

INTELIGENCIA ARTIFICIAL PARA GESTIÓN Y ATENCIÓN DE ALARMAS

AUTOR:

Samuel Arturo Padilla Trillos

PRESENTADO A:

Rene Leandro Varón. Ing.

Carlos Alberto Parra. Ing.



**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C. 2022**

Contenido

1.	Introducción.....	3
1.1	Resumen	3
1.2	Planteamiento del problema, posibles usuarios y justificación	3
2.	Fundamentos teóricos	4
2.1	Estado del arte	4
2.2	Marco teórico.....	5
2.2.1	Sistema de alarmas:	5
2.2.2	Series de tiempo.....	6
2.2.3	Machine learning	6
2.2.4	Modelos de aprendizaje supervisado	6
3.	Descripción de la solución.....	9
3.1	Objetivos.....	9
3.1.1	Objetivo general	9
3.1.2	Objetivos específicos.....	9
3.2	Descripción general de la solución.....	9
3.3	Diagrama de bloques	9
3.4	Especificaciones del sistema	10
3.5	Base de datos de análisis histórico de alarmas.	10
3.6	Procesamiento de datos	11
3.6.1	Filtrado de alarmas	11
3.6.2	Codificación.....	11
3.6.3	Generador de dataset de alarmas frecuentes y “chattering alarms” por hora	11
3.6.4	Generador de dataset original de alarmas con etiquetas.....	12
3.7	Predictor de alarmas	13
3.7.1	Entrenamiento del modelo.....	13
3.7.2	Evaluación del modelo	15
3.7.3	Persistencia de modelo	15
3.8	Interfaz gráfica.....	15
4.	Protocolo de Pruebas	16
5.	Resultados.....	16
5.1	Modelo de predicción de inundaciones	17
5.2	Modelo de predicción de alarmas frecuentes.....	17
5.3	Modelo de predicción de “chattering alarms”	18
6.	Conclusiones.....	19
6.1	Random forest vs SVM	19
6.2	Relación entre ventana de alarmas y porcentaje de predicción	19

6.3	Predicción del flujo de alarmas de la siguiente hora	19
6.4	Predicción de alarmas frecuentes y “chattering alarms”	19
6.5	Recomendaciones	19
7.	Bibliografía.....	19

1. Introducción

1.1 Resumen

El siguiente documento es el resultado del proyecto de investigación “INTELIGENCIA ARTIFICIAL PARA GESTIÓN Y ATENCIÓN DE ALARMAS” como opción de grado de Ingeniería Electrónica, en este documento se presenta una breve explicación de la problemática que está viviendo hoy en día la industria del petróleo con los sistemas de alarmas digitales y la cantidad exagerada de alarmas que lo componen.

Este documento plantea una solución basada en inteligencia artificial para la problemática que está viviendo hoy en día la industria del petróleo con los sistemas de alarmas digitales, en la solución se intenta predecir ciertos eventos de alarmas que distraen la atención del operario de eventos más relevantes. Para el planteamiento de la solución se utilizaron como base 3 documentos relacionados con la predicción de eventos de alarmas con base en modelos de inteligencia artificial, donde cada documento afronta la problemática implementando diferentes modelos de inteligencia artificial. En la solución planteada en este proyecto se utilizan 2 modelos de inteligencia artificial para predicción de alarmas frecuentes, “chattering alarms” y eventos de inundación, estos son “Random forest” y “SVM”. En el documento se exponen y sustentan los resultados de los dos métodos de inteligencia artificial, en los cuales se obtuvieron buenos resultados, finalmente se presentan las conclusiones basadas en los objetivos formulados. A continuación, se expone y describe el planteamiento del problema, posibles usuarios y la justificación.

1.2 Planteamiento del problema, posibles usuarios y justificación

Hoy en día vivimos en un mundo donde los sistemas digitales cada vez están más presente, y la industria del petróleo no es la excepción. En la actualidad los sistemas de alarmas son sistemas digitales, esto permite agregar alarmas al sistema de manera rápida y a muy bajos costos. Sin embargo, se pueden generar consecuencias graves para los operarios de planta, ya que, un sistema de alarmas puede llegar a tener muchas alarmas configuradas, a esta cantidad de alarmas el operario debe prestar atención y en ocasiones por el número superar su capacidad de respuesta a estas. Lo anterior es conocido en la industria como inundación de alarmas, evento que ha sido originador de problemas y accidentes en la industria.

Tory-Tech diseña, desarrolla, integra e implementa soluciones de software corporativo para las industrias de petróleo y gas y otras industrias en todo el mundo. Una de las soluciones que Tory-Tech ofrece a la industria es un software para gestión de alarmas.

Los sistemas de alarmas de las plantas de proceso tienen la función de avisar al operador que una situación anormal se está presentando en algún proceso. Estas alarmas son definidas mediante una filosofía de alarmas la cual es una guía para la implementación y la administración de estas, también provee las bases para seleccionar, priorizar y configurar una nueva alarma; esta filosofía de alarmas también define el número total de alarmas que un operador puede atender en cierta cantidad de tiempo de manera efectiva para operar bajo condiciones seguras.[1]

La atención de estas alarmas es probablemente uno de los mayores y más persistentes desafíos en las plantas de procesos, ya que, si la definición y configuración de alarmas no se efectúa de manera consistente y estructurada, el sistema de alarmas termina generando alertas todo el tiempo, ya sean alarmas obsoletas, alarmas de parloteo, alarmas de muy alta frecuencia, ocasionando eventos de “inundaciones de alarmas”, entre otras. Dichas alarmas son conocidas como “Malos Actores”, y básicamente corresponden a alarmas que desvían la atención de una alarma más importante llegando a limitar la capacidad de los operadores de planta para atender situaciones críticas.[2]

Esta ineficiente priorización y atención de alarmas en plantas petroleras puede desencadenar pérdidas millonarias para las empresas y accidentes graves. Muchas de las investigaciones realizadas por la “US Chemical Safety Board (CSB)” citan a las inundaciones de alarmas como uno de los principales causantes de incidentes en la industria, también “The British-based organisation Engineering Equipment & Materials Users’ Association (EEMUA)” llegó a la misma conclusión en su informe de 1999[3], ejemplo de esto fue la explosión ocurrida el 19 de agosto de 2004 en la instalación de esterilización de óxido de etileno de la compañía Sterigenics International Inc. Este incidente provocó heridas a cuatro trabajadores y causó un daño extenso en la planta de 66.000 pies cuadrados.[4]

Para la solución de esta problemática se propone implementar una aplicación basada en inteligencia artificial con el fin de facilitar las tareas de los operadores en los procesos de identificación y clasificación de los posibles malos actores de forma efectiva, mediante el uso de análisis de datos avanzados y teniendo en cuenta los patrones históricos para identificar qué tipos de alarmas requieren evaluar sus parámetros de configuración.

2. Fundamentos teóricos

2.1 Estado del arte

Para la consulta del estado del arte se consultaron las bases de datos que brinda la Pontificia Universidad Javeriana, se encontraron documentos que trataban sobre la implementación de sistemas de inteligencia artificial, pero finalmente se seleccionaron 3 documentos relacionados directamente con el tema de sistemas de inteligencia artificial aplicados a la gestión de alarmas.

A. *Intelligent Based Alarm Management System for Plant Automation*

En este documento se presenta la propuesta y los resultados de la implementación de un sistema de gestión de alarmas inteligente para evitar las inundaciones de alarmas, este usa aprendizaje automático para suprimir los malos actores, en esta implementación se utilizan los datos históricos de alarmas con los cuales el sistema aprende de la información pasada y de esta manera el operario no tiene que pasar por todas las alarmas que se generan. El operador puede actuar rápidamente y responder a una alarma importante[5].

Los métodos de aprendizaje automático requieren datos para el entrenamiento. Por lo tanto, la primera fase es la captura de datos. En esta fase, se recopilan datos que consisten en información de alarma, en la fase de preprocesamiento, se eliminan los malos actores y en la fase de entrenamiento, el sistema se entrena mediante los algoritmos de aprendizaje automático utilizando los datos recopilados y procesados[5].

A continuación, se muestra los resultados de los diferentes métodos de aprendizaje autónomo implementados:

Algoritmo	Clasificaciones correctas	Clasificaciones incorrectas	Exactitud
SVM	7176	6	99.9165%
DT(J48)	7168	14	99.8051%
Random Tree	7051	131	98.176%
One R	7172	10	99.8608%
PART	7169	13	99.819%

Tabla 1: Resultados de los algoritmos inteligentes.[5]

B. *A Machine Learning Approach to Predict Chattering Alarms*

En este documento se presenta los resultados de la implementación de un Sistema de gestión de alarmas inteligente en una planta de producción de amoniaco, se entrenó y evaluó una red neuronal profunda (DNN) para identificar el parloteo de las alarmas. Específicamente, el propósito del algoritmo es clasificar las alarmas en dos categorías: "Parloteo dentro de una hora" o "No parloteo dentro de una hora" [6].

