AIRE (Algoritmo Inteligente de Reconocimiento de Eventos) análisis y visualización de anomalías en archivos log, con técnicas de procesamiento de lenguaje natural.

Fabian Andres Vanegas Beltrán.

Ingeniero Electrónico.

Director.

Gonzalo Enrique Mejía Delgadillo.

Universidad de
La Sabana

Facultad de ingeniería.

Maestría en Analítica Aplicada.

Universidad de la Sabana.

PAGINA DE ACEPTACION.

_____

Profesor Carlos Mera.

_____

Profesor Juan Guillermo Dueñas.

Tabla de contenido

Introducción. 5

Modelo AIRE. 8

Pregunta de investigación. 10

Marco conceptual. 10

Marco teórico. 10

Analizador sintáctico Drain. 11

FastText. 11

Similitud cosenoidal. 12

Métricas 12

Estado del arte. 14

Modelos por tipo de analizador sintáctico Drain. 14

Modelos por tipo de embedimiento. 17

Evaluación de los modelos. 18

Objetivos. 19

Objetivo general. 19

Objetivos específicos. 19

Metodología. 20

Comprensión del del negocio. 20

Entendimiento de los datos. 21

Registros de la impresora digital. 22

Mensajes del Sistema. 23

Preparación de los datos. 24

Preprocesamiento mensajes de registro de la impresora digital. 25

Preprocesamiento mensajes de sistema. 28

Modelamiento. 28

Aplicación parser Drain. 28

Preprocesamiento flujo plantilla Drain. 30

Preprocesamiento flujo mensajes de sistema. 30

FastText. 31

Agrupamiento por similitud cosenoidal. 35
Almacenamiento y reporte de resultados. 37
Resultados. 39
Herramientas utilizadas. 40
Conclusiones. 43
Trabajos futuros. 45
Referencias bibliográficas. 47
Anexos. 53

Índice de Figuras.

Figura 1. Maquina HP índigo 10000. 10
Figura 2. Modelo analizador sintáctico Drain. 11
Figura 3. Muestra del Formato original de los registros de la máquina. 23
Figura 4. Muestra de los mensajes de la prensa compartidos por la fábrica. 24
Figura 5. Descripción general del proceso del registro de eventos para el analizador sintáctico Drain. 25
Figura 6. Muestra del Formato de los registros antes de ser procesados por Drain. 27
Figura 7. Histograma número de caracteres vs longitud del mensaje, obtenidos por Drain. 29
Figura 8. Representación gráfica de los grupos por Kmeans, de los vectores de los registros de la máquina. 33
Figura 9. Representación gráfica de los cluster Kmeans, de los vectores de los mensajes del sistema compartidos por la fábrica. 34
Figura 10. Flujo de información FastText de modelo Aire 35

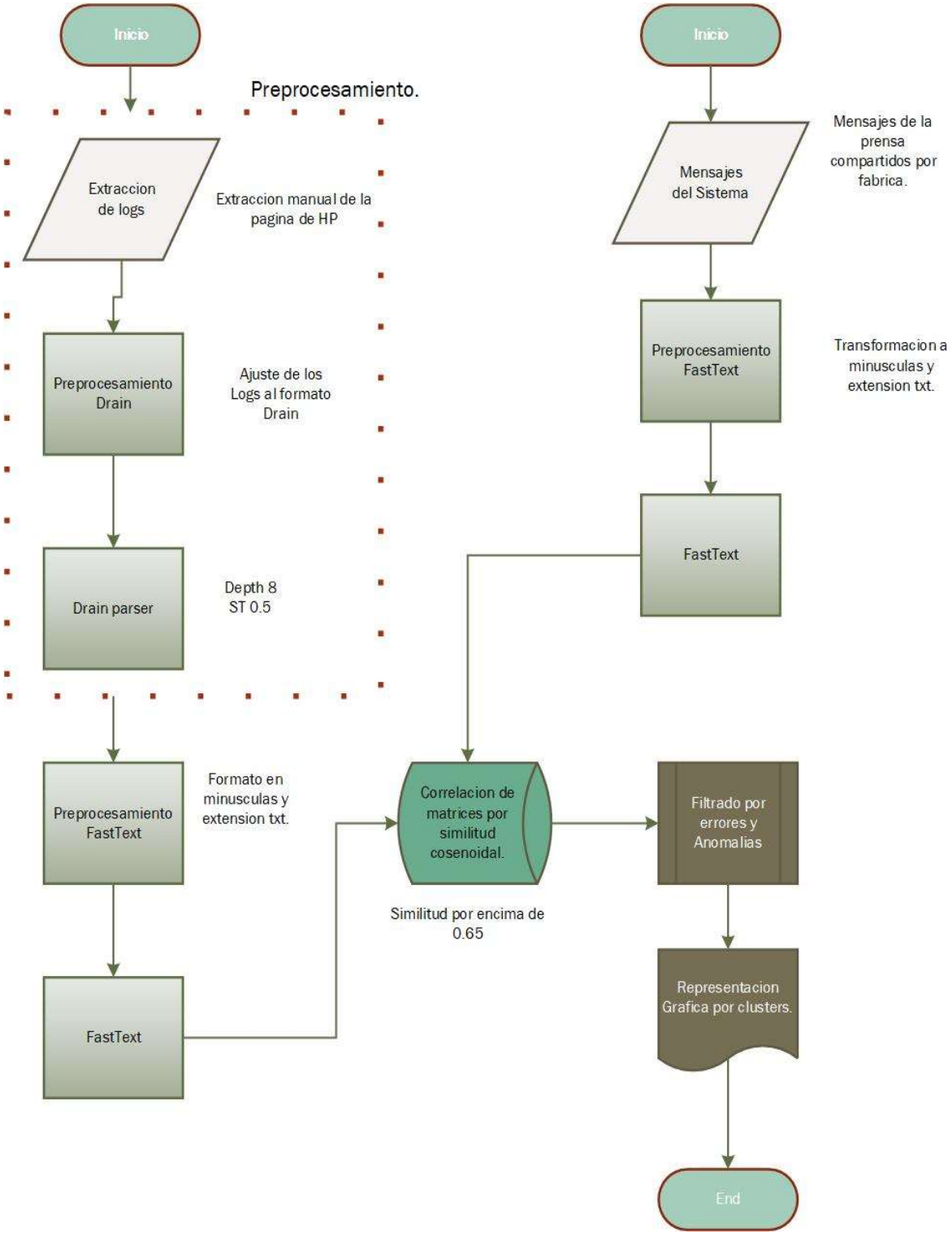
Figura 11. Mapa de calor entre las dos matrices vectoriales mensajes de log (Drain) y los mensajes del sistema. _____ 36
Figura 12. Agrupación realizado por Kmeans de los errores de la prensa. _____ 38
Figura 13. Agrupacion realizado por Kmeans de las advertencias de la prensa. 39
Figura 14. Evaluación del modelo por medio de accuracy ,recall y f1-score, con un total de 186100 de la prensa HP índigo 10000._____ 40
Figura 15. Evaluación del modelo por medio de accuracy, recall y f1-score, con un total de 89768 líneas evaluados por Drain. _____ 55
Figura 16. Evaluación del modelo por medio de accuracy, recall y f1-score, con un total de 436064 líneas._____ 56

Indices de Tablas.

