas en datos siguiendo la estructura NARX es simple y los conjuntos de datos se pueden generar de manera automatizada; adicionalmente, ofrece un buen desempeño en sistemas o procesos con dinámicas complejas.

Resumiendo, en el diseño de sensores inferenciales, basados en datos, resulta adecuado el uso de estructuras de sistemas dinámicos como NARX, NARMAX u otras que mejoren el desempeño del sensor, sin perder de vista el esfuerzo adicional que esto implique, puesto que es un factor relevante en el medio industrial guardar un equilibrio entre la calidad de la estimación del sensor y los costos en tiempo y desarrollo.

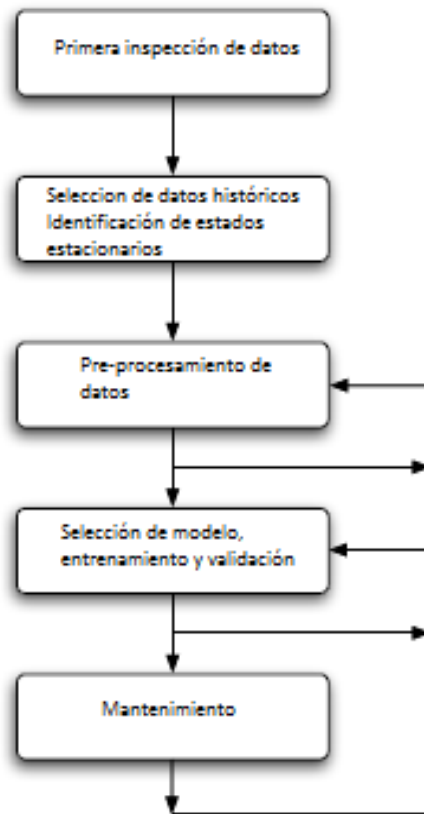
6. Metodología

En los capítulos anteriores se establecieron generalidades del comportamiento de las diferentes técnicas que están involucradas en el desarrollo de un sensor inferencial. Se caracterizaron las técnicas de aprendizaje seleccionadas con el fin de determinar cada uno de los parámetros que afectan el desempeño de las mismas. Además, se estudiaron 3 técnicas de filtrado para la detección de datos fuera de tendencia para mejorar la calidad del conjunto de datos disponible para entrenamiento y validación. Por último, se exploró el procedimiento para la selección y extracción de características, aspecto fundamental para mejorar el desempeño de la predicción y los tiempos de computo del entrenamiento. En el presente capítulo, se procederá con la descripción de una metodología para el desarrollo de sensores inferenciales, a partir, de los aspectos y las técnicas explorados en los capítulos anteriores, buscando obtener la mejor predicción de la variable deseada y evaluar su desempeño ante condiciones adversas como datos fuera de tendencia, muestreo insuficiente y pérdidas de datos.

6.1 Etapas Fundamentales

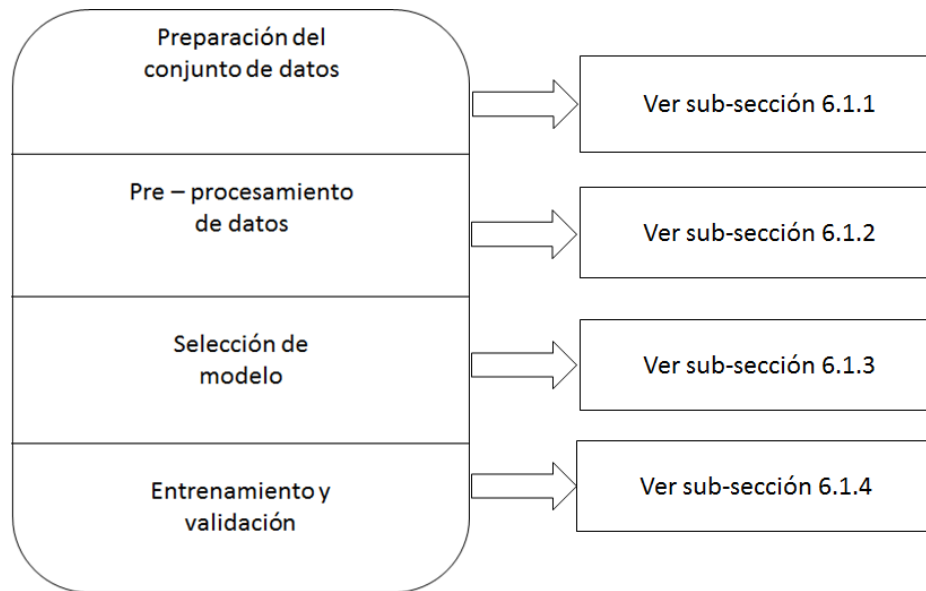
En la tesis doctoral (Kadlec, 2009) se establece que desde el punto de vista metodológico el sensor inferencial consta de 5 etapas que comprenden desde la inspección preliminar de datos hasta el mantenimiento del sensor inferencial, tal como se observa en la figura 6.1. No obstante con los procedimientos y técnicas estudiadas en capítulos anteriores se encontró que algunas de las etapas son de mayor importancia en la predicción resultante, mientras que otras etapas requieren un menor esfuerzo de desarrollo porque no influyen directamente en la calidad de la predicción del sensor.

Con respecto a la figura 6.1, en el presente documento se considera la etapa de inspección preliminar y la identificación de los estados estables como una sola etapa, que se denominará preparación del conjunto de datos, la cual incluye la primera inspección, identificación de datos fuera de tendencia obvios, la selección de los estados estacionarios del proceso y la eliminación de los estados apagados de la planta o proceso. La etapa de preprocesamiento de datos se considera como una etapa coincidiendo con la figura 6.1.

Figura 6.1 Fases de desarrollo de sensor inferencial. Tomado de (Kadlec, 2009)

La etapa de selección de modelo, entrenamiento y validación, se considera como 2 etapas separadas donde por un lado se tiene la selección de la estructura del modelo y por otro lado como una etapa independiente la de entrenamiento y validación; la razón para considerarlas por separado radica en que para cada procedimiento es necesario tomar decisiones de distinta naturaleza, debido a que la selección del modelo tiene que ver con la estructura que se asume que tienen los datos dentro del modelo, mientras que el entrenamiento y la validación se refiere a la selección de técnica de aprendizaje de máquina y a la selección adecuada de sus parámetros arquitecturales. En la figura 6.2 se presentan las etapas que se consideran fundamentales, es decir a las que se debe prestar especial atención e invertir mayor cantidad de tiempo con respecto a otras consideradas en trabajos como (Kadlec, 2009), en el desarrollo de un sensor inferencial para obtener una predicción lo más exacta posible.

Figura 6.2 Etapas fundamentales para el desarrollo de un sensor inferencial



Al comparar las figuras 6.1 y 6.2 se evidencia que no se tuvo en cuenta la etapa de mantenimiento del sensor inferencial. Aunque se trata de una etapa importante desde la óptica industrial, el mantenimiento del sensor es una actividad posterior al diseño e implementación del sensor inferencial y por lo tanto debería establecerse una metodología independiente para llevar a cabo esta tarea. En el mismo sentido, en (Kadlec et al., 2009) se consideran 2 vías para el mantenimiento, por un lado, el re-entrenamiento de la técnica de aprendizaje con el conjunto de datos nuevo y por otro lado la implementación del sensor inferencial a través de técnicas de aprendizaje adaptativas y evolutivas, las cuales están por fuera del alcance del presente documento. En otras palabras, el mantenimiento es una actividad independiente del diseño e implementación del sensor inferencial y por lo tanto no se considera una etapa para el diseño del mismo.

6.1.1 Preparación del conjunto de datos

Como se ha señalado a lo largo del documento, los datos que provienen de entornos industriales traen consigo una gran cantidad de características que se han tratado ampliamente. Teniendo esto en cuenta, antes de iniciar el trabajo con los datos es necesario llevar a cabo una serie de tareas que preparen el conjunto de datos para ser procesados por las diferentes técnicas de las etapas siguientes, dentro de esta lista de tareas se puede considerar lo siguiente:

- Inspección visual para detectar “outliers” que están altamente desviados y son fácilmente detectables.

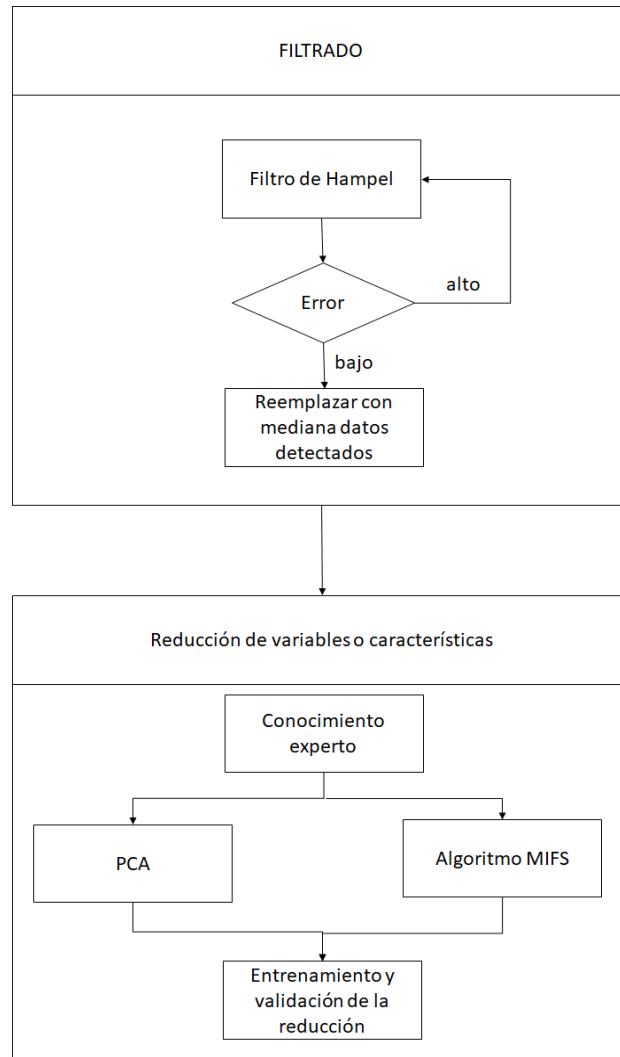
- Inspección visual para detectar pérdidas de datos en intervalos muy largos de tiempo. Los datos deben ser eliminados del conjunto asumiendo que en la industria se tienen cantidades grandes de datos y por lo tanto es mejor tomar series de tiempo donde no existan dichas pérdidas.
- A partir de la inspección visual también es posible identificar los rangos de tiempo donde se encuentra el proceso en estado estable, los transitorios y los apagados del proceso o la planta. El conjunto de datos debe contener preferiblemente solo información de estado estacionario, ya que el sensor inferencial típicamente estima variables de calidad de producto.

Después de realizar los ajustes anteriormente indicados al conjunto de datos obtenido del DCS de planta, se tendrá un conjunto con un cierto número de variables M y una cantidad de muestras N que caracterizan el comportamiento del proceso en estado estable y se podrán aplicar las técnicas exploradas a lo largo del documento.

6.1.2 Preprocesamiento de datos

Después de la preparación del conjunto de datos, aún no se tiene un conjunto de datos aptos para el entrenamiento y validación de la técnica de aprendizaje de máquina, debido a que en el conjunto persisten defectos como datos fuera de tendencia cuya desviación no es notoria frente al conjunto de datos, denominados datos fuera de tendencia no obvios, y el conjunto tiene muchas variables que tal vez, no son relevantes para la predicción de la variable e e incluso que pueden deteriorar la calidad de la predicción debido al denominado “*curse of dimensionality*” (Alpaydm, 2010).