Para el entrenamiento de la red neuronal profunda se creó una base de datos con las características de las alarmas y sus respectivas etiquetas, Una etiqueta puede ser "1" si la alarma muestra vibración dentro de una hora o "0" si la alarma no muestra vibración dentro de una hora [6]. A continuación, se muestran los resultados:

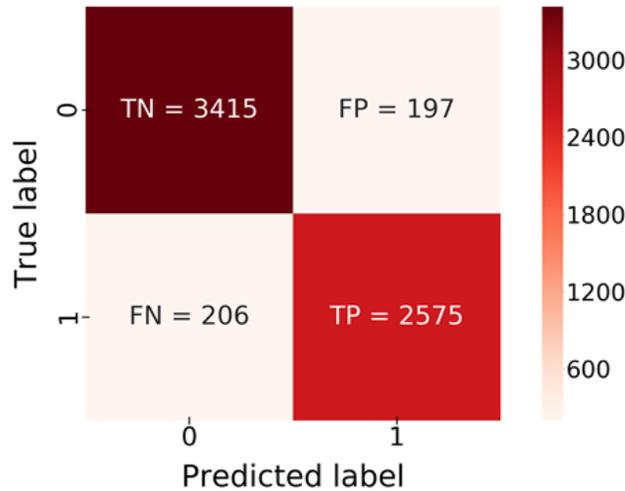


Imagen 1: Matriz de confusión de los resultados de la red neuronal. [6]

C. Alarm Ranking Model for Intelligent Management of Metro Systems Based on Statistical Machine Learning Methods

En este documento se presenta la propuesta y los resultados de la implementación de un sistema de gestión de alarmas inteligente que clasifica las alarmas entrantes entre 0 y 100 dependiendo su prioridad, para fines de predicción y clasificación de alarmas; se propone un proceso de fusión de datos especializado, que incluye la construcción detallada de las siguientes características: Densidad de alarmas, Entropía de estaciones, Entropía de sistemas funcionales, Retraso de reconocimiento, Factores a corto plazo (instantánea). A partir del cálculo de las características, se aplica un modelo de regresión logística múltiple para clasificar la alarma entrante, a continuación, se muestran los resultados obtenidos:[7]

Training Accuracy	Testing Accuracy	Best Coefficients
82.37%	81.29%	$\alpha_1=0.322$
		$\alpha_2=0.297$
		$\alpha_3=0.223$
		$\alpha_4=0.116$
		$\alpha_5=0.041$

Tabla 2: Resultados regresión logística múltiple[7].

2.2 Marco teórico

2.2.1 Sistema de alarmas:

Los sistemas de alarmas de las plantas de proceso tienen la función de avisar al operador que una situación anormal se está presentando en algún proceso. Dichos sistemas están compuestos por los siguientes elementos:

- Filosofía de Alarmas – Conjunto de documentos y procedimientos que definen prioridades, impactos, tiempos de atención, etc. para cada tipo de alarma a ser configurada dentro del sistema. La filosofía de alarmas también define el número total de alarmas que un operador puede atender en cierta cantidad de tiempo de manera efectiva para operar bajo condiciones seguras.[1]

- Base de Datos Maestra de Alarmas – Corresponde a la base de datos donde se tienen configuradas las alarmas para cada señal que interviene en el proceso de la planta. Para cada señal se configuran las respectivas alarmas, (Alto-Alto, Alto, Bajo, Bajo-Bajo, Rata de Cambio, etc.), incluyendo otra información relevante como prioridad, tiempos de reacción, etc. Esta base de datos se configura con base en los lineamientos definidos en la Filosofía de Alarmas.
- Base de Datos de Alarmas – Corresponde a la base de datos de las alarmas generadas de acuerdo con las condiciones del proceso y con base en lo configurado en la Base de Datos Maestra de Alarmas.

2.2.2 Series de tiempo

Las series de tiempo son sucesiones de datos medidos en un determinado momento y ordenados en forma cronológica. Los datos pueden estar espaciados a intervalos iguales o desiguales.[8]

2.2.2.1 Predicción de series de tiempo

Las predicciones de series de tiempo tienen por objetivo estimar los valores de la serie al futuro, donde aún no hay mediciones, para llevar a cabo este proceso se definen 2 variables: cantidad de períodos y horizonte de predicción. Cantidad de periodos es el número de periodos que se van a usar como base para la predicción y horizonte de predicción es la cantidad de periodos que se van a predecir.[8]

2.2.3 Machine learning

El machine learning es una rama de la IA que le permite a un sistema aprender a partir de datos en vez de programación explícita, conforme el algoritmo recibe más datos de entrenamiento, es posible generar un modelo preciso basado en datos.[9]

Las técnicas de machine learning son necesarias para generar un modelo predictivo. Dependiendo de la naturaleza del problema que se está atendiendo, existen diferentes enfoques basados en el tipo y volumen de los datos. En los siguientes apartados, se presentan las categorías del machine learning.[9]

2.2.3.1 Aprendizaje supervisado

El aprendizaje supervisado comienza con un conjunto etiquetados de datos y una cierta comprensión de cómo se clasifican estos datos. El aprendizaje supervisado tiene la intención de encontrar patrones en datos que se pueden aplicar a un proceso de analítica. Estos datos tienen características etiquetadas que definen el significado de los datos. Por ejemplo, se puede crear una aplicación de machine learning con base en imágenes y descripciones escritas que distinga entre millones de animales.[9]

2.2.3.2 Aprendizaje no supervisado

El aprendizaje no supervisado se utiliza cuando el problema requiere una cantidad masiva de datos sin etiquetar. Por ejemplo, las aplicaciones de redes sociales, tales como Twitter, Instagram y Snapchat, tienen grandes cantidades de datos sin etiquetar. La comprensión del significado detrás de estos datos requiere algoritmos que clasifican los datos con base en los patrones o clústeres que encuentra. El aprendizaje no supervisado lleva a cabo un proceso iterativo, analizando los datos sin intervención humana. Se utiliza con la tecnología de detección de spam en e-mails.[9]

2.2.4 Modelos de aprendizaje supervisado

El aprendizaje supervisado es una rama de Machine Learning, un método de análisis de datos que utiliza algoritmos que aprenden iterativamente de los datos para permitir que a partir del procesamiento intensivo se encuentre información escondida. A continuación, presentaremos algunos modelos utilizados para el aprendizaje supervisado.[9]

2.2.4.1 SVM (Support Vector Machine)

Las máquinas de soporte vectorial (SVM) son algoritmos de aprendizaje autónomo supervisado que se puede usar para desafíos de clasificación o regresión, pero es usado mayormente en problemas de clasificación. En el algoritmo SVM, los datos son representados como puntos en un espacio N-dimensional, donde N es la cantidad de características o campos del dataset y el valor de cada característica es el valor de una coordenada particular[10]

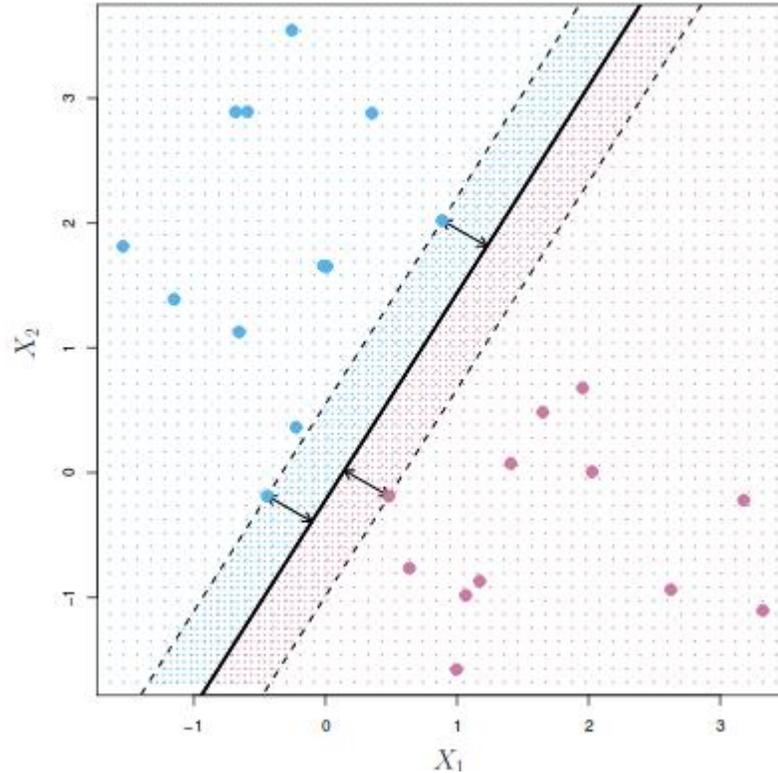


Imagen 2: Datos en el hiperplano[10]

Los vectores de soporte son simplemente las coordenadas de la observación individual. El clasificador SVM es una frontera que segrega mejor las dos clases (hiperplano/línea).[10]