Tabla 1. Comparativa de modelos del estado del arte. _____ 16
Tabla 2. Estructura y descripción del mensaje de registro. _____ 23
Tabla 3. Estructura y descripción del mensaje para ser procesado por Drain. __ 27
Tabla 4. Tamaño de matrices después de reconstrucción vectorial. _____ 32
Tabla 5. Métricas con un total de 89768 líneas evaluados por Drain. _____ 55
Tabla 6. Métricas de la prensa con un total de 436064 líneas evaluados por Drain. _____ 56

RESUMEN GRAFICO.

Arquitectura modelo Aire (Algoritmo Inteligente de Reconocimiento de Eventos).



RESUMEN.

La aplicación de técnicas de procesamiento de lenguaje natural, en registros de captura de información en máquinas de impresión es el enfoque del modelo Aire(Algoritmo Inteligente de Reconocimiento de Eventos) desarrollado para el tratamiento a este tipo de archivos. Estos registros son claves para la mejora continua, detección de fallas, entre otros campos de aplicación. Adicionalmente permite una fácil retro alimentación del modelo, si el fabricante realizara actualizaciones que incluyeran nuevos mensajes tanto de errores como de advertencia de la máquina.

La metodología que se contempló admite la extracción de los archivos de los registros generados diariamente por las maquinas en formato zip, estos son extraídos manualmente de la plataforma que proporciona el fabricante, así realizando la fase de preprocesamiento, con el fin de estandarizar la información a través del flujo de trabajo del modelo.

El analizador sintáctico Drain, transforma las líneas de texto del registro a una estructura morfológicamente basada en árboles, por tanto, Drain disminuye el tiempo de procesamiento y recursos a nivel computacional en comparación a métodos que utilizan estructuras fijas, ajustes manuales extensos y/o sistemas que analizan línea por línea, además Drain permite ser escalable así este algoritmo aumenta su aplicación en diferentes proyectos.

En un nivel paralelo del flujo de trabajo se procesaron el listado de mensajes de la impresora digital que se han generado, por medio de diversas versiones de software desarrolladas por la fábrica.

FastText fue la técnica elegida para construir la matriz de cada elemento del mensaje, descomponiendo el mensaje del registro de información, en fracciones las cuales se encuentre su valor vectorial, en una etapa simultanea del modelo se genera la segunda matriz a partir de los mensajes compartidos por la fábrica; las

dos matrices son reconstruidas en base tanto a la plantilla de Drain como los mensajes de la fábrica, así encontrar la relación por medio de la medida de similitud cosenoidal, filtrando la selección de fallas y advertencias del sistema de la prensa, finalmente, se realiza el agrupamiento de los mensajes en clústeres, relacionando el sub sistema de la máquina.

Los resultados de la evaluación del modelo muestran que, al aumentar el umbral de similitud cosenoidal, se observa una reducción en el recall. Esto se debe a que el modelo se vuelve más estricto en la definición de coincidencia, lo que resulta en la eliminación de algunos mensajes relevantes de la salida del modelo. Esta mayor selectividad conlleva a una disminución en el f1-score, que es la media armónica de la accuracy y el recall, reflejando un compromiso entre la capacidad del modelo para recuperar todos los mensajes relevantes y su accuracy en la clasificación. Sin embargo, el accuracy permanece alto, ya que las matrices FastText-Drain y los mensajes de la fábrica muestran una similitud considerable. Estos hallazgos destacan la importancia de ajustar el umbral de similitud para optimizar el rendimiento del modelo en la identificación de errores en los registros de la prensa HP Índigo 10000.

La implementación de esta metodología provee un avance significativo en el mantenimiento y diagnóstico de maquinaria de impresión digital, dado la adaptabilidad del modelo, la clasificacion a través de la variación de similitud cosenoidal y siendo escalable sin importar los días de recolección de información, abriendo nuevas posibilidades para la detección temprana de fallas y la optimización operativa.

ABSTRACT.

The application of natural language processing techniques to information capture logs in printing machines is the focus of the AIRE model (Intelligent Event Recognition Algorithm) developed for the treatment of such files. These logs are crucial for continuous improvement, fault detection, and other fields of application.

Additionally, it allows for easy feedback to the model if the manufacturer makes updates that include new errors or warning messages from the machine.

The methodology considered involves extracting the log files generated daily by the machines in zip format, which are manually extracted from the platform provided by the manufacturer, thus conducting the preprocessing phase to standardize the information through the workflow of the model.

The Drain parser transforms the text lines of the log into a morphologically tree-based structure, thereby reducing the processing time and computational resources compared to methods that use fixed structures, extensive manual adjustments, and/or systems that analyze line by line. Moreover, Drain allows for scalability, thus increasing its application in different projects.

At a parallel level of the workflow, the list of messages from the digital printer that have been generated, through various software versions developed by the factory, were processed. FastText was the chosen technique to construct the matrix of each element of the message by decomposing the information log message into fractions, each found to have a vector value. In a simultaneous stage of the model, the second matrix is generated from messages shared by the factory; the two matrices are reconstructed based on both the Drain template and the factory messages, thus finding the relationship through the measure of cosine similarity, filtering the selection of faults and warnings from the press system, and finally clustering the messages, relating to the machine's subsystem.

The implementation of this methodology provides a significant advancement in the maintenance and diagnostics of digital printing machinery, given the model's

adaptability, classification through the variation of cosine similarity, and scalability regardless of the days of information collection, opening new possibilities for early fault detection and operational optimization.

Introducción.

El registro de información (log, termino en inglés) se define como un archivo que resulta de la compilación de eventos de un sistema, donde se captura línea por línea de texto el proceso que se está realizando en un lapso de tiempo determinado, como resultado, se crean múltiples líneas de texto.

Este tipo de archivo está ligado a un sistema de telemetría (adquisición de datos), ya sea manual o automatizado por tanto dependiendo del modo de captación de datos(telemetría), aumentara la complejidad de este archivo.

Es importante destacar que actualmente no existen protocolos ni estándares unificados para la creación de estos archivos, lo que resulta en reportes semiestructurados y sin un formato consistente. Típicamente, estos registros incluyen elementos clave como una marca de tiempo (timestamp), un nivel de severidad (severity level) y un mensaje de registro (log message), pero la falta de uniformidad puede complicar su análisis y procesamiento.

Los registros de información permiten encontrar tipos de anomalías [1] tales como:

Anomalía puntual: que describe un solo evento, por ejemplo, el cambio de temperatura en el motor de un automóvil que se indicó como una anomalía.

Anomalías colectivas: son acciones que se registran por determinado tiempo o filtro de selección, por ejemplo, el registro de la temperatura del cambio del motor en un día.