Para el tratamiento de estos aspectos, es necesario aplicar algunas técnicas y algoritmos con el fin de acondicionar el conjunto de datos para que pueda ser procesado por cualquiera de las técnicas de aprendizaje de máquina seleccionadas. En la figura 6.3 se presenta el procedimiento propuesto para realizar el preprocesamiento de datos.

Figura 6.3 Esquema de metodología de preprocesamiento de datos

Tal como se observa en la figura 6.3, en el preprocesamiento de datos se consideran 2 procesos para acondicionar el conjunto de datos, por un lado, se realiza un proceso de filtrado de los datos con el fin de eliminar los datos fuera de tendencia e incluso el ruido si es necesario; por otro lado, se ejecuta alguna de las técnicas de reducción de datos para obtener un conjunto que contenga la información más relevante.

6.1.3 Selección de modelo

En el capítulo anterior se presentó una exploración de la selección de modelo, lo cual en pocas palabras consiste en dar una estructura a los datos de tal manera que la predicción obtenida sea la más exacta posible; allí se estableció, para el modelo de la columna de destilación binaria, que el modelo NARX es más recomendable debido a que provee una buena exactitud frente a la complejidad del modelo, por lo tanto, se descartó el modelo

NARMAX por la razón fundamental del esfuerzo adicional en tiempo y desarrollo que implica el entrenamiento y validación de una estructura NARMAX.

Así mismo, en esta etapa es necesario realizar una nueva selección de variables, con el fin de establecer cuáles de los auto-regresores son relevantes para mejorar la exactitud de la predicción y de esta manera eliminar el resto. Con esto se mejora el desempeño de la técnica de aprendizaje de máquina y evitar caer de nuevo en los problemas asociados a la dimensionalidad del conjunto de datos.

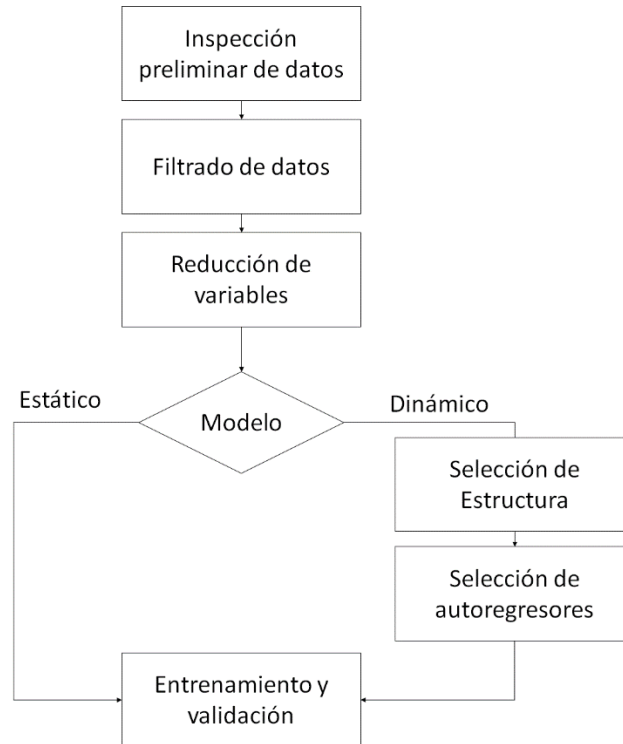
6.1.4 Entrenamiento y validación del modelo

Con el conjunto de datos adecuado, la estructura del modelo ya definido y los auto-regresores seleccionados se obtiene un conjunto de datos con las características mínimas para conseguir una predicción de buena calidad, mediante el entrenamiento de las técnicas de aprendizaje, ANN, SVM y ANFIS seleccionadas para el desarrollo del sensor virtual. Aplicando el método de validación cruzada⁴⁵ se puede obtener una medida del desempeño del aprendizaje de cada técnica (Ng, 2000) comprobando el comportamiento de la predicción frente a datos diferentes a los de entrenamiento. En diferentes trabajos referentes a sensores inferenciales (F. A. A. de Souza, 2014) se utiliza el método de “*cross-validation*” para establecer el desempeño de la predicción de la o las variables deseadas.

Así mismo, en la mayor parte de los trabajos referentes a sensores inferenciales consultados, se estableció que la métrica más utilizada para evaluar el desempeño de la predicción es el error medio cuadrático (MSE) como una medida general del comportamiento. En esta etapa deben afinarse los parámetros propios de las técnicas de aprendizaje de máquina, tratados en el capítulo 2, con el fin de llevar la predicción a la mayor exactitud y que el sensor produzca la medida más adecuada cuando entre a operar en línea.

⁴⁵ Conocido en los textos científicos como cross - validation

Figura 6.4 Metodología de diseño de un sensor inferencial basado en ANN, SVM o ANFIS



En la figura 6.4 se presenta un diagrama de flujo de la metodología que se propone a través de la exploración de cada una de las etapas que implica el desarrollo de un sensor inferencial que también puede ser traducido a la siguiente secuencia de pasos:

1. Realizar una inspección preliminar de los datos donde se detectan datos fuera de tendencia obvios, se identifican los estados estables, transitorios y apagados del proceso y se conforma un conjunto de datos inicial que contiene los estados estables de la planta tal y como fueron registrados por el DCS.
2. Realizar un preprocesamiento a los datos con el fin de adecuarlos para su utilización en el entrenamiento y validación del sensor inferencial resultante. En primer lugar, se debe realizar una tarea de filtrado de los datos para detectar los datos fuera de tendencia; se recomienda el filtro de Hampel con una adecuada selección de parámetros, con el fin de que aplique para cualquier conjunto de datos. Posteriormente debe realizarse la reducción de características o variables a través de selección o extracción de características; en el capítulo 4 se exploran los métodos de análisis de correlación, el algoritmo MIFS y análisis de componentes principales respectivamente que pueden resultar suficientes en conjuntos como el que se trata en el presente documento, sin embargo, existen reportados algoritmos más robustos para el tratamiento de conjuntos de datos con redundancia de variables o características co-lineales.
3. Seleccionar la estructura del modelo de datos, en el presente documento se consideró NARX y NARMAX, las cuales son estructuras para modelos no lineales que procesan datos de instantes anteriores para predecir el valor futuro de la

variable. Se recomienda realizar un análisis de desempeño bajo ambas estructuras para poder establecer el mejor desempeño del sensor.

4. Con el modelo definido es necesario hacer una selección de los auto-regresores que son relevantes para la predicción, esto reducirá la dimensión del conjunto obtenido al ajustar la estructura de los datos y liberará recursos de computo en el entrenamiento y validación del sensor. Para este fin resultan muy adecuados los métodos “*wrapper*” para asegurar un buen desempeño de predicción
5. Con el conjunto de datos obtenido después de realizar todo el procedimiento anterior, se procede dividir el conjunto de datos para aplicar la validación cruzada; se recomienda el primer 70% de la serie de tiempo para entrenamiento y el 30% restante para validación. Con esto, se realiza el entrenamiento de las técnicas de aprendizaje de máquina hasta encontrar convergencia al MSE más bajo posible.
6. Posteriormente se valida la predicción con el fin de evaluar la exactitud del sensor inferencial, para el caso en que la medida no tenga un buen desempeño se varían los parámetros correspondientes en las técnicas de aprendizaje y se repetirá desde el paso 5 hasta encontrar el comportamiento más adecuado para la medida de la variable de interés.

La metodología anteriormente descrita, es el resultado de la exploración de las técnicas más clásicas para el procesamiento de los datos en cada una de las etapas, sin embargo, hay múltiples trabajos reportados en aspectos como sensores inferenciales con datos fuera de tendencia, pérdida de datos, otras estructuras de datos, selección de auto-regresores, etc., los cuales aplican o proponen técnicas mucho más robustas para llevar a cabo las tareas antes mencionadas y con resultados de mejor desempeño (Bella et al., 2007; Garcia, de Carvalho Berni, & de Oliveira, 2008; Park & Han, 2000). No obstante, el alcance del presente trabajo no cubre un universo tan amplio de técnicas para el diseño de sensores inferenciales, razón por la cual las técnicas más clásicas constituyen una referencia para ilustrar el procedimiento y etapas de diseño de un sensor virtual.

6.2 Sensores inferenciales obtenidos

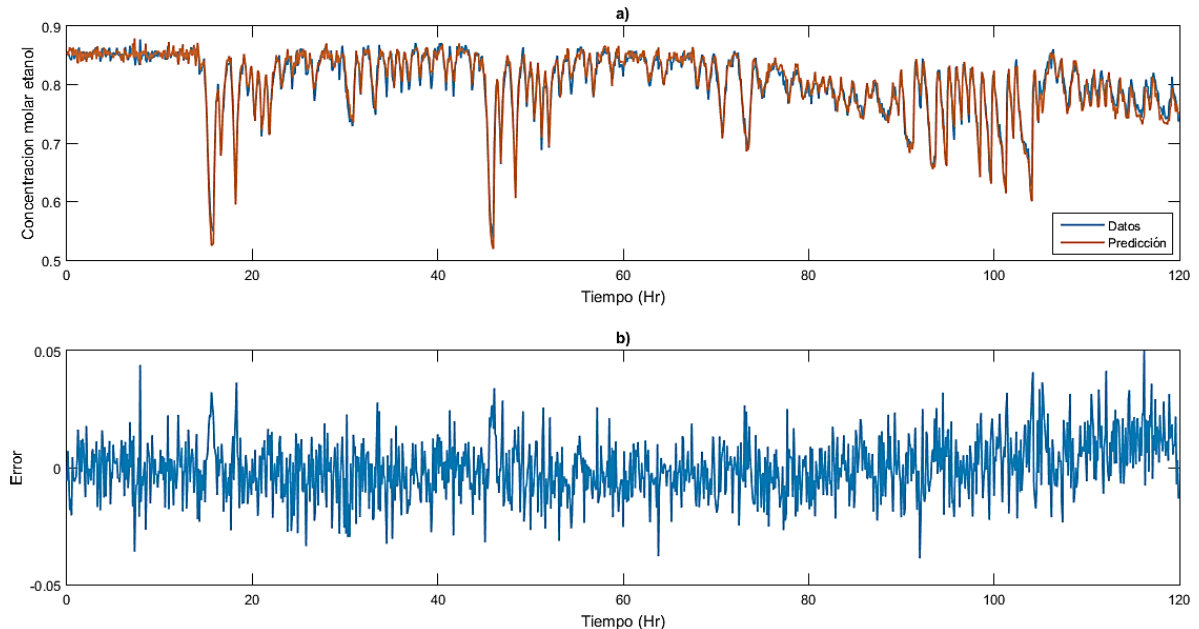
Después de seguir la metodología que se detalló en la subsección anterior se obtuvieron los siguientes resultados para cada una de las técnicas de aprendizaje de máquina seleccionadas. Se utilizó un conjunto de datos que corresponde a una columna de destilación binaria para la separación de una mezcla agua-etanol. Los datos se obtuvieron del modelo matemático que se trató en el capítulo 3 y se les agregó ruido de baja amplitud para simular un escenario industrial típico, donde se supone un sistema de instrumentación que está funcionando bajo una calibración adecuada.

6.2.1 Sensor Inferencial basado en ANN

Teniendo en cuenta lo anterior, se desarrolló un sensor inferencial aplicando el algoritmo MIFS, el cual utiliza teoría de la información para la reducción de características, se utiliza modelo NARX para la estructura de los datos y se entrena y valida el sensor virtual por medio de una red neuronal artificial, con una capa oculta, 20 neurodos en la capa oculta; se entrena por medio del algoritmo de retro-propagación descrito en el capítulo 2.