2.2.4.2 Random forest

Random forest es un algoritmo de aprendizaje automático supervisado aplicado en problemas de clasificación y regresión, sin embargo, ofrece mejores resultados en problemas de clasificación, este algoritmo se basa en el funcionamiento de los árboles de decisión. Un árbol de decisión es un mapa de las diferentes posibilidades de clasificación del problema relacionadas a partir de sus probabilidades de ocurrencia.[11]

El funcionamiento del algoritmo Random forest consta de asociar varios modelos de árboles de decisión para realizar predicciones o clasificaciones de mayor precisión, hay dos maneras de asociar estos modelos, estas dos maneras son “Bagging” y “Boosting”. [11]

- Bagging: crea un subconjunto de entrenamiento diferente a partir de datos de entrenamiento de muestra con reemplazo y el resultado final se basa en la votación por mayoría. Por ejemplo, Bosque aleatorio.[11]
- Boosting: combina aprendices débiles en aprendices fuertes mediante la creación de modelos secuenciales de modo que el modelo final tenga la mayor precisión.[11]

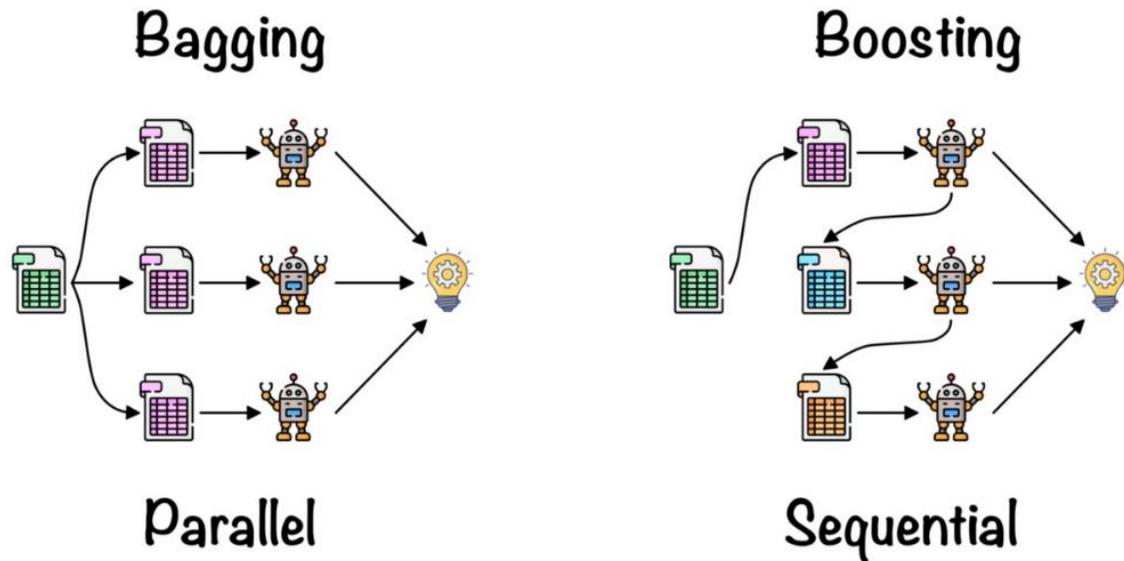


Imagen 3: Diagrama de agrupamiento de modelos de árboles de decisión[11]

Finalmente, luego de que todos los modelos de árboles de decisión, se realiza una votación, y el resultado más recurrente es la predicción o la clasificación que se escoge como resultado final.[11]

2.2.4.3 Redes neuronales

Las redes neuronales es un algoritmo de aprendizaje automático supervisado cuyo funcionamiento está inspirado en redes neuronales biológicas, Funciona interconectadas un número elevado de unidades de procesamiento o “neuronas” que parecen versiones abstractas de neuronas. Las unidades de procesamiento o “neuronas” se organizan en capas, hay tres partes normalmente en una red neuronal: una capa de entrada, con unidades que representan los campos de entrada; una o varias capas ocultas; y una capa de salida, con una unidad o unidades que representa el campo o los campos a predecir o clasificar.[12]

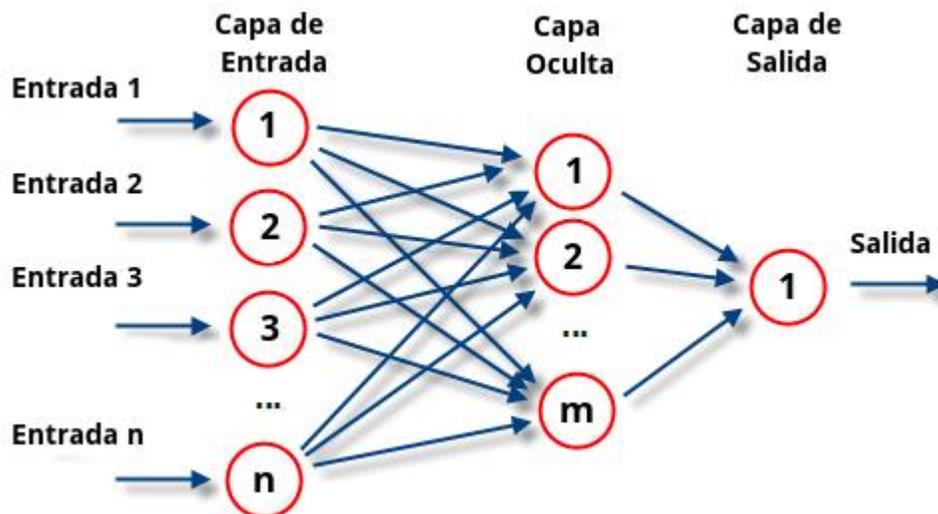


Imagen 4: Diagrama de una red neuronal[12]

Al principio todos los valores de Las unidades de procesamiento o pesos de las “neuronas” son aleatorios, por lo tanto, las respuestas que resultan de la red son, posiblemente, erróneas. La red neuronal pasa por un proceso de entrenamiento en el cual va cambiando los valores de Las unidades de procesamiento o pesos de las “neuronas” a partir de la diferencia entre el resultado de la red neuronal y la etiqueta. A medida que

progresa el entrenamiento, la red se va haciendo cada vez más precisa en la replicación de resultados conocidos. Una vez entrenada, la red se puede aplicar a casos futuros en los que se desconoce el resultado.[12]

3. Descripción de la solución

3.1 Objetivos

3.1.1 Objetivo general

- Diseñar un sistema para la predecir de posibles alarmas frecuentes, “chattering alarms” y eventos de inundación de alarmas dentro de un universo de alarmas utilizando técnicas de inteligencia artificial para asistir a operadores de centros de control.

3.1.2 Objetivos específicos

- Leer los datos de la base de datos donde se encuentran los datos históricos de las alarmas.
- Procesar los datos obtenidos de la base de datos donde se encuentran los datos históricos de las alarmas.
- Predecir las posibles alarmas frecuentes, “chattering alarms” y eventos de inundación de alarmas en las alarmas entrantes.
- Evaluar el sistema de predicción con ayuda de los datos de evaluación.

3.2 Descripción general de la solución

Para la solución de la problemática se implementará un algoritmo de machine learning para la predicción de posibles “Chattering alarms”, alarmas frecuentes e inundaciones de alarmas. Este sistema se comunicará con la Base de Datos de Alarmas en la cual se encuentran los datos históricos, para esto el sistema utilizará el lenguaje SQL para descargar los datos de entrenamiento del sistema, luego de que el sistema sea entrenado, predecirá posibles “Chattering alarms”, alarmas frecuentes e inundaciones de alarmas que se puedan presentar en la siguiente hora.

3.3 Diagrama de bloques

A continuación, se presenta el diagrama de bloques del sistema:

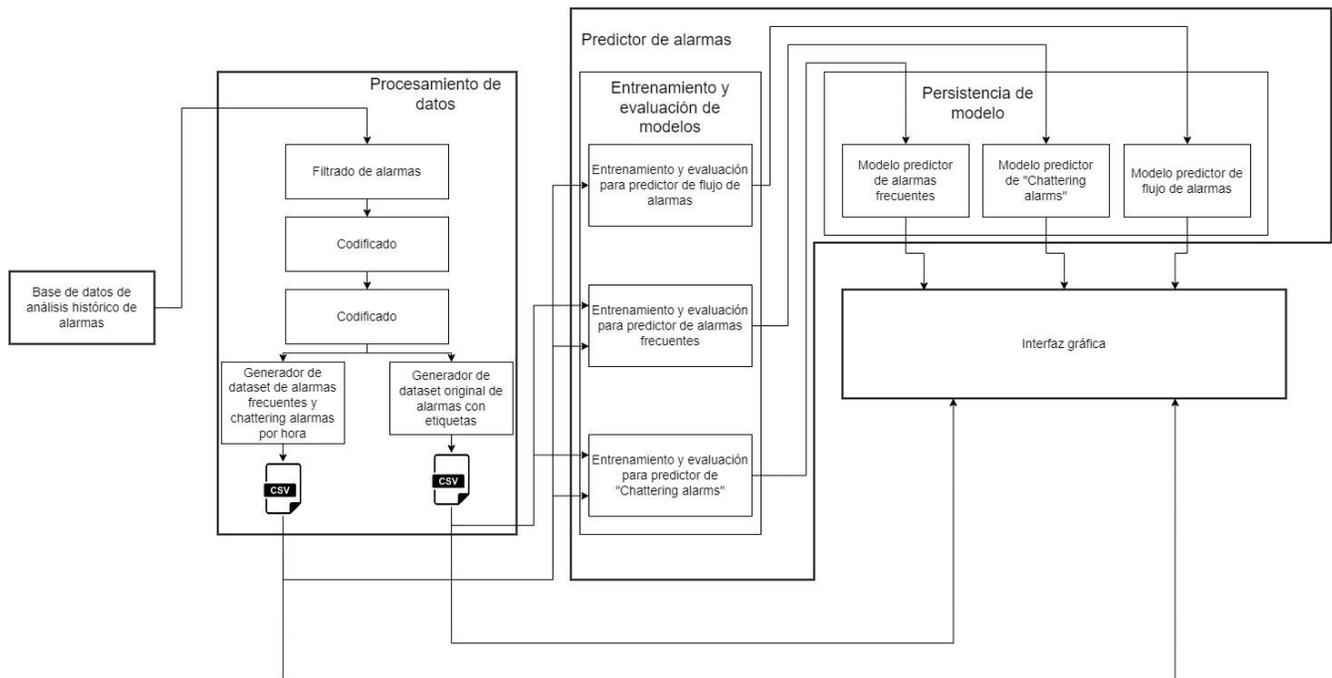


Imagen 5: Diagrama de bloques del sistema

3.4 Especificaciones del sistema

Los bloques del sistema deben tener las siguientes especificaciones:

- Base de datos de análisis histórico de alarmas: En este bloque se encuentran los datos históricos recolectados de las alarmas con las cuales se entrena el modelo estadístico.
- Procesamiento de datos: Este bloque recibe los datos del bloque “Base de datos de análisis histórico de alarmas” para procesarlos y estandarizarlos de tal manera que estén listos para ser usados.
- Predicador de alarmas: Este bloque recibe los datos procesados del bloque “Procesamiento de datos”, para entrenar y evaluar el algoritmo de predicción.
- Interfaz gráfica: Este bloque se encarga de presentar los resultados obtenidos del predicador de alarmas de una manera amigable para el operario.

A continuación, se presentan especificaciones generales del sistema:

- Comunicación con las bases de datos: La comunicación entre el bloque “Base de datos de análisis histórico de alarmas” y el bloque “Procesamiento de datos” se hará mediante el lenguaje SQL.
- Se busca que el sistema sea agnóstico, es decir, que solo se base en las alarmas entrantes para tomar una decisión, sin tomar en cuenta topología del punto.

3.5 Base de datos de análisis histórico de alarmas.

Como se mencionó anteriormente, aquí se encuentra el histórico de alarmas de un universo de alarmas, a continuación, se presenta una tabla con los datos más relevantes encontrados en la base de datos de análisis histórico de alarmas:

Nombre de dato	Tipo de dato	Descripción
timeStrMsec	String	Tiempo en el que se disparó la alarma
type	String	Tipo de notificación
process	String	Proceso en el que se disparó la alarma
point	String	Punto donde se disparo la alarma
description	String	Descripción de la alarma
stationName	String	Estación donde se disparó la alarma
message	String	Mensaje asociado a la alarma
sevInt	Entero	Severidad de la alarma

Tabla 3: Datos de la Base de datos de análisis histórico de alarmas

3.6 Procesamiento de datos

En el bloque de preprocesamiento encontramos 3 procesos los cuales son: Codificado, identificación de alarmas y agrupación de datos. Con estos procesos se busca procesar y estandarizar los datos, de tal manera que los datos de entrada tengan una alta correlación con los datos a predecir. A continuación, se explicarán los procesos realizados en el bloque de preprocesamiento.

3.6.1 Filtrado de alarmas

Entre los datos obtenidos de la base de datos de análisis histórico de alarmas hay un campo llamado “type”, este campo identifica el tipo de notificación de la base de datos, en la base de datos se encuentran notificaciones de tipo “alarm”, “control” y “event”, Como la finalidad del sistema es predecir posibles “Chattering alarms”, alarmas frecuentes e inundaciones de alarmas solo necesitamos las notificaciones de tipo “alarm”, por lo que en el proceso de filtrado de alarmas se separan las notificaciones de tipo “alarm”, y el resto de notificaciones no son tenidas en cuenta.

3.6.2 Codificación

Los modelos estadísticos, son modelos matemáticos que ayudan a clasificar o a predecir un evento, estos al ser modelos matemáticos, los datos de entrada y de salida que reciben estos modelos deben ser numéricos, por lo tanto, los datos que no son de tipo numérico deben tener un equivalente numérico el cual va a servir como dato de entrada y de salida de los modelos estadísticos. En el proceso de codificación se establece un equivalente numérico a los datos de tipo “String” para que estos puedan usarse como datos de entrada y de salida para el modelo estadístico.

3.6.3 Generador de dataset de alarmas frecuentes y “chattering alarms” por hora

Este bloque se encarga de generar un nuevo dataset a partir de los datos codificados, el nuevo dataset contiene la hora a la que pertenecen los datos (“Time”), los segundos promedio que transcurren entre cada alarma en esa hora (“Dif”), la información sobre el flujo de alarmas en esa hora donde 0 es Flujo Bajo, 1 es Flujo Moderado y 2 es Flujo Alto (“Flood”), las tres alarmas más frecuentes (“FA_PointN”, FA_DesN y FASevIntN) y las tres alarmas con más alta recurrencia en “Chattering alarm” (“CA_PointN”, CA_DesN y CASevIntN), De esta manera la cantidad de filas de datos se reduce a 24 por cada día de alarmas, Este proceso ayuda a tener una cantidad fija de datos en una ventana de tiempo, ya que antes del procesamiento de datos el tiempo transcurrido entre cada fila de datos no era homogéneo, por esta razón la cantidad de datos que habría en una ventana de tiempo no sería igual a lo largo del dataset original

timeStrMsec	type	process	point	description	message	sevInt
12/8/2021 6:59:20.805	control	BLTHost.RealTime	WOODHVNMTTR_trgt	Target Volume	Modified countdown.WOODHVNMTTR_tr	0
12/8/2021 6:59:20.805	control	BLTHost.RealTime	WOODHVNMTTR_trgt	WOODHAVEN TARGET ALARM	Issued command START 88 by WPL01CTL	0
12/8/2021 6:59:20.789	control	BLTHost.RealTime	WOODHVNMTTR_trgt	WOODHAVEN TARGET ALARM	Issued command STOP 88 by WPL01CTL	0
12/8/2021 6:59:20.789	control	BLTHost.RealTime	WOODHVNMTTR_trgt	Target Volume	Modified countdown.WOODHVNMTTR_tr	0
12/8/2021 6:59:14.680	alarm	OMNICOMM XMODBUS_Omnicommm 8604	FC_STOCS	STOCKBRIDGE_FC_(SP)	Communications on Conn_FC_STOCS stati	4
12/8/2021 6:59:14.680	alarm	OMNICOMM XMODBUS_Omnicommm 8604	FC_STOCS	STOCKBRIDGE_FC_(SP)	Communications on Conn_FC_STOCS rest	4
12/8/2021 6:59:12.055	alarm	OMNICOMM XMODBUS_Omnicommm 8604	FC_STOCS	STOCKBRIDGE_FC_(SP)	Communications on Conn_FC_STOCS stati	12
12/8/2021 6:59:12.055	event	OMNICOMM XMODBUS_Omnicommm 8604	Conn_FC_STOCS	STOCKBRIDGE_FC_(SP)	Network connection to 10.10.67.32 port	0
12/8/2021 6:58:59.415	control	BLTHost.RealTime	STCKBRG2AC203FMP_trgt	STOCKBRIDGE SPARTAN TARGET ALARM	Issued command START 83 by WPL01CTL	0
12/8/2021 6:58:59.399	control	BLTHost.RealTime	STCKBRGPAC203FMP_trgt	Target Volume	Modified countdown.STCKBRGPAC203FM	0
12/8/2021 6:58:59.383	control	BLTHost.RealTime	STCKBRGPAC203FMP_trgt	Target Volume	Modified countdown.STCKBRGPAC203FM	0
12/8/2021 6:58:59.383	control	BLTHost.RealTime	STCKBRG2AC203FMP_trgt	STOCKBRIDGE SPARTAN TARGET ALARM	Issued command STOP 83 by WPL01CTL	0
12/8/2021 6:58:39.867	control	AlarmSerialization_7300	WOODHVNMTTR_trgt	WOODHAVEN TARGET ALARM	Issued command ACKNOWLEDGE countd	0
12/8/2021 6:58:24.258	alarm	Countdown_12412	WOODHVNMTTR_trgt	WOODHAVEN TARGET ALARM	Countdown on batchXfer.WOODHVNMTI	16
12/8/2021 6:57:52.648	control	AlarmSerialization_7300	STCKBRG2AC203FMP_trgt	STOCKBRIDGE SPARTAN TARGET ALARM	Issued command ACKNOWLEDGE countd	0
12/8/2021 6:57:51.992	alarm	OMNICOMM XMODBUS_Omnicommm 8604	LOCKPRTMSTCB16BIT	LOCKPORT STORAGE (16 BIT)	REASONABILITY from NORMAL - Flowrat	20
12/8/2021 6:57:51.133	alarm	OMNICOMM XMODBUS_Omnicommm 8604	LOCKPRTMSTCB16BIT	LOCKPORT STORAGE (16 BIT)	NORMAL from REASONABILITY - Reasona	4
12/8/2021 6:57:45.992	alarm	OMNICOMM XMODBUS_Omnicommm 8604	LOCKPRTMSTCB16BIT	LOCKPORT STORAGE (16 BIT)	REASONABILITY from NORMAL - Flowrat	20
12/8/2021 6:57:45.154	alarm	OMNICOMM XMODBUS_Omnicommm 8604	LOCKPRTMSTCB16BIT	LOCKPORT STORAGE (16 BIT)	NORMAL from REASONABILITY - Reasona	4