Anomalías contextuales: son eventos, que rodean el evento, así como, la temperatura del motor en el recorrido que está haciendo el automóvil o el nivel de aceite del motor.

En la actualidad se utilizan diversas técnicas para el análisis de los registros que implica una verificación manualmente línea por línea, por medio de un

experto(humano), perdiendo el contexto que rodea la falla, aumentando la probabilidad de error, este método no es escalable, ya que al aumentar la información incrementa la complejidad de la observación.

Los sistemas automatizados en los cuales ya permiten verificar de una manera escalable los registros de información con mayor rapidez computacional, así, el usuario pueda observar tendencias o predicciones de los eventos.

El análisis y estudio de estos archivos ha crecido su uso en los últimos años a nivel global, en el ambiente de monitoreo de ciberataques, la exposición de información se ha incrementado exponencialmente a causa de las interconexiones de los sistemas. Esto lo demuestra el informe realizado por la policía nacional del 2022[38], el sector financiero aumentado las denuncias de ciber ataques en un 11.8% y las entidades de gobierno en un 10% con respecto al año anterior del reporte, también lo ratifica el informe realizado por la empresa privada especialista en el análisis de registros[39] en donde a nivel mundial, Latinoamérica se han incrementado los ataques cibernéticos en un %5 con relación del 2021 al 2022, siendo Estados Unidos el país con mayor incremento de ataques en 65%.

Por otra parte el estudio y análisis de los registros en aplicaciones como internet de las cosas (IOT), comunicación 5G o automatización de procesos industriales son fundamentales, dado que este tipo de archivos ayudan a la corrección a corto y mediano plazo de anomalías que presente los sistemas, por ende, los desarrolladores o ingenieros de campo pueden desplegar planes de solución a las fallas, tal como se ve en solución de bugs(errores de software) en los sistemas operativos o funcionamiento de maquinaria.

Uno de estos análisis de registros, usa minería de datos, de la mano con técnicas procesamiento de lenguaje natural [6], esta investigación propone un enfoque al proceso de estudio de los registros, transformando los datos de una forma semi estructurada a texto estructurado, encontrando procesos internos para inferir el funcionamiento del sistema.

Esta aplicación promueve e introduce la importancia de los parsers (analizadores sintácticos), que son algoritmos, construidos con múltiples técnicas matemáticas, para extraer la información del registro y trasformar esta información en un archivo ordenado morfológicamente, evitando suposiciones o prejuicios frente a la información, además con técnicas de aprendizaje de maquina incrementan su potencial, como resultado, los datos pueden ser clasificados eficiente y eficazmente.

Un analizador sintáctico en general, tiene 4 fases, preprocesamiento, una etapa de normalización del registro de la información, ya sea, transformar los mensajes a letra minúscula. La segunda fase es la clasificación, en la cual el algoritmo aplica las reglas para reconocer patrones entre las líneas de los registros, es decir puede realizar clasificaciones por fecha o por severidad entre otros. El post procesamiento es la etapa en donde el registro obtiene una forma estructurada, por último, la extracción de la plantilla, en donde se puede puntualizar en qué tipo de formato o extensión va a ser el archivo con la información previamente procesada.

Por otra parte, un modelo de embedimiento de palabras es una técnica en procesamiento de lenguaje natural (NLP) que transforma palabras y frases en vectores numéricos. Esta transformación permite a las máquinas interpretar texto de manera similar a como los humanos entienden el lenguaje, al representar semánticamente las palabras en un espacio dimensional donde las palabras con significados similares están ubicadas cerca unas de otras. La importancia de estos modelos radica en su capacidad para mejorar significativamente el rendimiento de diversas aplicaciones de NLP, como la traducción automática, la clasificación de texto, y los sistemas de recomendación, donde una comprensión profunda del contexto y el significado es esencial.

Entre los modelos más comunes y eficaces se encuentran Word2Vec, FastText y GloVe. Word2Vec, desarrollado por Google, utiliza una red neuronal para aprender representaciones vectoriales de palabras a partir de grandes conjuntos de datos textuales, destacando por su habilidad para capturar relaciones contextuales y semánticas. FastText, creado por Facebook, extiende esta idea al tratar cada

palabra como un conjunto de n-gramas de sub-palabras, lo que le permite manejar mejor las palabras fuera de vocabulario y entender la morfología de idiomas diversos. GloVe, desarrollado por Stanford, combinas técnicas de factorización matricial con estadísticas globales de co-ocurrencia de palabras de un corpus, proporcionando una rica captura de relaciones semánticas y sintácticas. Estos modelos son fundamentales en el avance de tecnologías de NLP, permitiendo que las aplicaciones no solo reconozcan palabras, sino que también comprendan su uso y significado en diferentes contextos.

Modelo AIRE.

El modelo AIRE, cuyas siglas representan "Análisis Inteligente de Registros de Eventos", se presenta como una solución innovadora para la monitorización y el mantenimiento automatizado de sistemas industriales, con un enfoque específico en las impresoras digitales HP Índigo. Basado en conceptos avanzados de procesamiento de texto y aprendizaje automático, este modelo aborda el desafío de analizar registros de eventos semi-estructurados y heterogéneos, con el propósito de identificar patrones, errores y advertencias que puedan incidir en el rendimiento de las impresoras. A nivel conceptual, el modelo AIRE busca interpretar y procesar grandes volúmenes de datos de registros de eventos generados por las impresoras HP Índigo, con el fin de optimizar el diagnóstico de fallas, la detección de problemas y la aplicación de medidas correctivas y preventivas.

AIRE no se limita a un enfoque estadístico, sino que integra técnicas avanzadas de procesamiento de texto y aprendizaje automático para cumplir sus objetivos. No es meramente una metodología de gestión de información, sino un modelo integral diseñado para entender, interpretar y extraer conocimiento útil de los registros de eventos de las impresoras digitales. Tampoco se restringe a un modelo de

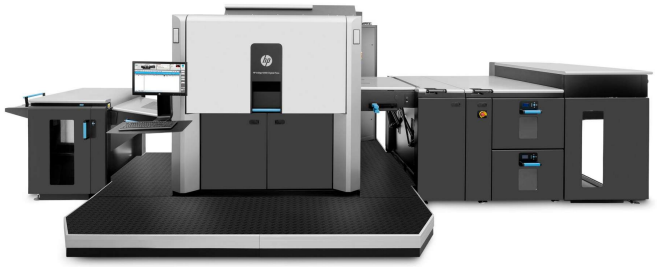
procesamiento de textos, ya que incorpora elementos de análisis de datos y detección de patrones. Es más adecuado describirlo como un modelo de aprendizaje automático aplicado al análisis de registros de eventos.

Las herramientas clave empleadas en el modelo son Drain y FastText. Drain es un analizador sintáctico diseñado para estructurar y organizar datos de registros de eventos, mientras que FastText es un modelo de representación de palabras en forma de vectores que facilita la interpretación de texto mediante algoritmos de aprendizaje automático. Ambas herramientas son fundamentales para procesar y analizar los datos de registros de eventos de las impresoras HP Índigo.