En la figura 6.5 es posible observar que el sensor inferencial obtenido mediante redes neuronales artificiales tiene una predicción adecuada ya que en la figura 6.5.a) se evidencia que el sensor predice los cambios de concentración de acuerdo a las variaciones en la entrada. En el mismo sentido, la figura 6.5.b) permite apreciar que el error máximo es de 0.06 con respecto a los datos del conjunto de validación lo cual es una predicción bastante exacta en este tipo de procesos.

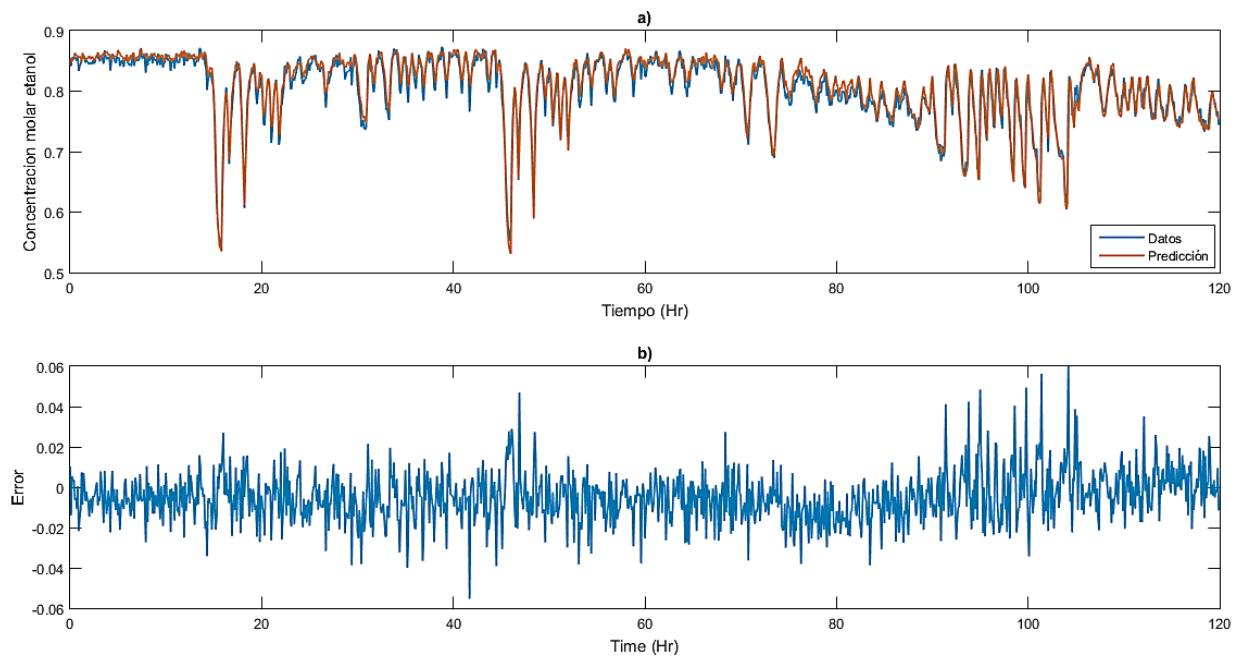
Figura 6.5 Resultados de validación de sensor inferencial ANN. a) Validación concentración de etanol. b) Error residual conjunto de datos de validación y predicción sensor inferencial



6.2.2 Sensor Inferencial basado en SVM

Se desarrolló un sensor inferencial aplicando análisis de correlación para la reducción de características, se utiliza modelo NARX para la estructura de los datos y se entrena y valida el sensor virtual por medio de una máquina de vectores de soporte, con kernel gaussiano; se entrena por medio del algoritmo Kernel-Adatron descrito en detalle en el capítulo 2.

Figura 6.6 Resultados de validación de sensor inferencial SVM. a) Validación concentración de etanol. b) Error residual conjunto de datos de validación y predicción sensor inferencial

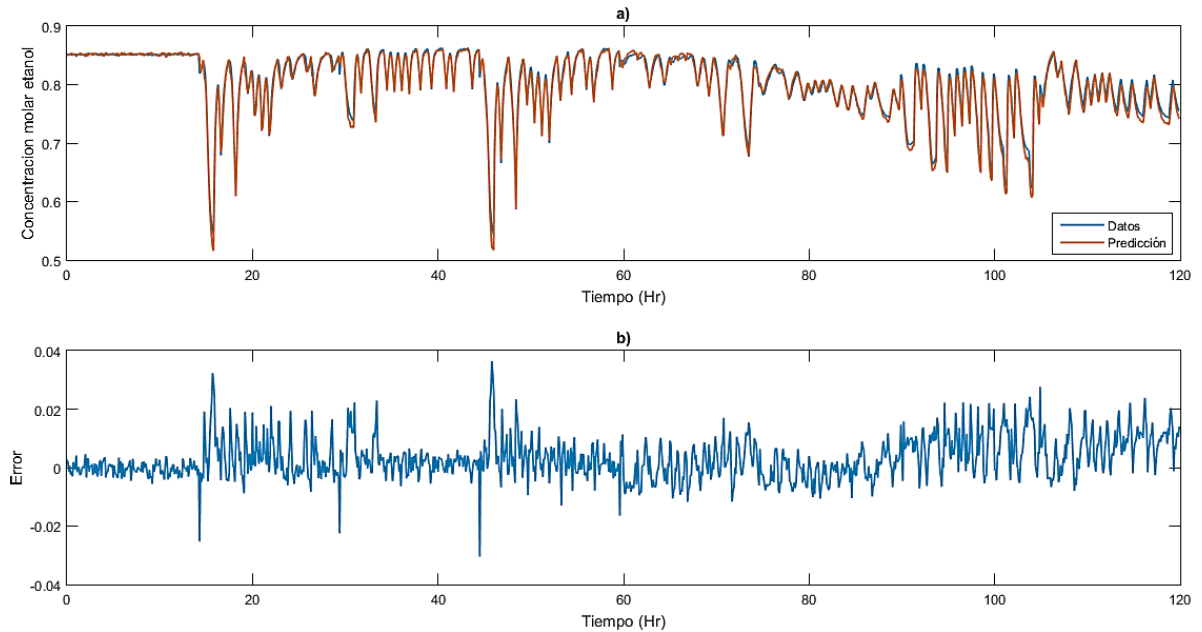


En la figura 6.6 se observa que el sensor inferencial implementado con SVM de kernel gaussiano, responde adecuadamente a la dinámica del proceso, frente a variaciones de diferentes tipos en las entradas. Así mismo, la predicción tiene una exactitud suficiente pues su error máximo es de 0.14 pero su error promedio es muy cercano a cero, el cual se puede verificar en la figura 6.6.b).

6.2.3 Sensor Inferencial basado en ANFIS

Se desarrolló un sensor inferencial aplicando análisis de correlación para la reducción de características, se utiliza modelo NARX para la estructura de los datos y se entrena y valida el sensor virtual por medio de un sistema ANFIS, con una base de 4 reglas obtenidas mediante el algoritmo de clustering; se entrena por medio del algoritmo de retro-propagación descrito en el capítulo 2.

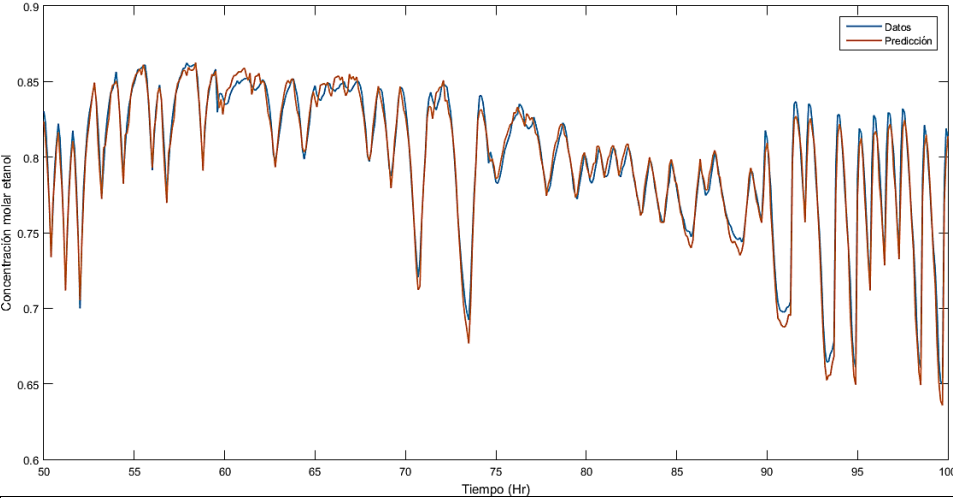
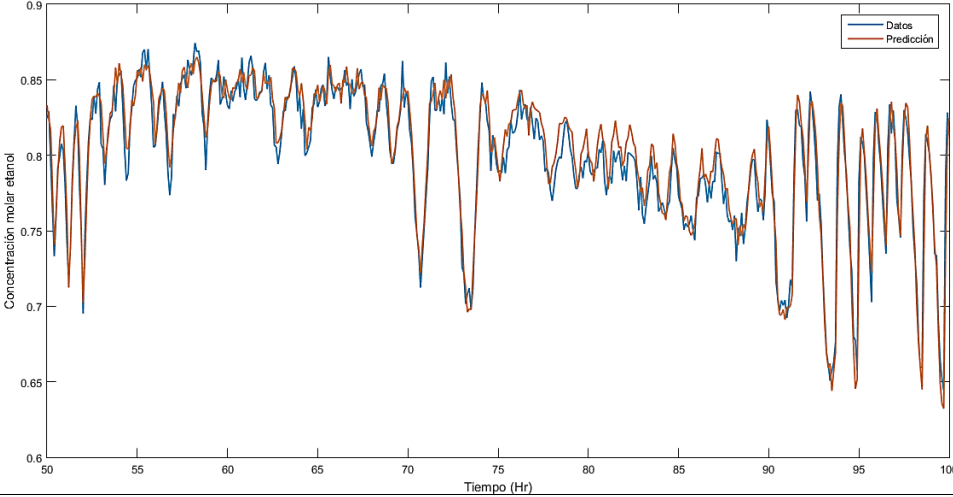
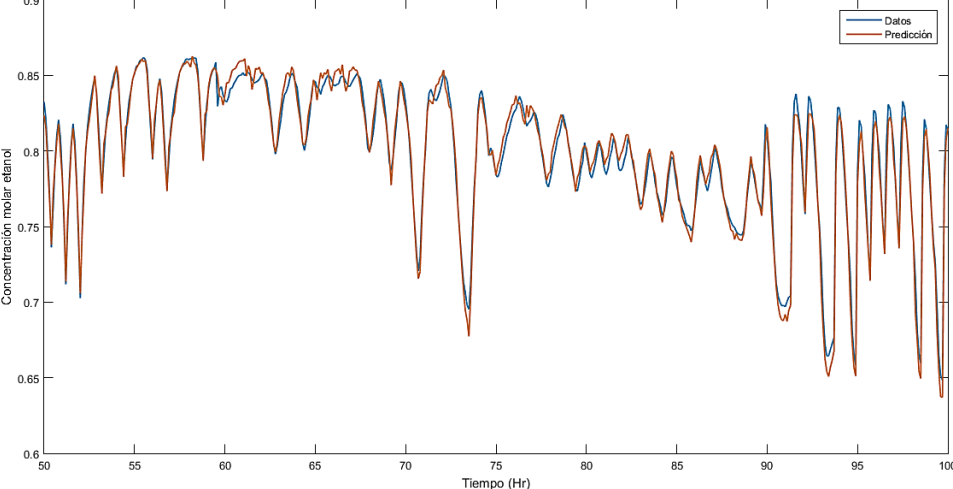
Figura 6.7 Resultados de validación de sensor inferencial ANFIS. a) Validación concentración de etanol. b) Error residual conjunto de datos de validación y predicción sensor inferencial



De acuerdo con la figura 6.7.a) el sensor inferencial desarrollado con ANFIS, se observa que la predicción lograda es adecuada con 6 reglas y está en el mismo rango que el resultado obtenido con las otras técnicas. Incluso, el error que se muestra en la figura 6.7.b) está comprendido en un intervalo menor que en los casos anteriores.