Imagen 6: Dataset original

Time	Dif	Flood	FA_Point1	FA_Des1	FASevInt1	FA_Point2	FA_Des2	FASevInt2	FA_Point3	FA_Des3	FASevInt3	CA_Point1	CA_Des1	CASevInt1	CA_Point2	CA_Des2	CASevInt2	CA_Point3	CA_Des3	CASevInt3
2021-12-11 07:00:00	6	1	301	1010	2	159	188	2	514	429	6	312	1017	0	183	930	0	613	697	3
2021-12-11 06:00:00	8	1	301	1010	2	159	188	2	165	195	4	220	1017	0	301	1010	2	159	188	2
2021-12-11 05:00:00	17	0	301	1010	2	159	188	2	165	195	4	301	1010	2	159	188	2	165	195	4
2021-12-11 04:00:00	13	0	301	1010	2	159	188	2	68	80	6	220	1017	0	159	188	2	301	1010	2
2021-12-11 03:00:00	8	1	301	1010	2	159	188	2	165	195	4	301	1010	2	1239	1067	6	159	188	2
2021-12-11 02:00:00	18	0	301	1010	2	186	206	2	116	167	2	301	1010	2	1056	951	6	186	206	2
2021-12-11 01:00:00	22	0	301	1010	2	159	188	4	165	195	4	301	1010	2	159	188	4	486	373	2
2021-12-11 00:00:00	16	0	301	1010	2	159	188	2	207	117	2	301	1010	2	828	646	4	490	377	6
2021-12-10 23:00:00	17	0	301	1010	2	743	557	6	280	1017	0	743	557	6	280	1017	0	301	1010	2
2021-12-10 22:00:00	16	0	301	1010	2	228	1017	0	717	550	2	228	1017	0	301	1010	2	729	745	3
2021-12-10 21:00:00	20	0	301	1010	2	828	646	4	733	595	3	828	646	4	301	1010	2	732	745	3
2021-12-10 20:00:00	26	0	301	1010	2	1164	118	2	544	745	3	301	1010	2	1164	118	2	544	745	3
2021-12-10 19:00:00	14	0	301	1010	2	828	646	4	159	188	2	828	646	4	280	1017	0	243	1017	0
2021-12-10 18:00:00	17	0	301	1010	2	486	373	2	545	444	3	301	1010	2	729	745	3	826	606	3
2021-12-10 17:00:00	15	0	301	1010	2	791	648	3	312	1017	0	791	648	3	312	1017	0	301	1010	2
2021-12-10 16:00:00	7	1	301	1010	2	1056	951	6	828	646	4	828	646	4	183	930	0	243	1017	0
2021-12-10 15:00:00	7	1	301	1010	2	828	646	4	1056	951	6	828	646	4	215	1017	0	190	1017	0
2021-12-10 14:00:00	9	1	301	1010	2	1056	951	6	986	853	6	264	1017	0	196	930	0	190	1017	0
2021-12-10 13:00:00	7	1	301	1010	2	1056	951	6	828	646	4	828	646	4	1056	951	6	258	1017	0
2021-12-10 12:00:00	16	0	301	1010	2	215	1017	0	129	179	3	215	1017	0	301	1010	2	119	168	2
2021-12-10 11:00:00	19	0	301	1010	2	312	1017	0	159	188	2	312	1017	0	301	1010	2	590	665	6
2021-12-10 10:00:00	15	0	301	1010	2	293	1017	0	289	907	2	293	1017	0	1248	1074	6	301	1010	2
2021-12-10 09:00:00	16	0	301	1010	2	137	169	2	119	168	2	215	1017	0	301	1010	2	241	752	2
2021-12-10 08:00:00	14	0	301	1010	2	293	1017	0	1168	1052	6	293	1017	0	301	1010	2	211	752	2
2021-12-10 07:00:00	16	0	301	1010	2	137	169	2	119	168	2	215	1017	0	301	1010	2	1174	1048	3
2021-12-10 06:00:00	17	0	301	1010	2	0	21	2	931	847	2	301	1010	2	0	21	2	966	880	2
2021-12-10 05:00:00	18	0	301	1010	2	228	1017	0	280	1017	0	228	1017	0	280	1017	0	301	1010	2
2021-12-10 04:00:00	14	0	301	1010	2	293	1017	0	986	853	6	293	1017	0	301	1010	2	412	745	3
2021-12-10 03:00:00	17	0	301	1010	2	137	169	2	119	168	2	215	1017	0	293	1017	0	301	1010	2
2021-12-10 02:00:00	9	1	301	1010	2	772	585	3	618	691	6	743	557	6	280	1017	0	301	1010	2
2021-12-10 01:00:00	8	1	301	1010	2	828	646	4	1239	1067	6	828	646	4	301	1010	2	937	789	3
2021-12-10 00:00:00	13	0	301	1010	2	1170	1055	4	486	373	2	1170	1055	4	312	1017	0	672	509	4

Imagen 7: Dataset de alarmas frecuentes y “chattering alarms” por hora

3.6.4 Generador de dataset original de alarmas con etiquetas

En este bloque se agregan tres columnas, la columna “FA” la cual es la etiqueta de la alarma para alarma frecuente, esta columna es binaria, cuando esta se encuentra en 1, significa que la alarma es considerada una alarma frecuente, por lo contrario, si se encuentra en 0, significa que esta no es considerada una alarma frecuente. Otra columna agregada es “CA”, la cual es la etiqueta de la alarma para “chattering alarm”, esta columna también es binaria, cuando esta se encuentra en 1, significa que la alarma es considerada “chattering alarm”, por lo contrario, si se encuentra en 0, significa que esta no es considerada “chattering alarm”. La tercera columna agregada se llama “dif”, la cual es una columna con números enteros en su contenido, este valor es la cantidad de segundos que tardó en generarse la alarma a partir de la alarma anterior. Finalmente se filtran las columnas necesarias para la clasificación, estas son: “timeStrMsec”, “dif”, “point”, “description”, “sevInt”, “FA” y “CA”