Los datos utilizados en esta investigación consisten en registros de eventos generados por las impresoras digitales HP Índigo. Estos datos constituyen un volumen considerable, con más de 185 mil líneas recopiladas durante tres días consecutivos de funcionamiento de la prensa HP Índigo 10000, aunque en los anexos se evaluaron diferentes cantidades de líneas, desde alrededor de 80,000 hasta 460,000 aproximadamente. Los datos son específicos del servicio y producto de las impresoras digitales HP Índigo y no están necesariamente disponibles como datos de acceso abierto. Sin embargo, pueden obtenerse a través de los sistemas de monitoreo y gestión de HP o mediante solicitudes directas a la empresa.

Con lo anteriormente descrito legitima el estudio de estos registros, su análisis y extracción de valiosa información aplicado en este caso en las máquinas de impresión digital desarrolladas por HP(Hewlett-Packard) Índigo en su serie 1X000(figura 1), proporcionando el diagnostico de fallas en campo, generando planes de mantenimiento preventivo y correctivo, por consecuencia aumentando la estabilidad de la prensa e incrementando la satisfacción del cliente.

Figura 1. Maquina HP índigo 10000.



Nota. Modelo Hp Indigo 10000 de hojas. Tomado de (NeoBIs, 2021)

Pregunta de investigación.

¿De qué manera las técnicas específicas de procesamiento de lenguaje natural, como el análisis sintáctico mediante Drain y el modelado de palabras con FastText, pueden ser aplicadas para extraer y analizar eficazmente información de los registros semiestructurados de las máquinas de impresión digital, facilitando la detección de fallas del sistema y permitiendo su efectivo agrupamiento?

Marco conceptual.

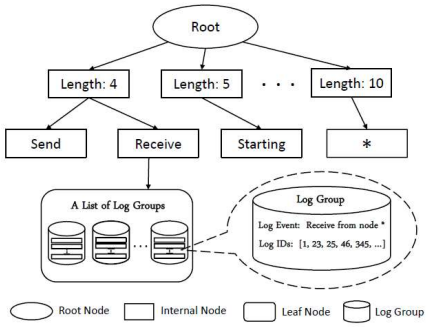
Marco teórico.

El modelo AIRE es aplicado en su totalidad a un entorno diseñado en el procesamiento de lenguaje natural (NLP) que incluye la fase realizada por Drain, FastText, y agrupamiento de fallas y advertencias(clusters), además de ser evaluado por medio de métricas, para visualizar la eficacia del modelo.

Analizador sintáctico Drain.

Drain[29] fue diseñado específicamente para manejar registros con estructuras de árbol, facilitando así la interpretación y personalización en el procesamiento de datos no estructurados. Este algoritmo se basa en la aplicación de reglas preestablecidas para convertir las líneas de texto de los registros en formatos estructurados aptos para análisis posterior. Drain se caracteriza por su operación mediante dos hiper-parámetros clave: la 'profundidad', que define el número de nodos en la estructura del árbol para el análisis (figura 2); y la 'longitud', que establece el número de subunidades de caracteres en cada línea de mensaje del registro. Esta configuración permite que el analizador sintáctico se adapte eficientemente a cambios en la generación de registros, resultantes de actualizaciones en los sistemas.

Figura 2. Modelo analizador sintáctico Drain.



Nota. Modelo estructura del parser Drain. Tomado de Drain: An Online Log Parsing Approach with Fixed Depth Tree, 2017

FastText.

FastText [40] es una herramienta desarrollada por Facebook que facilita la vectorización detallada de palabras al descomponerlas en subunidades, conocidas como n-gramas de caracteres. Este enfoque es fundamental para el análisis de registros de máquinas de impresión digital y otros contextos donde el lenguaje puede ser técnico o contener errores ortográficos. Al tratar las palabras como combinaciones de n-gramas, FastText permite capturar la estructura y similitudes

morfológicas, lo que es especialmente útil en idiomas con rica morfología o en datos con variantes de palabras no estándar. Optimizado para rapidez y escalabilidad, preserva el significado semántico de las palabras y minimiza la pérdida de vocabulario. Este modelo puede ser entrenado en grandes corpus de texto y luego aplicado eficientemente para mejorar la precisión y el alcance del análisis de texto.

Similitud cosenoidal.

La similitud cosenoidal mide el coseno del ángulo formado entre dos vectores (a y b) en un espacio vectorial. Se calcula como el producto escalar de los dos vectores dividido por el producto de sus magnitudes (||a|| y ||b||). Este cálculo refleja cómo de similares son los vectores en términos de orientación, sin considerar su magnitud. La similitud cosenoidal es especialmente útil para comparar los registros en análisis de texto, donde permite determinar qué tan relacionados están los contenidos de dos fuentes diferentes, basándose en sus vectores de características. La fórmula es:

$$Similitud\ cosenoidal = \frac{a\ .b}{||a||||b||} \quad (1)$$

Métricas

Accuracy.

Métrica que indica el porcentaje total de predicciones correctas realizadas por el modelo, incluyendo tanto los positivos como los negativos correctamente identificados. La fórmula es:

$$Accuracy = \frac{TP+TN}{Total\ de\ casos} \quad (2)$$

Donde:

TP : son los Verdaderos Positivos.

TN: son los Verdaderos Negativos.

FP: son los Falsos Positivos.

FN: son los Falsos Negativos.

Recall.

El Recall, también conocido como la tasa de verdaderos positivos o sensibilidad, mide la capacidad del modelo para identificar correctamente todas las instancias positivas. Es crítico en situaciones donde es importante capturar todos los casos relevantes, como en la detección de enfermedades. Se calcula con la siguiente fórmula:

$$Recall = \frac{TP}{TP+F} \quad (3)$$

F1-score.

Es una medida que permite relacionar tanto el accuracy como el recall en un solo número, permitiendo el balance entre estas dos medidas, se podría determinar qué tan armónicas son estos dos indicadores en el modelo.

$$F1\ score = 2\ x\ \frac{Accuracy\ x\ Recall}{Accuracy\ +Recall} \quad (4)$$

Estado del arte.

Modelos por tipo de analizador sintáctico Drain.

La evaluación de distintos modelos de procesamiento muestra la superioridad de Drain como analizador sintáctico con una precisión del 86.54% [40] respecto a los demás analizadores, lo que lo distingue por su capacidad robusta para clasificar registros de información de manera precisa. Su eficacia, que no se ve afectada por variaciones mínimas de preprocesamiento y la ausencia de requerimientos de entrenamiento fuera de línea, facilita su integración en entornos de tiempo real. Además, su diseño simple con solo dos hiperparámetros esenciales permite una configuración y adaptación sencillas, promoviendo su aceptación generalizada para el análisis avanzado de registros.