En la tabla 6.7 se presenta mayor detalle de la validación de los sensores, reduciendo el espacio de observación a partir de la hora 50 y hasta la hora 100. Así mismo, el error medio cuadrático que corresponde a cada uno de los sensores desarrollados, con el fin de poder tener un marco de comparación más detallado entre los sensores obtenidos con cada técnica.

Tabla 6.1 Comparativo desempeño sensores inferenciales basados en ANN, SVM y ANFIS

ANFIS		MSE=2.0745e-4
SVM		MSE=6.7709e-4
ANN		MSE=9.7948e-4

6.3 Pruebas de desempeño

La sección anterior confirma que por medio de la metodología desarrollada en el presente capítulo es posible llevar acabo el diseño de sensores inferenciales para cualquiera de las tres técnicas seleccionadas, ANN, SVM y ANFIS; teniendo en cuenta que los resultados obtenidos fueron satisfactorios en lo que respecta a la exactitud de la predicción en el sensor y al esfuerzo en cada una de las etapas de la metodología propuesta. A continuación, se presenta un análisis de desempeño de los sensores diseñados frente a características de los datos como datos fuera de tendencia, muestreo insuficiente y mínimo número de muestras para obtener una predicción aceptable.

6.3.1 Datos fuera de tendencia

En el capítulo 2 se estableció que las 3 técnicas se ven altamente afectadas ante la existencia de datos fuera de tendencia en el conjunto de datos, así mismo, en el capítulo 4 se exploraron múltiples técnicas de filtrado para la detección de los denominados “*outliers*”, encontrando que el filtro de Hampel es la técnica más adecuada para realizar esta tarea, por lo tanto, se realizan pruebas con conjuntos que tienen 0%, 25% y 50% de datos fuera de tendencia dentro del conjunto total. Los resultados que se muestran a continuación son obtenidos agregando “*outliers*” y ruido con una amplitud del 10% del rango de los datos.

Tabla 6.2 MSE de validación con conjuntos contaminados con datos fuera de tendencia

Cantidad	ANN		SVM		ANFIS	
	Filtro	Sin filtro	Filtro	Sin filtro	Filtro	Sin filtro
0%	3.0775e-4	3.0775e-4	6.9435e-4	6.9435e-4	1.8933e-4	1.8933e-4
5%	6.7579e-4	0.0010	0.0017	0.0042	1.0786e-4	0.0112
10%	0.0011	0.0015	0.0020	0.0024	8.7030e-5	0.0036
15%	0.0015	0.0030	0.0053	0.0075	8.3557e-5	0.0042
25%	0.0021	0.0064	0.0027	0.0057	1.0388e-4	0.0052

De la tabla 6.2 es posible observar que incluir tareas de filtrado de datos fuera de tendencia como parte del preprocesamiento de los datos es adecuado y mejora el desempeño en la estimación del sensor inferencial. Así mismo, los sistemas ANFIS presentan un desempeño muy adecuado después de realizar el filtrado de los datos, obteniendo errores hasta 10 veces menores que los obtenidos con ANN y SVM. Lo anterior sugiere que en conjuntos de datos con altas cantidades de patrones fuera de tendencia, ANFIS resulta

con predicciones más exactas. ANN y SVM son recomendables en datos con bajas cantidades de “outliers”, casos en los cuales las 3 técnicas presentan comportamientos similares frente a datos filtrados a través del filtro de Hampel.

6.3.2 Muestreo Insuficiente

En los entornos industriales se presentan diferencias de muestreo en los diferentes sensores y tarjetas de adquisición del DCS, razón por la cual es típico recurrir a la tasa de muestreo más lenta para poder obtener un conjunto de datos útil para el entrenamiento y validación de la técnica de aprendizaje (Casali et al., 1998; Kadlec et al., 2009; Sliškovic, Grbic, & Hocenski, 2011b). Con esto, es importante conocer la capacidad del sensor inferencial para llevar a cabo una predicción de una calidad adecuada frente a conjuntos de datos con pocas muestras.

Para llevar a cabo esta prueba, se tomará el conjunto de la columna de destilación, el cual contiene 4000 muestras y se reducirá de manera uniforme, emulando conjuntos que han sido adquiridos con una tasa de muestreo menor; se llevará a cabo la comparación entre las técnicas de aprendizaje seleccionadas frente a conjuntos con el 100%, 50%, 25% y 10%.

Tabla 6.3 MSE de validación frente a conjuntos con pocas muestras

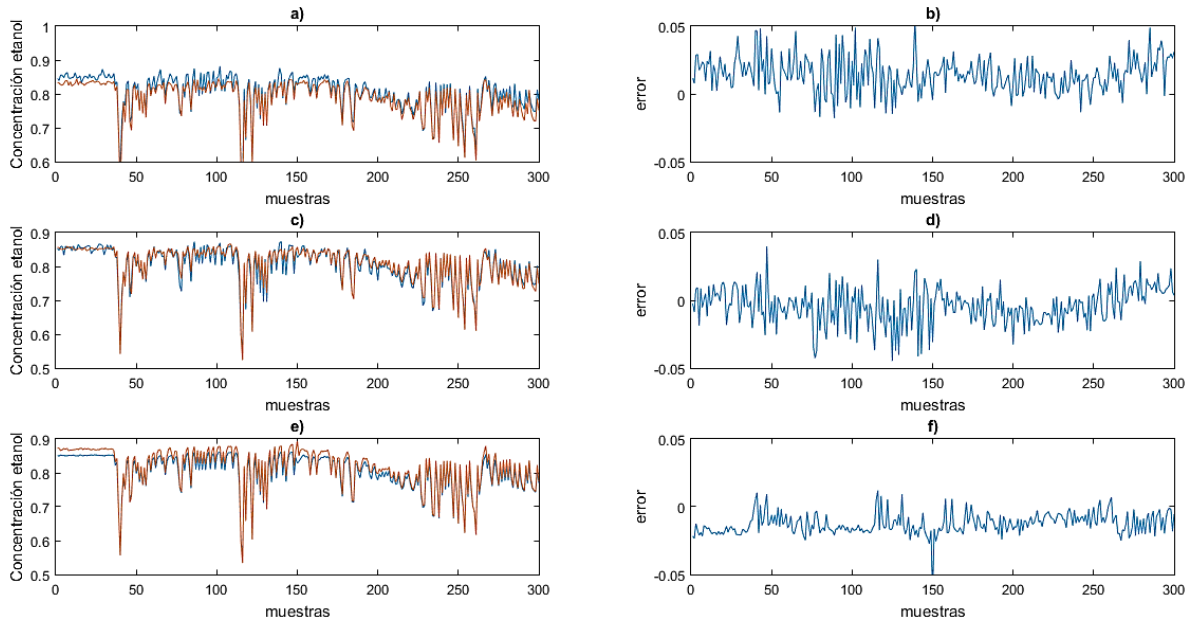
Cantidad	ANN	SVM	ANFIS
100%	2.5946e-4	2.0821e-4	1.9630e-4
50%	4.2624e-4	1.8791e-4	2.7535e-4
25%	2.9312e-4	1.8921e-4	2.8300e-4
10%	2.4816e-4	2.7955e-4	6.0385e-4

Tal como se evidencia en la tabla 6.2, el sensor inferencial, desarrollado a través de la técnica ANFIS, es el que presenta mayor influencia ante conjuntos con muestras insuficientes ya que en el peor caso se triplica el error de validación frente al conjunto con el 100% de datos; en contraste, SVM y ANN presentan una robustez mayor ya que la variación no es tan dramática y con un conjunto de 400 muestras la variación esta alrededor del 20% lo cual es significativamente menor al caso de ANFIS.

En la figura 6.8 se observa el comportamiento de la predicción de cada una de las técnicas frente al conjunto que solo conserva el 25% de las muestras. En dicha figura se observa

que el menor error corresponde a los sistemas ANFIS frente a las ANN y SVM, las cuales exhiben un desempeño inferior con un error máximo de predicción de ± 0.05 . Lo anterior indica que ANFIS puede ser recomendable en conjuntos de datos con una baja tasa de muestreo, sin embargo, las redes neuronales y las máquinas de vectores de soporte tienen un desempeño aceptable en variables cuya predicción admita un nivel de error bajo

Figura 6.8 Validación de sensores frente a conjunto del 25% del total. a) Validación ANN, b) Error residual de predicción para ANN c) Validación SVM, d) Error residual de predicción para SVM, e) Validación ANFIS, f) Error residual de predicción para ANFIS

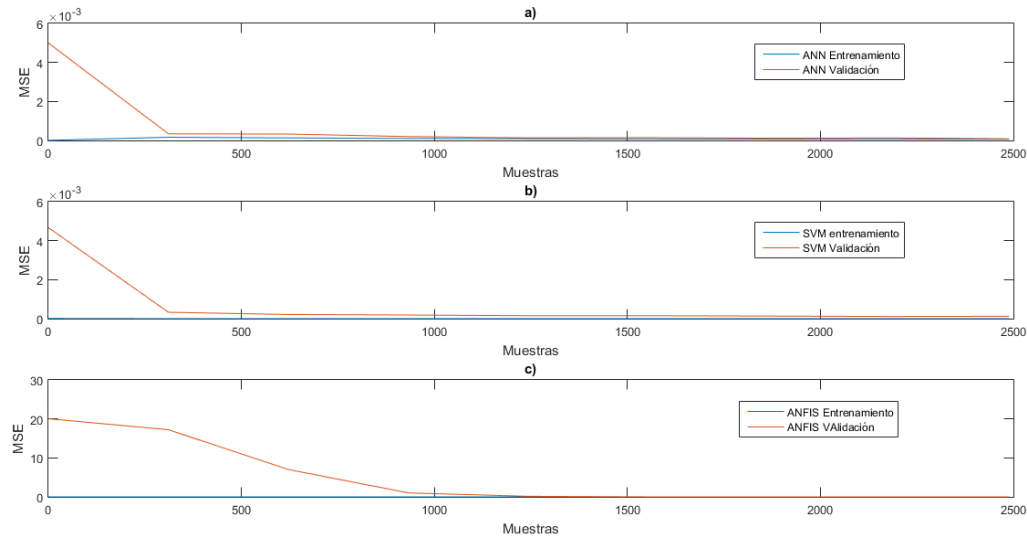


6.3.3 Mínimo número de datos de entrenamiento

Los sensores inferenciales tienen un requerimiento mínimo de datos para llevar a cabo un entrenamiento que conlleve a una predicción con un buen desempeño. La presente subsección pretende establecer el número mínimo de datos que son necesarios para el entrenamiento de los sensores inferenciales, ya que este es un parámetro de interés para el diseñador, si se quiere establecer un conjunto de datos adecuado para entrenar el sensor.

Se lleva a cabo un experimento, donde se entrenará sucesivamente el sensor inferencial aumentando progresivamente el número de muestras del conjunto de datos de entrenamiento hasta llegar al total, que para este caso es el conjunto de la columna de destilación con 4000 muestras. Se aplica a cada una de las técnicas seleccionadas con los siguientes resultados.