timeStrMsec	type	process	point	description	message	sevInt
12/8/2021 6:59:20.805	control	BLTHost.RealTime	WOODHVNMTTr_trgt	Target Volume	Modified countdown.WOODHVNMTTr_trgt	0
12/8/2021 6:59:20.805	control	BLTHost.RealTime	WOODHVNMTTr_trgt	WOODHAVEN TARGET ALARM	Issued command START 88 by WPL01CTLV	0
12/8/2021 6:59:20.789	control	BLTHost.RealTime	WOODHVNMTTr_trgt	WOODHAVEN TARGET ALARM	Issued command STOP 88 by WPL01CTLV	0
12/8/2021 6:59:20.789	control	BLTHost.RealTime	WOODHVNMTTr_trgt	Target Volume	Modified countdown.WOODHVNMTTr_trgt	0
12/8/2021 6:59:14.680	alarm	OMNICO MM XM ODBUS_Omnicom m 8604	FC_STOCS	STOCKBRIDGE_FC_(SP)	Communications on Conn_FC_STOCS stati	4
12/8/2021 6:59:14.680	alarm	OMNICO MM XM ODBUS_Omnicom m 8604	FC_STOCS	STOCKBRIDGE_FC_(SP)	Communications on Conn_FC_STOCS rest	4
12/8/2021 6:59:12.055	alarm	OMNICO MM XM ODBUS_Omnicom m 8604	FC_STOCS	STOCKBRIDGE_FC_(SP)	Communications on Conn_FC_STOCS stati	12
12/8/2021 6:59:12.055	event	OMNICO MM XM ODBUS_Omnicom m 8604	Conn_FC_STOCS	STOCKBRIDGE_FC_(SP)	Network connection to 10.10.67.32 port	0
12/8/2021 6:58:59.415	control	BLTHost.RealTime	STCKBRG2AC203FMP_trgt	STOCKBRIDGE SPARTAN TARGET ALARM	Issued command START 83 by WPL01CTLV	0
12/8/2021 6:58:59.399	control	BLTHost.RealTime	STCKBRGPAC203FMP_trgt	Target Volume	Modified countdown.STCKBRGPAC203FM	0
12/8/2021 6:58:59.383	control	BLTHost.RealTime	STCKBRGPAC203FMP_trgt	Target Volume	Modified countdown.STCKBRGPAC203FM	0
12/8/2021 6:58:59.383	control	BLTHost.RealTime	STCKBRG2AC203FMP_trgt	STOCKBRIDGE SPARTAN TARGET ALARM	Issued command STOP 83 by WPL01CTLV	0
12/8/2021 6:58:39.867	control	AlarmSerialization_7300	WOODHVNMTTr_trgt	WOODHAVEN TARGET ALARM	Issued command ACKNOWLEDGE countd	0
12/8/2021 6:58:24.258	alarm	Countdown_12412	WOODHVNMTTr_trgt	WOODHAVEN TARGET ALARM	Countdown on batchXfer.WOODHVNMTTr	16
12/8/2021 6:57:52.648	control	AlarmSerialization_7300	STCKBRG2AC203FMP_trgt	STOCKBRIDGE SPARTAN TARGET ALARM	Issued command ACKNOWLEDGE countd	0
12/8/2021 6:57:51.992	alarm	OMNICO MM XM ODBUS_Omnicom m 8604	LOCKPRTMSTCB16BIT	LOCKPORT STORAGE (16 BIT)	REASONABILITY from NORMAL - Flowrat	20
12/8/2021 6:57:51.133	alarm	OMNICO MM XM ODBUS_Omnicom m 8604	LOCKPRTMSTCB16BIT	LOCKPORT STORAGE (16 BIT)	NORMAL from REASONABILITY - Reasona	4
12/8/2021 6:57:45.992	alarm	OMNICO MM XM ODBUS_Omnicom m 8604	LOCKPRTMSTCB16BIT	LOCKPORT STORAGE (16 BIT)	REASONABILITY from NORMAL - Flowrat	20
12/8/2021 6:57:45.154	alarm	OMNICO MM XM ODBUS_Omnicom m 8604	LOCKPRTMSTCB16BIT	LOCKPORT STORAGE (16 BIT)	NORMAL from REASONABILITY - Reasona	4

Imagen 8: Dataset original

timeStrMsec	dif	point	description	sevInt	FA	CH
2021-12-11 06:59:31	10	113	161	3	1	0
2021-12-11 06:59:20	9	107	156	0	1	0
2021-12-11 06:59:11	16	5	26	0	1	0
2021-12-11 06:58:55	5	5	26	3	1	0
2021-12-11 06:58:49	0	301	1010	0	1	0
2021-12-11 06:58:49	2	301	1010	0	1	0
2021-12-11 06:58:46	3	301	1010	2	1	0
2021-12-11 06:58:43	0	8	34	3	1	0
2021-12-11 06:58:43	22	1183	27	0	1	0
2021-12-11 06:58:20	14	107	156	6	1	0
2021-12-11 06:58:05	0	159	188	0	1	0
2021-12-11 06:58:05	2	159	188	0	1	0
2021-12-11 06:58:03	0	165	195	4	1	0
2021-12-11 06:58:03	5	159	188	2	1	0
2021-12-11 06:57:57	0	1183	27	3	1	0
2021-12-11 06:57:57	21	7	200	3	1	0
2021-12-11 06:57:35	5	113	161	2	1	0
2021-12-11 06:57:30	9	611	695	4	1	0
2021-12-11 06:57:20	3	113	161	3	1	0
2021-12-11 06:57:17	0	301	1010	0	1	0
2021-12-11 06:57:17	2	301	1010	0	1	0
2021-12-11 06:57:14	89	301	1010	2	1	0
2021-12-11 06:55:45	0	301	1010	0	1	0

Imagen 9: Dataset con etiquetas

3.7 Predictor de alarmas

El sistema tiene 3 modelos predictores, un predictor de inundaciones de alarmas, un predictor de alarmas frecuentes y un predictor de “chattering alarms”, estos modelos pasan 3 procesos importantes para luego ser utilizados en la interfaz gráfica, estos procesos son: entrenamiento, evaluación y persistencia de modelo.

3.7.1 Entrenamiento del modelo

Los tres modelos predictores se entrenan mediante aprendizaje supervisado. El modelo predictor de inundaciones intenta predecir el flujo de alarmas para la siguiente hora, entre mayor sea el flujo de alarmas para la siguiente hora, mayor será la probabilidad de inundaciones de alarmas severas. Para este modelo se utiliza un bloque de datos del datasets de alarmas frecuentes y “chattering alarms” por hora como entrada, para predecir el valor de la columna “Flood” de la siguiente hora. La cantidad de horas que se toman en cuenta para predecir el valor de la columna “Flood” de la siguiente hora se escogen según los resultados del protocolo de pruebas. A continuación, se presenta la imagen 10 donde se explica visualmente este proceso.



Imagen 10: Diagrama en bloques de modelo predictor de inundaciones

El modelo predictor de “chattering alarms” predecirá si una alarma es considerada o no “chattering alarms” en la hora donde se generó la alarma. Para este modelo se utilizan los dos dataset generados, el dataset de alarmas frecuentes y “chattering alarms” por hora y el dataset original de alarmas con etiquetas. Los datos que se utilizan como entrada son un bloque de datos del datasets de alarmas frecuentes y “chattering alarms” por hora y la alarma entrante la cual se quiere predecir si es considerada “chattering alarms” o no, el bloque de datos del datasets de alarmas frecuentes y “chattering alarms” por hora que se utiliza como parte de la entrada debe ser de N horas anteriores desde la hora donde se generó la alarma que se quiere predecir. La cantidad de horas que se toman en cuenta para predecir si la alarma es considerada “chattering alarms” o no se escoge según los resultados del protocolo de pruebas. A continuación, en la imagen 11 se explica visualmente este proceso.

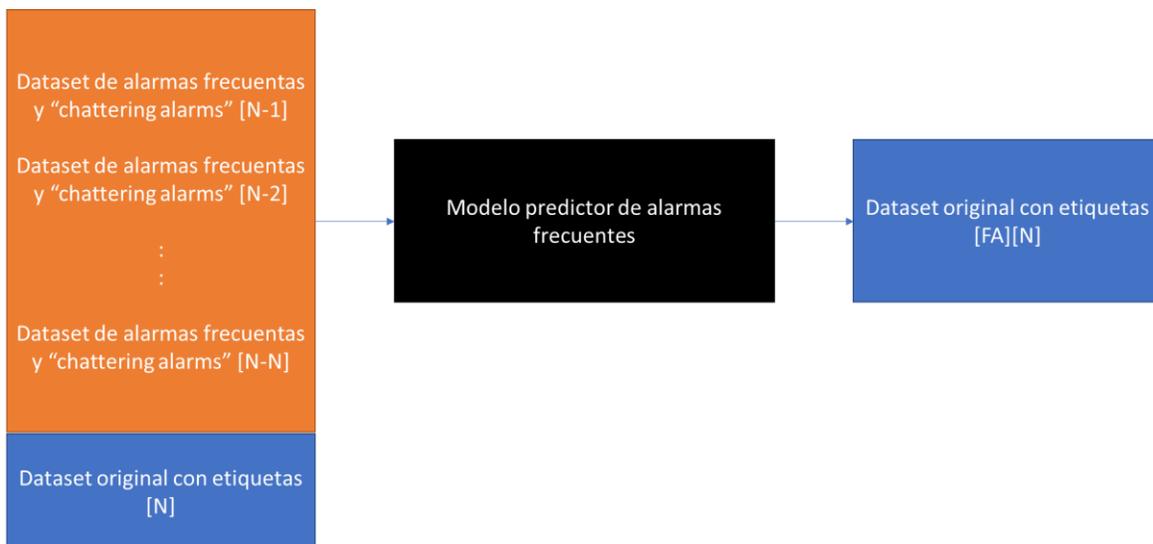


Imagen 11: Diagrama en bloques de modelo predictor de alarmas frecuentes

El modelo predictor de alarmas frecuentes predice si una alarma es considerada o no alarma frecuente en la hora donde se generó la alarma. Para este modelo se utilizan los dos dataset generados, el dataset de alarmas frecuentes y “chattering alarms” por hora y el dataset original de alarmas con etiquetas. Los datos que se utilizan como entrada son un bloque de datos del datasets de alarmas frecuentes y “chattering alarms” por hora y la alarma entrante, la cual se quiere predecir si es considerada alarma frecuente o no, el bloque de datos del datasets de alarmas frecuentes y “chattering alarms” por hora que se utiliza como parte de la entrada debe ser de N horas anteriores desde la hora donde se generó la alarma que se quiere predecir. La cantidad de horas que se toman en cuenta para

predecir si la alarma es considerada una alarma frecuente o no se escoge según los resultados del protocolo de pruebas. A continuación, se presenta en la imagen 12, donde se explica visualmente este proceso.