Modelos notables que integraron Drain en sus arquitecturas, como LogGPT[22], LogQA[23] y Robustlog[20] (Tabla 1), destacan la contribución significativa del analizador sintáctico en sus operaciones. LogGPT lo emplea en la extracción y clasificación inicial de datos de registros, facilitando así la elaboración de reportes detallados y la propuesta de medidas preventivas a través de ChatGPT. De igual manera, LogQA aprovecha la capacidad de Drain para transformar los registros en estructuras morfológicas de árbol, posibilitando la realización de consultas directas sobre las características de los registros. Este enfoque abre la puerta a análisis más profundos e interactivos.

El modelo Deeptralog[25], que se enfoca en el análisis de microservicios, implementa Drain para convertir eventos en líneas estructuradas, demostrando la flexibilidad del analizador al integrarse con herramientas como Glove y TF-IDF para mejorar la representación y ponderación de eventos. Este enfoque holístico brinda un panorama completo del funcionamiento de sistemas complejos.

Finalmente, Robustlog[20] aplica Drain en su etapa de preprocesamiento, eliminando contenido irrelevante y estableciendo una base sólida para la

identificación de patrones dentro de grandes volúmenes de registros de información. La combinación de Drain con FastText y el análisis de frecuencias mediante TF-IDF potencia la habilidad del modelo para distinguir entre eventos normales y anomalías, optimizando así la detección de inestabilidades en los sistemas.

En conclusión, la implementación de Drain en estos modelos subraya su valor como un analizador sintáctico esencial. Su capacidad para adaptarse a diferentes contextos y su compatibilidad con avanzadas técnicas de aprendizaje automático confirman la elección de Drain como un componente fundamental en el análisis de registros de información moderno. Como se refleja en la tabla 1. Este resumen presenta distintos modelos que han incorporado a Drain, destacando su uso práctico y las bases de datos sobre las cuales se han evaluado.

Tabla 1. Comparativa de modelos del estado del arte.

Modelo	Preprocesamiento	Modelo NLP	Modelo ML	Métricas del modelo	Data Sets
Log2Event[7]	Baricentro – TF IDF	Word2vec	El mejor de: Random Forest, Naive-Bayes, Red Neuronal.	F1-score, Recall, Accuracy	BGL
log Anomaly Deteccion[9]	Tokenizacion	BPE	Modo comparativo de vectores	Velocidad de procesamiento	Sandia laboratorios nacionales en Albuquerque
Bertlog.[13]	Tokenizacion	BERT	Red Neuronal	Accuracy, precisión, F1-score, Recall	HDFS
HitAnomaly.[14]	Parser Drain	Word2Vec	Drain	precisión, F1-score, Recall	HDFS. BGL, Open Stack
LogUAD[15]	TF-IDF	TD-IDF, Word2vec	Clúster	F1-score	BGL.
Lightlog.[17]	PCA, PPA	Word2vec	TCN	F1-score, Recall, Precision.	BGL, HDFS
DeepLog.[18]	N-gram	None	LSTM	F1-score, Recall, Precision.	HDFS, OpenStack
LogAnomaly[19]	Parsing FT-Tree	Template2vec	LSTM	F1-score, Recall, Precision.	BGL, HDFS
RobustLog.[20]	Parsing Drain	FastText – TF-IDF	Bi-LSTM	F1-score, Recall, Acuraccy	HDFS
LogMine[21]	Tokenizacion	Distancia entre Clusters	Cluster	Accuracy, tiempo de procesamiento.	6 tipos de diferentes datos, de diferentes no públicos,
LogQA[23]	Parsing Drain	BERT	Interfaz propia	F1-score, Recall, Precision.	HDFS. OpenSSH, Spark
LogGPT[22]	Parsing Drain	GPT	Interfaz	Comparación con otros modelos, F1-score	BGL, Spirit.
ConAnomaly[24]	POS	Log2vec-Word2vec	LSTM	F1-score, Recall, Acuraccy.	BGL, HDFS.
DeeptraLog.[25]	Parsing Drain	Glove – TF-IDF	GGNN	F1-score, Recall, Precision.	Microservicion industriales no publicos
Syslog.[26]	FT-tree	FT-Tree	Statistical Analysis	Aleatorio Index, para medición de similitud.	Proveedor servicion en la nube
Swisslog[27]	Parsing Diccionarios - LCS	BERT	BI-LSTM	Precisión, Recall, F1-score, accuracy parsing	HDFS,BGL, Android, Hadoop.

Nota. modelos analizados por características del modelo. Diseño propio por el autor del proyecto de grado.

Modelos por tipo de embedimiento.

Los modelos de embedimientos son fundamentales en el procesamiento de lenguaje natural, permitiendo que las máquinas procesen texto de manera que refleje su significado semántico y sintáctico.

Word2Vec.

Modelo reconocido por su capacidad para mapear semánticamente palabras similares en espacios vectoriales cercanos, como se evidencia en su aplicación dentro de modelos como Logmine[21] y Deeplog[18], donde la técnica ha permitido un análisis más profundo de los registros de información y la detección de anomalías en registros de sistemas complejos como también en el modelo LogEvent2Vec[7], Word2Vec ayuda a convertir registros complejos de sistemas de red en vectores multidimensionales, lo que permite la identificación temprana de actividades sospechosas o no autorizadas.

FastText.

Integra y amplía las capacidades de modelos preexistentes como Word2Vec, destaca especialmente por su enfoque en el tratamiento de subpalabras. Esta funcionalidad avanzada no sólo permite una comprensión más detallada y matizada de la estructura interna de las palabras, sino que también juega un papel fundamental en la prevención de la pérdida de vocabulario, asegurando así una cobertura más amplia y efectiva de términos tanto comunes como emergentes. Este enfoque es esencial para manejar vocabularios diversificados en campos en constante evolución. Adicionalmente, FastText se posiciona como una extensión lógica de modelos como LogQA[23] y Robustlog[20], que dependen de una comprensión morfológica exhaustiva para lograr una clasificación y análisis más precisos de registros y datos textuales.

GloVe.

Es otra técnica poderosa que se utiliza para capturar tanto la co-ocurrencia como las relaciones semánticas complejas en grandes conjuntos de datos. Los modelos como Deeptralog[25], que emplean técnicas de embedimiento para analizar y representar microservicios en un sistema complejo, podrían beneficiarse de la dimensionalidad reducida y las ricas relaciones semánticas que GloVe proporciona.

En adición a estos, se presentan técnicas como BPE (Byte Pair Encoding)y BERT, que representan métodos avanzados. BPE, por ejemplo, es particularmente útil en Log anomaly[19]para identificar anomalías en registros de sistemas, mientras que BERT se aplica en BERtLog[13] para automatizar la clasificación de eventos de logs, gracias a su sofisticado entendimiento del contexto y la semántica.

Evaluación de los modelos.

Las métricas de Accuracy, F1-Score y Recall son fundamentales para evaluar modelos de aprendizaje automático, particularmente en la detección de anomalías. Accuracy es efectiva cuando las clases están balanceadas y refleja la proporción general de predicciones correctas. El F1-Score es crucial cuando las clases están desbalanceadas, ya que combina accuracy y el recall en una métrica que equilibra la identificación de verdaderos positivos y la exclusión de falsos positivos. El Recall es prioritario en contextos donde los costos de los falsos negativos son significativos, asegurando que la mayoría de los eventos positivos sean correctamente identificados.