Figura 6.9 Comportamiento del MSE de validación para conjuntos con diferentes números de muestras. a) ANN, b) SVM, c) ANFIS



En la figura 6.9 se aprecia como varía el MSE de cada una de las técnicas con el aumento de muestras en el conjunto de datos. En la tabla 6.3, se relacionan los tamaños de conjunto mínimo para obtener una predicción aceptable.

Tabla 6.4 Cantidades mínimas de datos para conseguir una predicción adecuada

	MSE Validación	Cantidad Muestras
ANN	1.9694e-4	1000
SVM	1.8648e-1	1000
ANFIS	4.9148e-4	2000

De la tabla 6.4 se puede observar que, para el conjunto de datos del proceso de destilación, tratado en el capítulo 3, las técnicas requieren entre 1000 y 2000 muestras para poder realizar una predicción adecuada. Así mismo, se identifican los sensores inferenciales basados en ANFIS como los más sensibles a esta característica, por lo tanto, si la colección de datos no ha tenido suficiente tiempo para recopilar la dinámica del proceso completa se sugiere evitar el uso de ANFIS.

6.4 Conclusiones

Después de haber realizado el diseño, entrenamiento y validación de tres sensores inferenciales basados en datos, a través de las técnicas de aprendizaje, ANN, SVM y ANFIS y de establecer una metodología para tal fin se concluye lo siguiente:

- A partir de la metodología planteada en el presente capítulo, es posible diseñar sensores inferenciales para variables presentes en proceso de orientación industrial. Garantizando un adecuado desempeño para la estimación de variables relacionadas con la calidad de producto, en este caso la concentración de etanol en un proceso de separación de una mezcla etanol-agua.
- Las etapas propuestas en la metodología planteada son los requerimientos mínimos para llevar a cabo el diseño de sensores inferenciales por medio de las técnicas seleccionadas. En la literatura existen reportes del uso de múltiples técnicas para el manejo de inconvenientes de los conjuntos de datos, tales como, pérdida de datos, datos fuera de tendencia, ruido, etc., sin embargo, esto resulta complementario para mejorar el desempeño de los sensores en casos donde la exactitud requerida es muy alta.
- Los sensores inferenciales diseñados, a partir de la metodología propuesta en este capítulo, presentan un buen desempeño, con errores de predicción bajos, por lo cual la metodología fue suficiente. En caso de requerirse mayor exactitud o una mejora en el desempeño frente a los problemas típicos de los datos industriales, es posible agregar las técnicas que sean necesarias a la metodología propuesta, puesto que fue estructurada con los requerimientos mínimos para obtener un sensor inferencial funcional.
- Entre los sensores inferenciales diseñados, los elaborados con máquinas de vectores de soporte presentan mayor robustez frente a datos con frecuencia de muestreo baja; esto debido fundamentalmente a su propia naturaleza ya que al usar la función de pérdidas ϵ -intensiva está incorporando robustez frente a estos aspectos.
- De las tres técnicas exploradas y aplicadas en los sensores inferenciales, la que presenta un comportamiento más homogéneo frente a los problemas típicos de los datos industriales son las ANN, en otras palabras, no cambia sus niveles de predicción frente a cantidades de muestras, ruido, cantidad de características, etc. De las pruebas realizadas, el sensor inferencial basado en ANN presentó degradación drástica frente a conjuntos con datos fuera de tendencia, los aspectos restantes se manejaron mediante el ajuste de los parámetros propios de la red neuronal.
- Los sensores inferenciales basados en ANFIS, presentan una mejora en el comportamiento al aplicar la estructura NARX, obteniendo robustez frente a datos con bajo muestreo y un mejor comportamiento al entrenar con datos filtrados. A lo largo del desarrollo de la metodología, se encontró que los sistemas ANFIS tienen unos requerimientos altos de calidad de los datos, es decir, para obtener estimaciones con muy bajo error es indispensable contar con un conjunto de datos con el mínimo de datos fuera de tendencia, el mínimo ruido, la mayor cantidad de muestras posible y que el conjunto tenga la información suficiente del proceso para

poder estimar la dinámica del mismo. Así mismo es muy recomendable el uso de estructuras como NARX o NARMAX para mejorar el desempeño en la predicción de los sensores basados en ANFIS. De lo anterior, se establece que el diseño de un sensor virtual por medio de ANFIS requiere un esfuerzo adicional en el preprocesamiento de los datos, por ende, mayores costos monetarios y de tiempo.

Para resumir, con la metodología establecida es posible diseñar sensores virtuales basados en ANN, SVM y ANFIS, con un desempeño adecuado para proceso industriales. Por otro lado, los sensores inferenciales basados en ANN y SVM son adecuados para los entornos industriales donde la calidad de los datos es baja y se requiere del mínimo esfuerzo tanto en desarrollo como en tiempo. Los sensores basados en ANFIS requieren de alta calidad en los datos.

7. Conclusiones

Los sensores inferenciales constituyen una excelente herramienta, ante variables de difícil o impráctica medición en procesos industriales, dichas variables en su mayoría tienen relación con la calidad de los productos de cada proceso, razón por la cual existe un alto interés en conocer estas medidas en tiempo real con el fin de monitorear o incluso llevar a cabo tareas de control directamente sobre los factores que influyen en la calidad de los productos. El presente capítulo pretende dar una mirada al trabajo desarrollado a lo largo del documento, las etapas y el procedimiento para completarlas y por último un establecimiento de las perspectivas futuras que tiene el trabajo.

7.1 Resumen del proyecto

El objetivo general del presente proyecto era la consistía en el establecimiento de una metodología de diseño e implementación de sensores inferenciales basados en datos bajo las técnicas de aprendizaje de máquina, ANN, SVM y ANFIS, a partir de un comparativo de desempeño entre 3 sensores basados en dichas técnicas.

Para la consecución de este objetivo se realizó una extensa búsqueda de literatura donde se reportan sensores inferenciales desarrollados bajo las técnicas antes mencionadas, con el fin de establecer el estado del arte, los ámbitos y sectores de la industria donde se han aplicado y las consideraciones en cuanto a condiciones de datos, modelo y entrenamiento y convergencia. A partir de esta exploración se logró establecer unas etapas mínimas para el diseño e implementación de un sensor inferencial, las cuales comprenden el preprocesamiento de datos, la selección del modelo y por último el entrenamiento y validación de la técnica de aprendizaje que se esté usando.

Con el estado del arte establecido y algunos rasgos metodológicos identificados se procedió a la generación de un conjunto de datos para el entrenamiento y validación de los sensores, para lo cual, se implementó a nivel de simulación, el modelo matemático de una columna de destilación binaria para la separación de una mezcla de agua-etanol, obteniéndose un conjunto de datos de 4000 y 60 variables correspondientes a presión temperatura y flujo, las cuales serían fácilmente medibles en un entorno industrial. Luego, se procedió a realizar el preprocesamiento de los datos, mediante la detección de datos fuera de tendencia y la reducción de características del conjunto de datos, etapa en la cual se identificaron múltiples técnicas de las cuales se utilizaron las más clásicas. Seguido a

esta etapa, lo siguiente fue la selección y aplicación de una estructura de modelo dinámico como NARX, NARMAX u otra que permita dar un mejor desempeño al sistema frente a la dinámica no lineal que posee el proceso seleccionado. Por último, se procedió al entrenamiento y validación de cada una de las técnicas seleccionadas, ajustando sus respectivos parámetros y desarrollando un comparativo de desempeño frente a inconvenientes típicos de los sensores inferenciales.

La metodología que se propone en el presente documento, capítulo 6, se estableció a partir de la exploración de múltiples técnicas, análisis paramétricos y comparativos en cada una de las etapas buscando el mejor equilibrio entre tiempo de desarrollo, conocimiento y desempeño del sensor inferencial.

7.2 Conclusiones generales

En cuanto a las técnicas seleccionadas, es decir, ANN, SVM y ANFIS, se encontró mayor dificultad en la implementación de los sistemas ANFIS debido a la poca documentación y código disponible. Principalmente en los aspectos de la implementación del algoritmo de clustering y la disposición de las matrices con las cuales se ejecuta la estimación de mínimos cuadrados que permite obtener los parámetros Takagi-Sugeno del consecuente. Así mismo después de realizar el análisis de desempeño de las técnicas ante diferentes aspectos se encontró que las arquitecturas tienen parámetros con cierta equivalencia a pesar de que cada técnica tiene una naturaleza propia y un método totalmente diferente para la solución del problema de regresión.

Por otro lado, los resultados obtenidos en el diseño del sensor inferencial presentado en el capítulo 6, indican que ANN y SVM presentan un mejor desempeño no solo en la validación de los datos generados inicialmente en el conjunto, sino también, frente a problemas de calidad en los datos como datos fuera de tendencia. En contraste ANFIS, presenta un desempeño muy inferior cuando es entrenado con datos de baja calidad, y al ser esta una característica de los datos provenientes de la industria lo hace el menos apto de los 3 para ser utilizado en el desarrollo de un sensor inferencial.

De igual forma, en el trabajo se estableció que ninguna de las 3 técnicas, por si sola, es más apta para ser utilizada en un sensor inferencial, ya que requiere de una serie de técnicas para el acondicionamiento de los datos y metodologías de entrenamiento que permitan tener un desempeño adecuado para los requerimientos de los procesos con orientación industrial.

La metodología desarrollada se considera apta para su aplicación en ambientes industriales reales, debido a que cada una de las etapas identificadas en el diseño de los sensores inferenciales fue cuidadosamente tratada, de tal manera que el escenario de simulación tuviera los elementos típicos de un proceso industrial, en razón a esto se generó

el conjunto de datos de entrenamiento y validación a partir de un modelo matemático de una columna de destilación binaria, considerando únicamente variables que son medidas típicamente en plantas de este tipo. Así mismo, todas las pruebas de validación de los sensores inferenciales diseñados se realizaron con los datos del modelo matemático, adicionando ruido, datos fuera de tendencia y emulando diferentes situaciones que se presentan de manera frecuente en procesos industriales.

Por último, la selección de la técnica de aprendizaje para el desarrollo del sensor depende de varios factores, entre ellos la calidad de los datos que se tiene y el tipo de proceso que se está modelando, ya que como se estableció en el capítulo 2, su desempeño varía ante dinámicas muy complejas en los conjuntos de datos, produciendo degradación en la estimación.

7.3 Trabajo futuro

Con el desarrollo del presente trabajo, han surgido varios posibles trabajos de investigación para el futuro. Algunos de estos son:

- **Técnicas de selección de características y auto-regresores**
En el capítulo 4 se mencionó la existencia de técnicas más robustas en la reducción de variables de los conjuntos de datos, como los coeficientes de información o incluso técnicas denominadas “*wrapper*”, las cuales pueden producir una selección de características con mayor certeza. Teniendo esto en cuenta un futuro trabajo está en la exploración de técnicas para la selección de variables que reduzcan la incertidumbre en la eliminación de las variables y que contribuyan de una manera más contundente en el desempeño del sensor inferencial.
- **Técnicas de reconstrucción de datos**
En el capítulo 2, se realizaron pruebas de desempeño de las técnicas ante pérdidas de datos y se encontró una drástica degradación en las predicciones de las mismas. Debido a que este fenómeno se presenta en procesos industriales sería útil la exploración de técnicas de reconstrucción de datos en sensores virtuales con variables que tienen intervalos de datos perdidos.
- **Selección de modelo**
En el capítulo 5 se realizó una exploración del uso de modelos dinámicos como NARX y NARMAX y pruebas de desempeño entre ambos, sin embargo, un trabajo de investigación de importancia en el área de los sensores inferenciales abordaría una metodología en la selección del modelo más apropiado frente a aspectos como la técnica de aprendizaje seleccionada, tamaño del conjunto de datos, entre otros.
- **Sensores Evolutivos**
Aunque los sensores inferenciales son de amplio uso, cargan con la desventaja de perder exactitud en la estimación de la variable de interés ante fenómenos dinámicos externos al proceso como clima, desgaste de piezas en la planta o cambios en las materias primas, lo cual desemboca en una modificación del modelo que se está prediciendo. Un trabajo interesante, podría abordar el establecimiento

de una metodología para el desarrollo de sensores inferenciales bajo técnicas evolutivas como sistemas difusos evolutivos, algoritmos genéticos u otra.