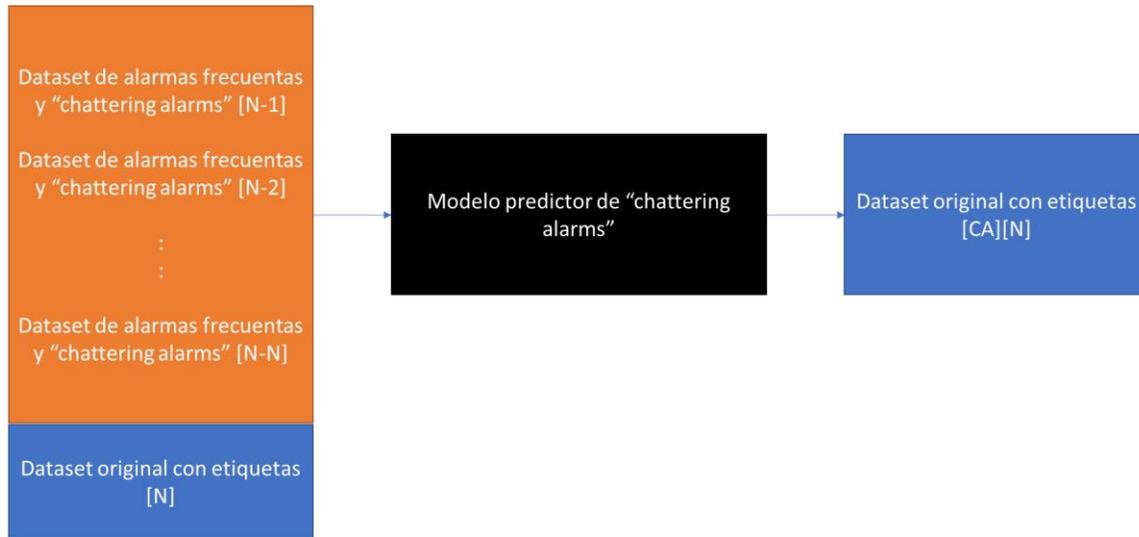


Imagen 12: Diagrama en bloques de modelo predictor de "chattering alarms"

3.7.2 Evaluación del modelo

En este proceso se busca verificar que los modelos entrenados fueron entrenados correctamente. Primeramente, se deben utilizar los modelos entrenados para predecir las variables de cada modelo con datos diferentes a los que se utilizaron para entrenar los modelos, de esta manera se tiene el vector de las variables predichas y el vector de las variables reales. Luego se obtienen 2 parámetros para cada modelo, los cuales entregan un acercamiento a que tan bien entrenados están los modelos, estos parámetros son, el porcentaje de acierto del vector de variables predichas sobre el vector de variables reales y el valor de correlación entre el vector de variables predichas y el vector de variables reales. Se considera que un modelo está bien entrenado cuando tiene un porcentaje de acierto mayor al 90% y una correlación mayor a 0.75.

3.7.3 Persistencia de modelo

El proceso de entrenamiento de evaluación de un modelo de machine learning muchas en ocasiones es rápido, pero en muchas otras ocasiones puede tardar días o semanas, por lo que muchas veces luego de que un modelo es entrenado y cumple con los criterios de evaluación para considerarlo como un modelo bien entrenado, los modelo se exportan ya entrenados y evaluados para no volver a repartir entre proceso al momento que volver a usar el modelo de machine learning, si no que se importa el modelo que se exportó anteriormente.

En este proceso se hace exactamente lo mismo que se explicó anteriormente, luego de que los modelos entrenados obtengan un porcentaje de acierto mayor al 90% y una correlación mayor a 0.75 son exportados para luego ser usados en la interfaz gráfica.

3.8 Interfaz gráfica

Con la interfaz gráfica se busca que el usuario pueda observar de manera amigable y organizada los resultados de los tres modelos de predicción entrenados, evaluados y exportados anteriormente. Para la implementación de la interfaz gráfica se utilizó el framework de Python "Flask", este framework está escrito en Python y se utiliza para crear aplicaciones web rápidamente y con un mínimo número de líneas de código, A continuación, se presenta una imagen de la interfaz gráfica terminada.

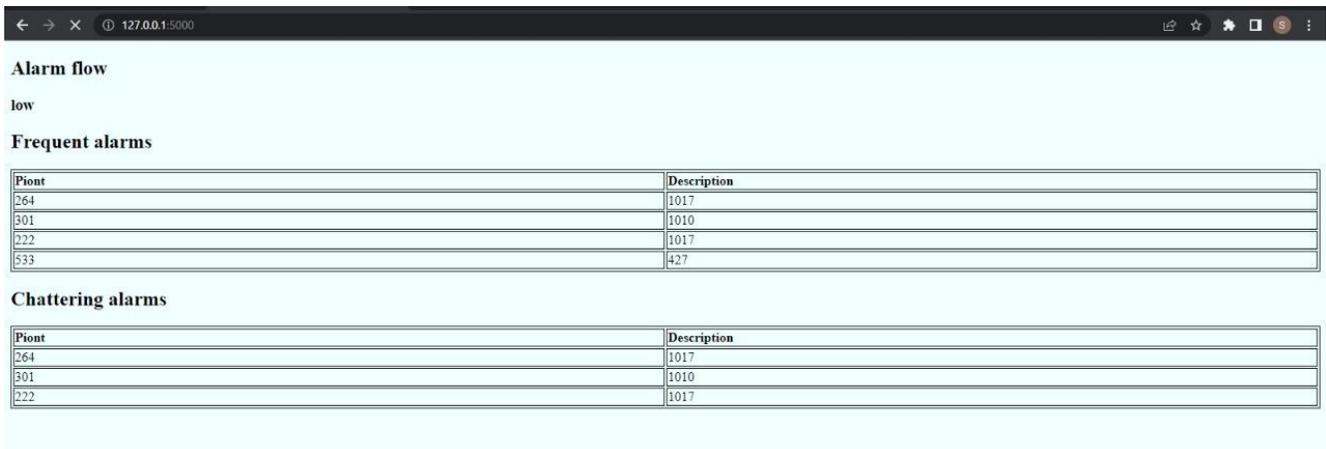


Imagen 13: Interfaz gráfica.

La interfaz grafica se actualiza cada minuto, en este proceso se vuelve a hacer la predicción de las variables a partir de la hora de la actualización, de esta manera cuando se muestre de nuevo la interfaz gráfica, se actualizarán las tablas de alarmas frecuentes, “chattering alarms” y el flujo de alarmas.

4. Protocolo de Pruebas

El predictor de alarmas es un modelo estadístico el cual tiene unos parámetros que especifican las características del modelo, en el protocolo de pruebas se cambiarán 3 parámetros, estos son, la cantidad de horas que ingresan al modelo estadístico para predecir las variables de la siguiente hora de alarmas, el modelo estadístico que se utiliza para predecir las variables de la siguiente hora de alarmas y los parámetros del modelo estadístico.



Imagen 14: Diagrama en bloques de protocolo de pruebas.

En el protocolo de pruebas se implementan 2 modelos estadísticos de aprendizaje supervisado, SVM, Random forest y redes neuronales, a estos modelos se le realizaron pruebas cambiando la cantidad de periodos de alarmas en los datos de entrada y cambiando los parámetros del modelo. En el caso de los periodos de alarmas en los datos de entrada, se realizaron las pruebas con 8, 16 y 24 periodos de alarmas en la entrada de los modelos estadísticos de aprendizaje supervisado. Los tipos de parámetros de los modelos estadísticos de aprendizaje supervisado cambian según el modelo, en el caso de SVM se cambiará el tipo de “Kernel” el cual es la forma del hiperplano y en Random Forest se cambiarán a cantidad de modelos de árboles de decisión asociados.

Los resultados serán guardados en matrices de datos para luego escoger los mejores resultados para cada campo del conjunto de datos.

5. Resultados

En este segmento se presentan los mejores resultados obtenido a partir del protocolo de pruebas para cada modelo de predicción, se presenta el modelo utilizado con sus respectivos parámetros, la cantidad de horas de alarmas que se utilizaron como entrada del modelo, el porcentaje de acierto del modelo y la correlación del modelo.

5.1 Modelo de predicción de inundaciones

Luego de realizar las diferentes pruebas de este modelo se obtuvieron muy buenos resultados en la predicción, en el porcentaje de acierto y en la correlación del modelo. A continuación, se en la tabla 4 se muestra la información de los modelos de “Random Forest” y “SVM” con los mejores resultados:

Mejor modelo (RF) Flood	
Ventana de alarmas	16horas
Numero de arboles de decision	1500
Probabilidad de acierto Chattering alarms	98.5454
Mejor modelo (SVM) Flood	
Ventana de alarmas	16horas
Grado del polinomio	3
Probabilidad de acierto Chattering alarms	96.5

Tabla 4: Resultados modelo de predicción de inundaciones

Se observa que hay una relación entre las horas en la ventana de alarmas y el porcentaje de predicción del flujo de alarmas en la siguiente hora. Aunque ambos modelos tienen un alto porcentaje de predicción, el modelo “Random forest” presenta un mejor porcentaje de predicción que el modelo “SVM”.