Los modelos proporcionados en la tabla 1, en su gran mayoría fueron evaluados en fundamento a datos públicos, de esta manera realizar una comparación entre modelos por las medidas explicadas anteriormente, a continuación, se realiza una breve reseña de las bases de datos.

Los conjuntos de datos como BGL y Spirit, con millones de mensajes de supercomputadoras y clústers de Linux, respectivamente, proporcionan una rica fuente de datos para probar la efectividad de los modelos de detección de anomalías en sistemas complejos y altamente paralelos. Mientras tanto, los conjuntos de datos de HDFS y OpenStack permiten la evaluación en entornos de procesamiento distribuido y computación en la nube, con una proporción significativa de registros anómalos que imitan situaciones del mundo real. Estos conjuntos de datos son esenciales para el desarrollo y la validación de modelos robustos, que deben ser capaces de manejar la diversidad y la escala de los desafíos presentados por estos entornos. La información más detallada de estas bases de datos se puede observar en el anexo 1.

Objetivos.

Objetivo general.

Optimizar el diagnóstico de anomalías en dispositivos mediante técnicas avanzadas de procesamiento de lenguaje natural (NLP) aplicadas al análisis de archivos de registro (logs), creando las agrupaciones de eventos para una identificación y análisis eficientes de las fallas.

Objetivos específicos.

- Recolectar diversos modelos de procesamiento de lenguaje natural (NLP) y seleccionar el más adecuado para el análisis de archivos de registro (logs) basándose en su eficiencia, precisión y capacidad de adaptarse al contexto específico del diagnóstico de fallas.
- Adaptar los registros a la estructura del analizador sintáctico Drain para optimizar el procesamiento de la información manteniendo el contexto relevante para la detección de anomalías.

- Desarrollar un modelo de NLP basado en la técnica seleccionada que identifique anomalías y advertencias mediante análisis semántico y procesos de embedimiento.
- Implementar herramientas de visualización para la interpretación intuitiva de las agrupaciones de clústeres, facilitando la identificación de fallas y el análisis de subsistemas.

Metodología.

Para el desarrollo del modelo AIRE se aplica la metodología Crisp DM diseñada por IBM, esta metodología y sus fases incluyen: comprensión del negocio, entendimiento de los datos, preparación de los datos, modelamiento, evaluación y despliegue, las cuales se desarrollan en el modelo, a pesar de realizar la mayoría de las fases, la única fase que se descarta fue la fase de despliegue que se reserva para trabajos futuros.

Comprensión del del negocio.

La eficiencia en el diagnóstico de fallas mediante el análisis de registros de información es esencial para garantizar la operación óptima y el mantenimiento de las impresoras digitales HP Índigo, afectando directamente la satisfacción del cliente. Sin embargo, este análisis enfrenta desafíos significativos debido a la naturaleza semiestructurada de los registros de información, a la constante evolución de las fallas y configuraciones del sistema, lo que incrementa la complejidad del diagnóstico.

Para abordar estos retos, se han seleccionado herramientas avanzadas como Drain y FastText, cuya eficacia ha sido probada en entornos tanto industriales como académicos. Drain organiza y estructura los datos de registros de información, mientras que FastText facilita su interpretación a través de algoritmos de

aprendizaje automático, proporcionando una solución integral y adaptativa que ha sido ampliamente adoptada en la industria, respaldada por desarrolladores de renombre como Facebook.

En la Tabla 1, se muestra que se han estado desarrollando modelos con diferentes aproximaciones para encontrar la mejor rapidez y clasificación de eventos, que incluyen advertencias y errores. Estos modelos están diseñados para mejorar la precisión y eficiencia del proceso de diagnóstico.

Los registros de información tienen un papel esencial no solo en la impresión digital, sino también en una variedad de sectores críticos. En la seguridad, estos registros son fundamentales para el monitoreo de dispositivos esenciales, como cámaras de seguridad y radioteléfonos, permitiendo la reconstrucción precisa de eventos significativos para la sociedad, tales como robos y trayectos. Adicionalmente, en el campo aeronáutico, los registros de información son utilizados para monitorear y evaluar el rendimiento de las aeronaves, lo que facilita la mejora continua de los productos y asegura la seguridad y eficiencia operacional. Estas aplicaciones demuestran la versatilidad y el valor estratégico de los registros en múltiples contextos, subrayando su importancia en la toma de decisiones y en la mejora de los servicios y productos ofrecidos entre otros campos.

El resultado de implementar estas soluciones es una mejora sustancial en la fiabilidad y estabilidad de las impresoras, lo que permite al ingeniero de soporte ofrecer acciones correctivas y preventivas eficaces, culminando en una mayor satisfacción del cliente.

Entendimiento de los datos.

Se identificó la página web del fabricante como la fuente principal de los datos, diseñada específicamente para la recolección de eventos de las máquinas de impresión. Desde esta página, se descarga manualmente la librería diaria de archivos ZIP, que compila información relevante sobre los eventos registrados por las prensas HP Índigo. Este archivo incluye los datos recolectados desde el

encendido hasta el apagado de la máquina. Este paso fue crucial para asegurar tanto la autenticidad como la relevancia de los datos descargados, elementos fundamentales para el análisis posterior.

Registros de la impresora digital.

Los archivos de extensión `.log`, accesibles como archivos de texto, se compilan dentro del archivo ZIP principal y están diseñados específicamente para documentar los eventos de las maquinas HP Índigo. Se descargan desde la página web de la compañía, que está estructurada para almacenar y gestionar los datos de los equipos de la serie HP Indigo 1X000. El tamaño de cada archivo `.log` varía según el uso de la prensa en producción, ya que captura el funcionamiento de todos sus dispositivos durante el día.

Con cada reinicio de la máquina, se genera un nuevo archivo `.log` que se incluye en el ZIP diario. Estos archivos proporcionan un registro exhaustivo de las operaciones de las máquinas, conteniendo aproximadamente archivos mayores a 80.000 líneas de mensajes que reflejan los registros digitales de la impresora. La estructura típica encontrada en los mensajes es la siguiente:

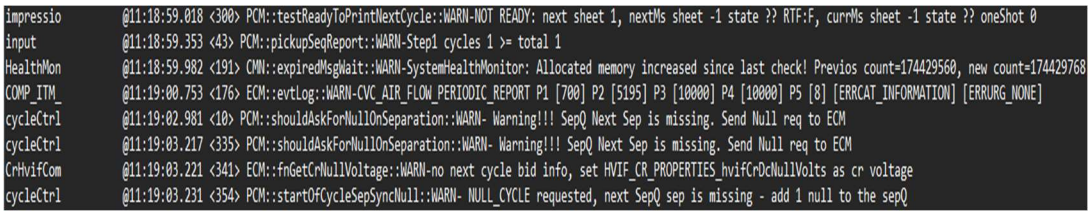
Tarea @Hora:minuto:segundo:milisegundos_<identificador>_componente:mensaje

La estructura de estos mensajes es clave para entender los eventos y operaciones documentadas. Cada línea sigue un formato específico, descrito detalladamente en la tabla 2. En la figura 3 se puede observar una muestra del archivo original como están organizados los mensajes.