8. Anexos

8.1 Código Redes Neuronales Artificiales (ANN)

```
%*****%
%          ANN algoritmo de retro-propagación          %
%          - MLP 1 salida                             %
%          - Función de activación lineal en la salida %
%          - Normaliza valores de entrada y salida   %
%          - Función de activación sigmoidal en la capa oculta%
%      Argumentos:                                     %
%      * traindataset: conjunto de datos de entrenamiento %
%      * testdataset : conjunto de datos de validación   %
%      * n (eta): tasa de aprendizaje                   %
%      * M          : Numero de neurodos en la capa oculta %
%      * I          : Numero de iteraciones             %
%      * alpha     : pendiente de cambio en la función sigmoidal %
%*****%
function [traindataset yHat yHat_val etotal e
etotal_val]=annBackprop(traindataset,testdataset,n,M,I,alpha)
%Normalización de conjuntos de entrenamiento y validación
norm_dataset=normaliz(traindataset,'ann-maxmin');
norm_dataset_val=normaliz(testdataset,'ann-maxmin');
datasetsize=size(traindataset);
%Cantidad de muestras
Q=datasetsize(1);
%create dataset input
a_norm=norm_dataset(:,1:size(norm_dataset,2)-1);
%Se agrega terminos de bias
X=[a_norm zeros(Q,1)]; %ones bias term
val_X=[norm_dataset_val(:,1:end-1) ones(size(norm_dataset_val,1),1)];
%Conjunto de salida
Y=traindataset(:,size(traindataset,2));
inputLayerSize=size(X,2);
outputLayerSize=size(Y,2);
hiddenLayerSize=M;
i=1;
etotal(1)=1;
%*****Retropropagación*****%
%inicialización aleatoria de pesos +-0.05
%wnm pesos capa oculta
wnm = -0.05 + (0.05-(-0.05)).*rand(inputLayerSize,hiddenLayerSize);
%um pesos capa salida
um= (-0.05 + (0.05-(-0.05)).*rand(hiddenLayerSize+1,1));
```

```

while(etotal(i)>0.00001)
    i=i+1;

    for q=1:size(X,1)
        %feedforward
        z2=X(q,:)*wnm ;
        a2=[1./(1+exp(-(alpha*z2))) 1];
        z3=a2*um ;
        yHat(q)=z3 ;

        %tamaños de matrices
        %z2=>1xM
        %a2=>1xM+1
        %z3=>1x1
        %yHat=>1x1

        %Backpropagation
        delta_o=yHat(q)-Y(q) ;
        gradiente_o=delta_o*a2 ;

        %delta_o=>1x1
        %gradiente_o=>1xM

        delta_h=delta_o*um'.*a2.*(1-a2);
        gradiente_h=(delta_h(1:hiddenLayerSize) '*X(q,:) ) ;
        %delta_h=>1xM+1

    %gradiente_h=>MxN
    %Actualizacion de pesos
    wnm=wnm-(n*gradiente_h');
    %traindataset=wnm;
    um=um-(n*gradiente_o');
    end

    %Error acumulado por iteraciones entrenamiento
    etotal(i)=(1/Q)*sum((yHat'-Y).^2);
    etotal(i)

    %Error acumulado de validacion
    for q=1:size(testdataset,1)
        %feedforward para validación
        z2_val=val_X(q,:)*wnm ;
        a2_val=[1./(1+exp(-(alpha*z2_val))) 1];
        z3_val=a2_val*um ;
        yHat_val(q)=z3_val ;

        %z2=>1xM
        %a2=>1xM+1
        %z3=>1x1
        %yHat=>1x1
    end

    etotal_val(i)=(1/size(testdataset,1))*sum((yHat_val'-
testdataset(:,end)).^2);
    if(i==I)
        etotal(i)=0;
    end
end

%error final
e=(yHat'-Y).^2;
end

```


8.2 Código Sistemas de Inferencia ANFIS

```

%*****ANFIS*****%
%           Algoritmo de inferencia difusa ANFIS           %
%           con consecuente tipo Takagi-sugeno entrenado por %
%           algoritmo de retro-propagación                 %
%   Argumentos:                                           %
%   * dataset      : Conjunto de datos de entrenamiento   %
%   * dataset_val  : Conjunto de datos de validación     %
%   * epochs       : Cantidad máxima de iteraciones      %
%   * ra           : Radio de clustering                 %
%   * n            : Tasa de aprendizaje  $\eta$  (eta)      %
%*****%

function [MF,output2,traindataset, O5,O5_val, etotal,etotal_val, e
]=anfis_algo(dataset,dataset_val,epochs,ra,n)
%tamaño de conjunto de datos
traindataset=dataset;
datasetsize=size(traindataset);
Q=datasetsize(1);
input=traindataset(:,1:end-1)';
output=traindataset(:,end)';
input_val=dataset_val(:,1:end-1)';
%*****Función auxiliar para clustering
substractivo*****%
[xk_ast, pk_ast, sigma1]=
full_subtractive_clustering(input,output,ra,0.5,0.15);
%Normalización de conjuntos de datos
input=normaliz(input,'maxmin');
input_val=normaliz(input_val,'maxmin');
datos=size(input,2);
variables=size(input,1);
%Cantidad de reglas
MF=size(pk_ast,1);
%*****Ancho de funciones de pertenencia*****%
%tomado de extracting fuzzy rules from data (chiu)
for i=1:MF
sigma(i,:)=sigma1;
end
%inicialization
O1=zeros(MF,variables);
O2=zeros(MF,variables);
O3=zeros(MF,datos);
dO2_dO1=zeros(MF,variables,datos);
dE_dc=zeros(MF,variables,datos);
dE_dsigma=zeros(MF,variables,datos);
etotal=zeros(1,epochs);
n1=n;
n2=n;
iter=0;
etotal=1;

for iter=1:epochs
%Inicializacion derivadas del error
dE_dc=zeros(MF,variables,datos);
dE_dsigma=zeros(MF,variables,datos);

```

```

%*****
    for t=1:datos
%=====
%*****Fordward Pass*****
%=====
%Calculo de la capa 1
        for i=1:MF %MFxvariablesxdata
            O1(i, :, t)=exp(-0.5*((input(:, t))'-xk_ast(i, 1:end-
1))./sigma(i, 1:end-1)).^2);
        end
    end
    %Calculo de la capa 2
    O2=prod(O1, 2); %MFx1xdata
    O2=reshape(O2, MF, datos); %MFxdata
    %Calculo de la capa 3
    for i=1:MF
        O3(i, :)=O2(i, :)./sum(O2); %MFxdata
    end
    %Calculo de la capa 4
    A_=[input'];
    for t=1:datos
        for i=1:MF
            A(i, :, t)=O3(i, t).*A_(t, :);
        end
    end
    %Vector de parámetros del consecuente
    temp2(t, :)=reshape(A(:, :, t)', 1, (variables)*MF); %vector de
parametros
    end
    %Cálculo de mínimos cuadrados
    X=temp2\output';

    %Sumatoria capa 5
    O5=temp2*X;
    %Error cuadrático
    e=(output'-O5).^2;
    %Error MSEde aprendizaje
    etotal(iter)=(1/datos)*sum(e); %error total por iteracion
    etotal(iter)
    % Salida de validación
    O5_val=anfis_feedfwd(input_val, xk_ast, sigma, X, variables, MF);
    %Error MSE de validación
    etotal_val(iter)=(1/size(dataset_val, 1))*sum((dataset_val(:, end)-
O5_val).^2);

%=====
%*****Retro-propagación*****
%=====
%Cálculo de las derivadas
    dE_dO5=-(2/datos)*(output'-O5); %dataX1
    X_=reshape(X, variables, MF);
    dO5_dO3=(A_*X_); %dataxMF
    for i=1:MF
        dO3_dO2(i, :)=(sum(O2)-O2(i, :))./(sum(O2)).^2; %MFxdata
    end
end

```

```

end
    for j=1:variables
        temp=O1;
        temp(:,j,:)=[];
        dO2_dO1(:,j,:)=prod(temp,2);    %MFxvariablesxdatos
    end
    %dO2_dO1=reshape(dO2_dO1,MF,datos);
    for t=1:datos
        for i=1:MF
            d_dc(i,:,t)=(((input(:,t))'-xk_ast(i,1:end-
1))./sigma(i,1:end-1).^2);
            d_dsigma(i,:,t)=((((input(:,t))'-xk_ast(i,1:end-
1)).^2)./sigma(i,1:end-1).^3);
        end
    end
    dO1_dc=d_dc.*O1;    %MFxvariablesxdata
    dO1_dsigma=d_dsigma.*O1;    %MFxvariablesxdata
    for j=1:variables
        for i=1:MF

dE_dc(i,j,:)=dE_dO5.*dO5_dO3(:,i).*dO3_dO2(i,:)'.*reshape(dO2_dO1(i,j,:),
,datos,1).*reshape(dO1_dc(i,j,:),datos,1);

dE_dsigma(i,j,:)=dE_dO5.*dO5_dO3(:,i).*dO3_dO2(i,:)'.*reshape(dO2_dO1(i,
j,:),datos,1).*reshape(dO1_dsigma(i,j,:),datos,1);
        end
    end
    dE_dc=sum(dE_dc,3);
    dE_dsigma=sum(dE_dsigma,3);
    %Actualización de parámetros
    xk_ast(:,1:variables)=xk_ast(:,1:variables)-(n1*dE_dc);
    sigma(:,1:variables)=sigma(:,1:variables)-(n2*dE_dsigma);

%=====
%***** Fin iteración*****
%=====
    end
    output2=X_;
end

```

8.3 Estimación parámetros Takagi-Sugeno

La arquitectura considerada para sistemas ANFIS tiene en cuenta consecuentes con funciones Takagi-Sugeno de la siguiente forma:

$$f = r_{n_i} * x_n + r_{n-1_i} * x_{n-1} + \dots + r_{1_i} * x_1 + r_{0_i}$$

Donde $x_n \dots x_1$ son las variables del conjunto de datos y $r_{n_i} \dots r_{0_i}$ son los parámetros Takagi-Sugeno que se desea calcular. Para esto se aplica la estimación por mínimos cuadrados (LSE) debido a que el modelo es una combinación lineal.