El porcentaje de acierto de modelo “Random forest” es casi perfecto, sin embargo, es importante ver la matriz de correlación, ya que el coeficiente de correlación que relaciona las variables predichas y las reales entrega un acercamiento numérico de que tanto se parecen la tendencia de la variable predicha con la tendencia de la variable real

Mtx_Corr	Flood_Raeal	Flood_RF
Flood_Raeal	1	0.95
Flood_RF	0.95	1

Tabla 5: matriz de correlación del modelo de predicción de inundaciones

A partir de la matriz de correlación podemos inferir que la tendencia de la variable predicha y la tendencia de la variable real se parecen en un 95%.

5.2 Modelo de predicción de alarmas frecuentes

Luego de realizar las diferentes pruebas de este modelo se obtuvieron buenos resultados en la predicción, en el porcentaje de acierto y en la correlación del modelo de “Random forest”. A continuación, se muestra una tabla con la información de los modelos de “Random Forest” y “SVM” con los mejores resultados:

Mejor modelo (RF) alarmas frecuentes	
Ventana de alarmas	16horas
Numero de arboles de decision	1500
Probabilidad de acierto alarmas frecuentes	92.2424
Mejor modelo (SVM) alarmas frecuentes	
Ventana de alarmas	16horas
Grado del polinomio	2
Probabilidad de acierto alarmas frecuentes	65.76

Tabla 6: Resultados modelo de predicción de alarmas frecuentes

Se puede observar que hay una relación entre las horas en la ventana de alarmas y el porcentaje de predicción del flujo de alarmas en la siguiente hora. El porcentaje de acierto de modelo “Random forest” es mejor que el porcentaje de acierto de modelo “SVM”, es importante ver la matriz de correlación, ya que el coeficiente de correlación que relaciona las variables predichas y las reales entrega acercamiento numérico de que tanto se parecen la tendencia de la variable predicha con la tendencia de la variable real

Mtx_Corr	Freq_Alarm_Raeal	Freq_Alarm_RF
Freq_Alarm_Raeal	1	0.88
Freq_Alarm_RF	0.88	1

Tabla 7: matriz de correlación del modelo de predicción de alarmas frecuentes

A partir de la matriz de correlación se puede inferir que la tendencia de la variable predicha y la tendencia de la variable real se parecen en un 88%.

5.3 Modelo de predicción de “chattering alarms”

Luego de realizar las diferentes pruebas de este modelo se obtuvieron muy buenos resultados en la predicción, en el porcentaje de acierto y en la correlación del modelo. A continuación, se muestra una tabla con la información de los modelos de “Random Forest” y “SVM” con los mejores resultados:

Mejor modelo (RF) Chattering alarms	
Ventana de alarmas	16horas
Numero de arboles de decision	1500
Probabilidad de acierto Chattering alarms	92.5454
Mejor modelo (SVM) Chattering alarms	
Ventana de alarmas	16horas
Grado del polinomio	3
Probabilidad de acierto Chattering alarms	90.4762

Tabla 8: Resultados modelo de predicción de “chattering alarms”

Se puede observar que hay una relación entre las horas en la ventana de alarmas y el porcentaje de predicción del flujo de alarmas en la siguiente hora. Aunque ambos modelos tienen un alto porcentaje de predicción, el modelo “Random forest” presenta un mejor porcentaje de predicción que el modelo “SVM”.

Es importante ver la matriz de correlación, ya que el coeficiente de correlación que relaciona las variables predichas y las reales entrega un acercamiento numérico de que tanto se parecen la tendencia de la variable predicha con la tendencia de la variable real

Mtx_Corr	Chatt_Alarm_Raeal	Chatt_Alarm_RF
Chatt_Alarm_Raeal	1	0.86
Chatt_Alarm_RF	0.86	1

Tabla 9: matriz de correlación del modelo de “chattering alarms”

A partir de la matriz de correlación se puede inferir que la tendencia de la variable predicha y la tendencia de la variable real se parecen en un 86%.

6. Conclusiones

6.1 Random forest vs SVM

Se observa en los resultados del protocolo de pruebas el modelo “Random forest” presenta mejores resultados que el modelo “SVM” en los tres modelos predictores. Esto se debe a que el modelo “Random forest” puede manejar una gran cantidad de variables de entrada e identificar las más significativas, característica que ayuda bastante a la predicción de las variables, debido a que la cantidad de datos que se utiliza como entrada al modelo es mayor a 300 datos. Otra característica importante del modelo “Random forest” es que utilizar múltiples árboles de decisión se reduce considerablemente el riesgo de overfitting.

6.2 Relación entre ventana de alarmas y porcentaje de predicción

Se observa que, en todos los modelos de predicción, la ventana de alarmas utilizada es de 16 horas, esta cantidad de horas tiene relación con el tiempo de operación de los oleoductos, los oleoductos tienen un tiempo de operación de 8 horas, por lo que se concluye que los modelos de predicción funcionan mejor con una ventana de alarmas de dos ciclos de operación de los oleoductos

6.3 Predicción del flujo de alarmas de la siguiente hora

Al momento de predecir al flujo de alarmas en la siguiente hora, se busca darle una idea al operario de qué tan pesado va a estar la siguiente hora en cuanto a trabajo de atención de alarmas. En un principio se pensó predecir la cantidad de alarmas que se generarían en la siguiente hora, pero lo importante en este caso no es saber la cantidad exacta de alarmas que se generarán, sino tener una idea del flujo de las alarmas para preparar al operario y de ser necesario repartir la atención de alarmas entre más operarios.

6.4 Predicción de alarmas frecuentes y “chattering alarms”

En el proceso de predicción de alarmas frecuentes y “chattering alarms”, estas son predichas al momento de que se genera la alarma en esa hora, de esta manera el operario tendrá una idea de las alarmas frecuentes y “chattering alarms” luego de alrededor de 10 minutos de alarmas de esa hora.

6.5 Recomendaciones

Al momento de implementar una interfaz gráfica con los modelos entrenados exportados, es recomendable utilizar librerías o “frameworks” que se basen en el mismo lenguaje de programación con el cual se generaron, entrenaron, evaluaron y exportaron los modelos para evitar problemas de compatibilidad.

7. Bibliografía

- [1] I. Queirolo, “Gestión de alarmas: un punto clave en la planificación de la seguridad,” <http://www.petrotecnica.com.ar/febrero2011/sin/Alarmas.pdf>.
- [2] C. Riley, “How AIOps Revolutionizes Alarm Management,” <https://dzone.com/articles/how-aiops-revolutionizes-alarm-management>, Oct. 08, 2018.
- [3] S. F. and D. L. P. I. DUSTIN BEEBE, “Alarm floods and plant incidents,” <https://www.digitalrefining.com/article/1000558/alarm-floods-and-plant-incidents#.YRM2e4hKjIV>, Jul. 2012.
- [4] REVISTA VISTUAL PRO, “ACCIDENTES INDUSTRIALES|NUMERO 78,” <https://www.virtualpro.co/revista/accidentes-industriales/23>.
- [5] A. S. Bote, D. Kshirsagar, A. Madkaikar, and B. Shah, “Intelligent Based Alarm Management System for Plant Automation,” 2018.

- [6] N. Tamascelli, T. Arslan, S. L. Shah, N. Paltrinieri, and V. Cozzani, “A machine learning approach to predict chattering alarms,” *Chemical Engineering Transactions*, vol. 82, pp. 187–192, 2020, doi: 10.3303/CET2082032.
- [7] J. Xu, S. Zhou, Y. Tang, D. Huang, and Q. Zhu, “Alarm Ranking Model for Intelligent Management of Metro Systems Based on Statistical Machine Learning Methods,” Oct. 2020. doi: 10.1109/PHM-Shanghai49105.2020.9280930.
- [8] Pricing Revenue Managment, “Series de Tiempo.”
- [9] Iberdrola, “QUÉ ES EL ‘MACHINE LEARNING,’” [https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatizado#:~:text=El%20Machine%20Learning%20es%20una,elaborar%20predicciones%20\(an%C3%A1lisis%20predictivo\).](https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatizado#:~:text=El%20Machine%20Learning%20es%20una,elaborar%20predicciones%20(an%C3%A1lisis%20predictivo).), 2022.
- [10] Joaquín Amat Rodrigo, “Máquinas de Vector Soporte (Support Vector Machines, SVMs),” https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines, Apr. 2017.
- [11] Sruthi E R, “Understanding Random Forest,” <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>, Jun. 17, 2021.
- [12] ATRIA Innovation, “Qué son las redes neuronales y sus funciones,” <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>, Oct. 22, 2019.