Tabla 2. Estructura y descripción del mensaje de registro.

Estructura	Descripción
Tarea	Muestra que rutina del funcionamiento de la maquina está realizando.
@Hora:minuto:segundo:milisegundos	Indica el lapso en donde se reporta el evento.
_<identificador>_componente	Provee la información del componente que está realizando la rutina
mensaje	ofrece detalles sobre la acción del componente, en relación con dispositivos externos como sensores.

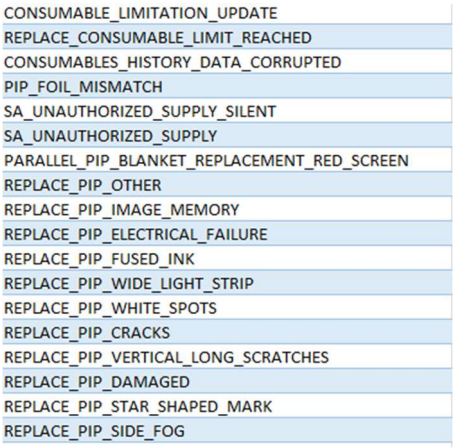
Figura 3. Muestra del Formato original de los registros de la máquina.



Mensajes del Sistema.

Los mensajes del sistema, obtenidos directamente de la fábrica, son mensajes que documentan errores y eventos significativos de las impresoras HP Índigo, capturados a través de las diversas versiones de software instaladas en la máquina. Este archivo específico contiene un listado de 4,280 mensajes compilados por el fabricante, reflejando una amplia gama de operaciones y estados de la máquina. La muestra de los mensajes se puede ver en la figura 4.

Figura 4. Muestra de los mensajes de la prensa compartidos por la fábrica.



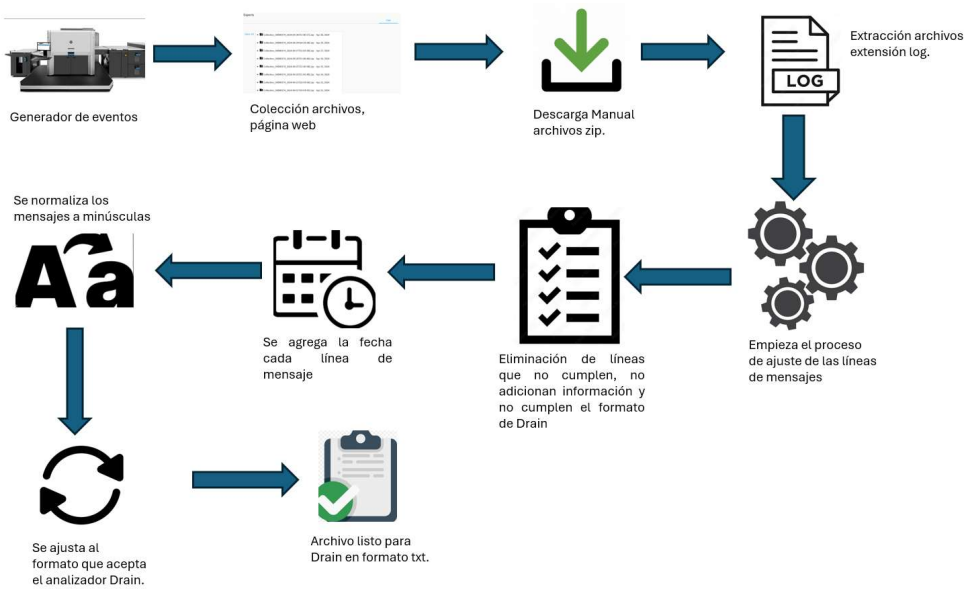
CONSUMABLE_LIMITATION_UPDATE
REPLACE_CONSUMABLE_LIMIT_REACHED
CONSUMABLES_HISTORY_DATA_CORRUPTED
PIP_FOIL_MISMATCH
SA_UNAUTHORIZED_SUPPLY_SILENT
SA_UNAUTHORIZED_SUPPLY
PARALLEL_PIP_BLANKET_REPLACEMENT_RED_SCREEN
REPLACE_PIP_OTHER
REPLACE_PIP_IMAGE_MEMORY
REPLACE_PIP_ELECTRICAL_FAILURE
REPLACE_PIP_FUSED_INK
REPLACE_PIP_WIDE_LIGHT_STRIP
REPLACE_PIP_WHITE_SPOTS
REPLACE_PIP_CRACKS
REPLACE_PIP_VERTICAL_LONG_SCRATCHES
REPLACE_PIP_DAMAGED
REPLACE_PIP_STAR_SHAPED_MARK
REPLACE_PIP_SIDE_FOG

Nota. Muestra de los mensajes de la prensa compartidos por la fábrica. (Indigo, 2021)

Preparación de los datos.

Durante esta fase de CRISP-DM, se realizaron ajustes a los datos contenidos en archivos con extensión `.log`, recolectados del archivo ZIP principal. Estos ajustes fueron necesarios para normalizar y facilitar el procesamiento mediante el analizador sintáctico Drain. En paralelo, se ajustó el listado de mensajes de la máquina proporcionado por el fabricante, el resumen grafico del proceso de la captura y transformación del archivo para el procesamiento del analizador sintáctico Drain se puede observar en la figura 5.

Figura 5. Descripción general del proceso del registro de eventos para el analizador sintáctico Drain.



Preprocesamiento mensajes de registro de la impresora digital.

Ajuste a Formato Drain:

Al descargar manualmente el archivo zip, se utiliza la librería `zipfile` en Python como parte del proceso de AIRE para abrir y buscar archivos .log. Es crucial mencionar que AIRE puede extraer información de múltiples zips de manera simultánea, generando varios archivos logs sin modificar la fecha original de creación de estos archivos. Se procede entonces a totalizar el tamaño final de los archivos recolectados, garantizando que toda la data ha sido efectivamente agregada.

Durante el proceso de filtrado, se revisan las líneas para asegurar que cumplan con la estructura inicial descrita en la Figura 3. Este paso permite eliminar líneas no

deseadas, como banners y secuencias de conexión representadas por caracteres especiales. A las líneas que sí cumplen con la estructura, se les añade la fecha de generación del archivo, información extraída del nombre del archivo original. Posteriormente, todas las líneas son convertidas a minúsculas y se reorganiza el orden de los mensajes para optimizar el procesamiento por parte de Drain y cumplir con su formato de análisis sintáctico. La normalización de la fecha facilita la organización cronológica de los mensajes.