El problema de la estimación por mínimos cuadrados parte del siguiente enunciado:

$$|A * X - output|^2$$

Donde,

$$A = [\bar{w}_1 f_1 \quad \bar{w}_2 f_2 \quad \dots \quad \bar{w}_n f_n] \quad output = \begin{bmatrix} output_1 \\ output_2 \\ \vdots \\ output_n \end{bmatrix} \quad X = \begin{bmatrix} r_n \\ r_{n-1} \\ \vdots \\ r_1 \\ r_0 \end{bmatrix}$$

El vector output corresponde a las salidas deseadas, la matriz A corresponde a la salida de la capa 4 y el vector X corresponde a los parámetros que se desea calcular aplicando la definición de LSE:

$$X = (A^T A)^{-1} A^T B$$

En caso de que la matriz A sea singular, es decir su determinante sea 0 (cero) es posible resolver el problema por un método iterativo:

$$S_{i+1} = S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}}$$

$$X_{i+1} = X_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T X_i)$$

8.4 Código Algoritmo clustering substractivo

```

%*****
%           Subtractive clustering algorithm
%
%
%           Arguments:
%           * input: dataset input
%           * output: dataset output
%           * ra      : cluster ratio
%           * eps_over: superior limit cluster (0.5)
%           * eps_sub : inferior limit cluster (0.15)
%*****

```

```

function [xk_ast, pk_ast, sigma]=
full_subtractive_clustering(input,output,ra,eps_over,eps_sub)
x=[input' output'];
minX = min(x);
maxX = max(x);
x=normaliz(x','maxmin');
minX = min(x);
maxX = max(x);
datos=size(input,2);           %# muestras
k=1;
stop=0;
retest=0;
pi=zeros(datos,1);
alpha=4/(ra^2);           %parametro calculo inicial de potenciales para cada
punto
rb=1.25*ra;           %sugerido en multiples trabajos
beta=4/(rb^2);
%*****Compute first potential for each
point*****%
for i=1:datos
    dis=0;
    for j=1:datos
        dis=dis+exp(-alpha*norm(x(i,:)-x(j,:))^2);           %calculo sumatoria
    end
    pi(i)=dis;
end
%*****First cluster center and
potential*****%
[maxpi,indx]=max(pi);           %maximo potencial
[pk_ast(1,:),indx1]=max(pi);           %maximo potencial (primer cluster)
xk_ast(1,:)=x(indx1,:);           %punto con el máximo potencial

while(stop==0)

    %*****Potential revision*****%

    for i=1:datos
        pi(i,:)=pi(i,:)-(maxpi*exp(-beta*norm(x(i,:)-x(indx,:)).^2));
    end
    [maxpi,indx]=max(pi);           %cluster candidate
end

if(maxpi>(eps_over*pk_ast(1,:)))

    %*****kth cluster center and potential*****%
    k=k+1;
    [pk_ast(k,1),indx]=max(pi);
    xk_ast(k,:)=x(indx,:);
    stop=0;
    retest=0;
elseif(maxpi<(eps_sub*pk_ast(1,:)))
    stop=1;
else
    for i=1:size(pk_ast)

```

```

d=x(indx,:)-xk_ast(i,:);
abs_d=d*d';
qd_dist(i,:)=abs_d;
end
dmin=sqrt(min(qd_dist));

if(((dmin/ra)+(maxpi/pk_ast(1,:))>=1)
    k=k+1;
    [pk_ast(k,1),indx]=max(pi);
    xk_ast(k,:)=x(indx,:);
    stop=0;

else
    pi(indx)=0;
    [maxpi,indx]=max(pi);
    stop=0;
end
end
end
%tomado de algoritmo substracting clustering anfis toolbox matlab
sigma = (ra* (maxX - minX)) / sqrt(8.0); %compute membership function
width
end

```

8.5 Algoritmo clustering substractivo

A continuación, el algoritmo de clustering substractivo reportado por (Chiu, 1994)

1. Calculo de los potenciales para cada dato del conjunto

$$P_i = \sum_{j=1}^n e^{-\alpha \|x_i - x_j\|^2} ; \quad \alpha = \frac{4}{r_a^2}$$

Donde r_a corresponde al radio del cluster y es una constante positiva

2. Se selecciona el dato con mayor potencial como el primer cluster

$$P_1 = \max(P_i)$$

3. Se recalcula el potencial de los datos restantes con el fin de eliminar aquellos que estén dentro de la vecindad del primer cluster

$$P_i = P_i - P_k^* e^{-\beta \|x_i - x_k\|^2} ; \quad \beta = \frac{4}{r_b^2}$$

Donde r_b es un número positivo constante y se recomienda $r_b=1.5*r_a$. k es el número de cluster

4. Luego de revisar los potenciales de cada uno de los datos restantes se selecciona nuevamente el mayor potencial como un candidato a cluster

$$P_k = \max(P_i)$$

5. El criterio para aceptar el dato como cluster es el siguiente

If $P_k^* > \bar{\epsilon}P_1^*$ Se acepta como cluster el dato

If $P_k^* > \epsilon P_1^*$ No se acepta como cluster el dato y se finaliza el clustering

En otro caso

If $\frac{d_{min}}{r_a} + \frac{P_k^*}{P_1^*} \geq 1$ Se acepta como cluster el dato

Donde d_{min} es la mínima distancia entre el dato y los clusters ya aceptados

En otro caso se rechaza el dato como cluster, $P_k^* = 0$ y se prueba con el siguiente potencial más grande

6. Se repite desde el numeral 3 hasta que cumpla el criterio para terminar el algoritmo

Algoritmo 8.1 Algoritmo de clustering substractivo Tomado de (Chiu, 1994)

8.6 Código Máquinas de vectores de soporte (SMV)

```

%*****%
%           SVM entrenado por algoritmo Kernel-Adatron           %
%           Máquina de vectores de soporte para regresión       %
% Argumentos:                                                  %
% * norm_dataset      : Conjunto de datos entrenamiento        %
% * norm_dataset_val: Conjunto de datos de validación          %
% * iter              : Cantidad máxima de iteraciones         %
% * eps               : Desviación máxima permitida en la SVM  %
% * C                 : Término de penalización para datos     %
%                    de la SVM                                 %
% * type              : Tipo de kernel a utilizar              %
% * sigma             : Ancho del kernel gaussiano            %
% * p                 : Grado de polinomio en kernel polinómico %
%*****%

function [norm_dataset fx fx_val etotal e
etotal_val]=svm(norm_dataset,norm_dataset_val,iter,eps,C,type,sigma,p)
%Tamaño del conjunto de datos
datasetsize=size(norm_dataset);
output=norm_dataset(:,size(norm_dataset,2));
output_val=norm_dataset_val(:,end);
%Normalización de conjunto de datos

```

```

norm_dataset=normaliz(norm_dataset,'maxmin');
norm_dataset_val=normaliz(norm_dataset_val,'maxmin');
%Número de muestras
Q=datasize(1);
input=norm_dataset(:,1:size(norm_dataset,2)-1);
normal_val=norm_dataset_val(:,1:end-1);
datos=Q;
n=1;
etotal=1;
%Inicialización de parámetros
alpha=zeros(datos,1);
alpha_ast=zeros(datos,1);
eta=zeros(1,datos);
b=0;
i=0;
j=0;

%Función auxiliar para calcular kernel
k=kernel(input,input,type,sigma,p);
k_val=kernel(input,normal_val,type,sigma,2);
%Calcular n (eta) tasa de aprendizaje
if(strcmp(type,'Polynomic'))
    for j=1:datos
        eta(j)=(1./(norm(input(j,:))^2+1)^2);
    end
elseif(strcmp(type,'Gaussian'))
    for j=1:datos
        eta(j)=1/k;
    end
else
    for j=1:datos
        eta(j)=0.001;
    end
end
%Algoritmo Kernel Adatron
while(etotal(n)>0.00001)

%Calculo delta_alpha and delta_alpha_ast
delta_alpha=eta'.*(-((alpha-alpha_ast)'*k)'+output-eps-b);

%Actualización de alpha
for i=1:datos
if(alpha(i)+delta_alpha(i)<0)
    alpha(i)=0;
end
if((alpha(i)+delta_alpha(i)>=0)&&(alpha(i)+delta_alpha(i)<=C))
    alpha(i)=alpha(i)+delta_alpha(i);
end
if(alpha(i)+delta_alpha(i)>C)
    alpha(i)=C;
end
end

```



```
end

delta_alpha_ast=eta'.*((alpha-alpha_ast) '*k) '-output-eps+b);
%Actualización de alpha_ast
for i=1:datos
if(alpha_ast(i)+delta_alpha_ast(i)<0)
    alpha_ast(i)=0;
end
if((alpha_ast(i)+delta_alpha_ast(i)>=0)&&(alpha_ast(i)+delta_alpha_ast(i)
)<=C))
    alpha_ast(i)=alpha_ast(i)+delta_alpha_ast(i);
end
if(alpha_ast(i)+delta_alpha_ast(i)>C)
    alpha_ast(i)=C;
end
end
% Cálculo de término de Bias
b=b+sum(delta_alpha-delta_alpha_ast);
n=n+1;
%Salida de aprendizaje SVM
fx=((alpha-alpha_ast) '*k) '+b;
%Salida de validación de SVM
fx_val=((alpha-alpha_ast) '*k_val) '+b;
%Error MSE de validación
etotal_val(n)=(1/size(norm_dataset_val,1))*sum((output_val-fx_val).^2);
%Error MSE de aprendizaje
etotal(n)=(1/Q)*sum((output-fx).^2);
etotal(n)
if(n==iter)
    etotal(n)=0;
end
end
e=(1/Q)*((output-fx).^2);
```

9. Bibliografía

- Acuña, G., Curilem, M., & Cubillos, F. (2014). Desarrollo de un sensor virtual basado en modelo NARMAX y máquina de vectores de soporte para molienda semiautógena. *RIAI - Revista Iberoamericana de Automatica E Informatica Industrial*, 11(1), 109–116. <https://doi.org/10.1016/j.riai.2013.09.008>
- Alpaydın, E. (2010). *Introduction to Machine Learning Second Edition*. The MIT Press.
- Alzate Ibañez, A. M. (2010). Modelado y control de una columna de destilación binaria. *Universidad Nacional de Colombia, Manizales, Tesis de M*, 191.
- Babuška, R. (2009). FUZZY AND NEURAL CONTROL DISC Course Lecture Notes (November 2009), (November), 55–72. Retrieved from http://www.dcsc.tudelft.nl/~disc_fnc/transp/fncontrol.pdf
- Battiti, R. (1994). Using Mutual Information for Selecting Features in Supervised Neural Net Learning. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 5(4).
- Behnasr, M., & Jazayeri-Rad, H. (2015). Robust data-driven soft sensor based on iteratively weighted least squares support vector regression optimized by the cuckoo optimization algorithm. *Journal of Natural Gas Science and Engineering*, 22, 35–41. <https://doi.org/10.1016/j.jngse.2014.11.017>
- Bella, A. Di, Fortuna, L., & Graziani, S. (2007). A Comparative Analysis of the Influence of Methods for Outliers Detection on the Performance of Data Driven Models. *Imtc 2007*, 1–5. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4258091
- Ben-gal, I. (2005). Outlier Detection. *Data Mining and Knowledge Discovery Handbook*, 131–146. https://doi.org/10.1007/0-387-25465-x_7
- Bhartiya, S., & Whiteley, J. R. (2001). Development of inferential measurements using neural networks. *ISA Transactions*, 40(4), 307–323.
- Billings, S. A., & Voon, W. S. F. (1986). Correlation based model validity tests for non-linear models. *International Journal of Control*, 44(1), 235–244. <https://doi.org/10.1080/00207178608933593>

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning. Pattern Recognition* (Vol. 4). <https://doi.org/10.1117/1.2819119>
- Bolf, N., Ivandic, M., & Galinec, G. (2008). Soft sensors for crude distillation unit product properties estimation and control. In *2008 Mediterranean Conference on Control and Automation - Conference Proceedings, MED'08* (pp. 1804–1809).
- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152. <https://doi.org/10.1.1.21.3818>
- C. L. Mallows. (1973). Some Comments on Cp. *Technometrics*, 15(November), 87–94.
- Campbell, C., & Cristianini, N. (1999). Simple learning algorithms for training support vector machines. *Unpublished Manuscript*, 1–29. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.7938&rep=rep1&type=pdf>
- Casali, A., Gonzalez, G., Torres, F., Vallebuona, G., Castelli, L., & Gimenez, P. (1998). Particle size distribution soft-sensor for a grinding circuit. *Powder Technology*, 99(1), 15–21. [https://doi.org/10.1016/S0032-5910\(98\)00084-9](https://doi.org/10.1016/S0032-5910(98)00084-9)
- Chang, Y., Lv, Z., Wang, F., Mao, Z., & Wang, X. (2006). Soft sensing modeling based on stacked least square-support vector machine and its application. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 1, 4846–4850. <https://doi.org/10.1109/WCICA.2006.1713305>
- Chen, X., Gao, F., & Chen, G. (2004). A soft-sensor development for melt-flow-length measurement during injection mold filling. *Materials Science and Engineering A*, 384(1–2), 245–254. <https://doi.org/10.1016/j.msea.2004.06.039>
- Cheng, Z., & Liu, X. (2015). Optimal online soft sensor for product quality monitoring in propylene polymerization process. *Neurocomputing*, 149, 1216–1224. <https://doi.org/10.1016/j.neucom.2014.09.006>
- Chitralkha, S. B., & Shah, S. L. (2010a). Application of support vector regression for developing soft sensors for nonlinear processes. *Canadian Journal of Chemical Engineering*, 88(5), 696–709.
- Chitralkha, S. B., & Shah, S. L. (2010b). Support Vector Regression for soft sensor design of nonlinear processes. In *Control & Automation (MED), 2010 18th Mediterranean Conference on* (pp. 569–574).
- Chiu, S. L. (1994). A cluster extension method with extension to fuzzy model identification. *Proceedings of 1994 IEEE 3rd International Fuzzy Systems*

- Conference, 1240–1245. <https://doi.org/10.1109/FUZZY.1994.343644>
- De Silva, A. M., & Leong, P. H. W. (2015). Grammar-Based Feature Generation for Time-Series Prediction. In *SpringerBriefs in Computational Intelligence* (p. 105). <https://doi.org/10.1007/978-981-287-411-5>
- Fortuna, L., Graziani, S., & Xibilia, M. G. (2005). Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice*, 13(4), 499–508.
- Funsten, B. T. (2015). ECG Classification with an Adaptive Neuro-Fuzzy Inference System. *California Polytechnic State University*, (June). <https://doi.org/10.1017/CBO9781107415324.004>
- Gao Lin. (2011). Ester rate Soft-sensor in PET process. *IEEE*, 3.
- Garcia, C., de Carvalho Berni, C., & de Oliveira, C. E. N. (2008). Hardware/firmware implementation of a soft sensor using an improved version of a fuzzy identification algorithm. *ISA Transactions*, 47(2), 157–170. <https://doi.org/10.1016/j.isatra.2007.09.004>
- Gonzaga, J. C. B., Meleiro, L. a C., Kiang, C., & Maciel Filho, R. (2009). ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Computers and Chemical Engineering*, 33(1), 43–49. <https://doi.org/10.1016/j.compchemeng.2008.05.019>
- Graziani, S., Pagano, F., & Xibilia, M. G. (2010). Soft sensor for a Propylene Splitter with seasonal variations. *2010 IEEE International Instrumentation and Measurement Technology Conference, I2MTC 2010 - Proceedings*, 273–278. <https://doi.org/10.1109/IMTC.2010.5488032>
- Guyon, I., Elisseeff, A., & De, A. M. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, 1157–1182. <https://doi.org/10.1016/j.aca.2011.07.027>
- Hamdan, H. (2013). An Exploration of the Adaptive Neuro-Fuzzy Inference System (ANFIS) in Modelling Survival, (March), 33–35.
- Huang, J., Zhu, Z., & Tao, H. (2010). A Support Vector Machines for the Polymerization Rate of Vinyl Acetate. *Proceedings of the 8th World Congress on Intelligent Control and Automation*, 7046–7049.
- Iswandy, K., & König, A. (2011). Hybrid virtual sensor based on RBFN or SVR compared

- for an embedded application. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6882 LNAI, pp. 335–344). https://doi.org/10.1007/978-3-642-23863-5_34
- Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy And Soft Computing Jang: a computational approach to learning and machine intelligence*.
- Jang, J. S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665–685. <https://doi.org/10.1109/21.256541>
- Jassar, S., Liao, Z., & Zhao, L. (2009). Adaptive neuro-fuzzy based inferential sensor model for estimating the average air temperature in space heating systems. *Building and Environment*, 44(8), 1609–1616.
- Kadlec, P. (2009). *On robust and adaptive soft sensors*. Computing. Bournemouth: Bournemouth University. Retrieved from <http://eprints.bournemouth.ac.uk/13288/>
- Kadlec, P., Gabrys, B., & Strandt, S. (2009). Data-driven Soft Sensors in the process industry. *Computers and Chemical Engineering*.
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2), 273–324. [https://doi.org/10.1016/S0004-3702\(97\)00043-X](https://doi.org/10.1016/S0004-3702(97)00043-X)
- Kovalev, A. A., Kotlyar, V. V., Porfirev, A. P., & Kalinkina, D. S. (2016). Math-Net . Ru, 39(2), 172–180.
- Liu, C., & Yan, F. (2011). Radical basis function neural network-based NOx soft sensor technique. *2011 International Conference on Electrical and Control Engineering*, 1(2), 1152–1156. <https://doi.org/10.1109/ICECENG.2011.6057893>
- Liukkonen, M., Hälikkä, E., Hiltunen, T., & Hiltunen, Y. (2012). Dynamic soft sensors for NOx emissions in a circulating fluidized bed boiler. *Applied Energy*, 97(x), 483–490. <https://doi.org/10.1016/j.apenergy.2012.01.074>
- Ljung, L. (1987). *System Identification Theory for User*. PTR Prentice Hall Upper Saddle River NJ (Vol. 25). New Jersey: Prentice Hall. [https://doi.org/10.1016/0005-1098\(89\)90019-8](https://doi.org/10.1016/0005-1098(89)90019-8)
- Ludwig, O., Nunes, U., Araújo, R., Schnitman, L., & Lepikson, H. A. (2009). Applications of information theory, genetic algorithms, and neural models to predict oil flow. *Communications in Nonlinear Science and Numerical Simulation*, 14(7), 2870–2885. <https://doi.org/10.1016/j.cnsns.2008.12.011>
- Luigi Fortuna, Salvatore Graziani, and M. G. X. (2005). Virtual instruments in refineries. *IEEE Instrumentation and Measurement Magazine*, (October).

- Luyben, W. L. (1999). Modeling, simulation and control for chemical engineers, 741. <https://doi.org/10.1002/pol.1973.130110416>
- Ng, A. (2000). Support Vector Machines. In *Intelligent Systems and their Applications IEEE* (Vol. pt.1, pp. 1–25). Stanford. <https://doi.org/10.1016/j.aca.2011.07.027>
- Park, S., & Han, C. (2000). Anonline soft sensor based on multivariate smoothing procedure for quality estimation in distillation columns. *Computers and Chemical Engineering*, 24, 871–877.
- Pearson, R. K. (2001). Exploring process data. *Journal of Process Control*, 11(2), 179–194. [https://doi.org/10.1016/S0959-1524\(00\)00046-9](https://doi.org/10.1016/S0959-1524(00)00046-9)
- Pearson, R. K. (2002). Outliers in process modeling and identification. *IEEE Transactions on Control Systems Technology*, 10(1), 55–63. <https://doi.org/10.1109/87.974338>
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(8), 1226–1238. <https://doi.org/10.1109/TPAMI.2005.159>
- Perry John H. (1997). *Chemical Engineers' Handbook. Second edition* (Perry, John H., ed.). *Journal of Chemical Education* (Vol. 19). <https://doi.org/10.1021/ed019p449.2>
- Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods*, 376. <https://doi.org/10.1109/ISKE.2008.4731075>
- Roffel, B., & Betlem, B. (2007). *Process Dynamics and Control: Modeling for Control and Prediction*.
- Shi, J., Liu, X., & Sun, Y. (2006). Melt index prediction by neural networks based on independent component analysis and multi-scale analysis. *Neurocomputing*, 70(1–3), 280–287. <https://doi.org/10.1016/j.neucom.2006.01.029>
- Skogestad, S. (2010). *Chemical and energy process engineering*. CRC Press. <https://doi.org/10.1103/PhysRevC.88.014310>
- Sliškovic, D., Grbic, R., & Hocenski, Ž. (2011a). Methods for Plant Data-Based Process Modeling in Soft-Sensor Development. *Automatika*, 52(4), 306–318. <https://doi.org/10.7305:116406>
- Sliškovic, D., Grbic, R., & Hocenski, Ž. (2011b). Methods for Plant Data-Based Process Modeling in Soft-Sensor Development. *AUTOMATIKA*, 52, 306–318.

- Smola, A., Burges, C., Drucker, H., Golowich, S., Hemmen, L. Van, Muller, K., ... Vapnik, V. (1996). Regression Estimation with Support Vector Learning Machines in collaboration with. *NeuroCOLT Technical Report*.
- Smola, A. J., Sch, B., & Schölkopf, B. (2004). A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(3), 199–222.
<https://doi.org/10.1023/B:STCO.0000035301.49549.88>
- Smola, A. J., & Schölkopf, B. (2004). A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(3), 199–222. <https://doi.org/10.1023/B:Stco.0000035301.49549.88>
- Souza, F., Santos, P., & Araújo, R. (2010). Variable and delay selection using neural networks and mutual information for data-driven soft sensors. *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*, (I), 1–8. <https://doi.org/10.1109/ETFA.2010.5641329>
- Souza, F. A. A. de. (2014). *Computational Intelligence Methodologies for soft-sensors development in Industrial Processes*. Coimbra, Brazil: Universidade de Coimbra.
- Sun, K., Liu, J., Kang, J.-L., Jang, S.-S., Wong, D. S.-H., & Chen, D.-S. (2014). Development of a variable selection method for soft sensor using artificial neural network and nonnegative garrote. *Journal of Process Control*, 24(7), 1068–1075. <https://doi.org/10.1016/j.jprocont.2014.05.010>
- Torkkola, K. (2003). Feature Extraction by Non Parametric Mutual Information Maximization. *J. Mach. Learn. Res.*, 3, 1415–1438.
<https://doi.org/10.1162/153244303322753742>
- Vapnik, V., & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24(6), 774–780. <https://doi.org/citeulike-article-id:619639>
- Veropoulos, K. (2001). Machine Learning Approaches to Medical Decision Making, (May), 292.
- Wang, P. W. P. (2010). Design and application of soft sensor object in water treatment plant. *Networking, Sensing and Control (ICNSC), 2010 International Conference on*. <https://doi.org/10.1109/ICNSC.2010.5461526>
- Warne, K., Prasad, G., Rezvani, S., & Maguire, L. (2004). Statistical and computational intelligence techniques for inferential model development: A comparative evaluation and a novel proposition for fusion. *Engineering Applications of Artificial Intelligence*, 17(8), 871–885. <https://doi.org/10.1016/j.engappai.2004.08.020>

Xu, S., & Liu, X. (2014). Melt index prediction by fuzzy functions with dynamic fuzzy neural networks. *Neurocomputing*, *142*, 291–298.

<https://doi.org/10.1016/j.neucom.2014.03.056>

Yafen, L., Qi, L., Huijuan, W., & Ningsheng, M. (2006). Soft Sensing Based on LS-SVM and Its Application to a Distillation Column. *Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on*, *1*, 177–182.

<https://doi.org/10.1109/ISDA.2006.246>