Para asegurar que toda la información recolectada y ajustada sea evaluada eficientemente por Drain, se adopta el procesamiento por paquetes. Debido a las complejidades y limitaciones computacionales, esta técnica permite procesar los archivos en paquetes sin afectar la secuencialidad de la información. Esto es crucial, ya que la fecha de los archivos, que incluye hora, minutos y milisegundos, ayuda a mantener el orden cronológico. De esta manera, Drain puede evaluar toda la información utilizando una estructura de árboles. Además, se lleva a cabo una reducción de mensajes, ya que Drain permite eliminar líneas duplicadas o aquellas con estructuras similares, utilizando criterios específicos para la agrupación y reducción de datos sin comprometer la integridad de la información procesada.

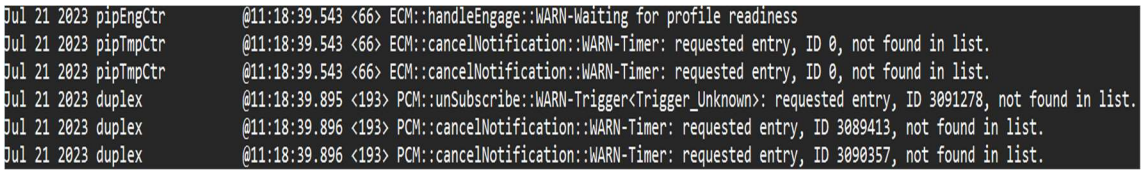
El formato ajustado para Drain se define de la siguiente manera, con detalles adicionales proporcionados en la Tabla 3 y visualizados en la Figura 6:

log_format = '<Date> <Task> @<Time> <Level> <Component>:<Content>'

Tabla 3. Estructura y descripción del mensaje para ser procesado por Drain.

Estructura	Descripcion
Date:	formato de fecha, esta línea se creó en base al nombre del log que contiene la fecha el cual fue mes, día, año
Task	Tarea o rutina que realizar el dispositivo
Time:	la información fue capturada con el patrón de estar acompañada del símbolo @, el formato de esta línea es: @ hora:minuto:segundos:milisengundos.
Level:	símbolo que se ajustó para el formato y lo contiene las líneas seleccionadas que fue <0> (ver figura 5).
Component:	es dispositivo que está haciendo la tarea o capturando la información.
Content:	Información de la telemetria que se está realizando, como información de los sensores o estado de la rutina que se está desarrollando.

Figura 6. Muestra del Formato de los registros antes de ser procesados por Drain.



Preprocesamiento mensajes de sistema.

Los mensajes del sistema, extraídos del archivo en formato `.xlsm`(formato de hoja de cálculo XML se basa en archivo Office Open XML) de la fábrica, fueron transformados a minúsculas y almacenados en un archivo de texto. Este formato estandarizado facilita el procesamiento posterior con FastText, optimizando la interpretación de los datos.

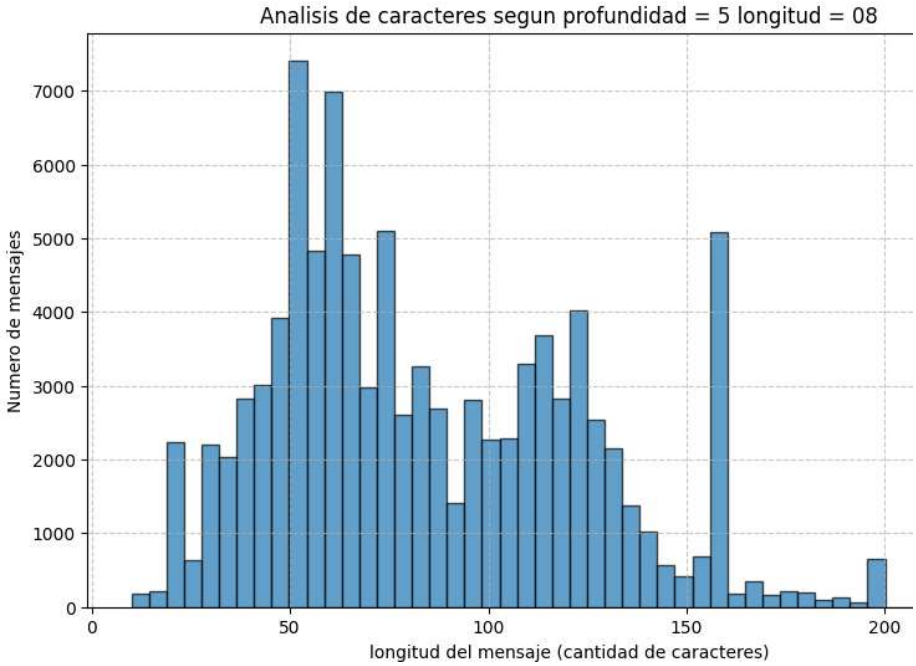
Modelamiento.

En esta sección, se detalla la fase de modelado, que constituye el núcleo del modelo AIRE. El análisis se llevó a cabo utilizando el analizador sintáctico Drain sobre los datos del registro principal del archivo ZIP. Para optimizar el rendimiento del analizador, se realizaron pruebas variando los hiper-parámetros de similitud y profundidad.

Aplicación parser Drain.

Inicialmente, se estableció un valor de similitud de 8. Este valor fue seleccionado tras observar que permitía una construcción más independiente de las líneas de datos, lo cual es crucial para mejorar la claridad y la precisión del análisis sintáctico. En cuanto a la profundidad, se optó por un valor de 5, buscando un equilibrio entre la capacidad de procesar datos extensos y la minimización de la carga computacional, sin comprometer la calidad de la información procesada.

Figura 7. Histograma número de caracteres vs longitud del mensaje, obtenidos por Drain.



Para validar la eficacia de los ajustes de los hiper-parámetros, se realizó un análisis mediante un histograma, tal como se muestra en la figura 7. Este histograma revela la distribución de las longitudes de los mensajes procesados por el analizador sintáctico, destacando dos picos significativos en longitudes de aproximadamente 30-40 caracteres y 150 caracteres. Estos picos sugieren que el sistema procesa comúnmente dos tipos predominantes de mensajes: respuestas rápidas y breves, y entradas más largas y descriptivas, indicando diferentes formas de comunicación o tipos de entradas de datos dentro del sistema, adicionalmente se realizó un análisis detallado que se presenta a continuación:

Picos y Valles: Los picos en las longitudes de 30-40 y 150 caracteres reflejan las longitudes más comunes de los mensajes, lo que puede indicar patrones de uso del sistema o restricciones en la entrada de datos. Los valles entre 60-70 y 120-130 caracteres sugieren transiciones entre diferentes estilos de comunicación.

Distribución Asimétrica: La distribución asimétrica muestra una mayor concentración de mensajes cortos, indicando que son más frecuentes que los mensajes largos.

Rango de Longitudes: La mayoría de los mensajes no superan los 1200 caracteres, con una longitud máxima de hasta 1400 caracteres. Esta concentración en longitudes de hasta 150 caracteres refleja posiblemente una limitación o preferencia operativa para mensajes más breves dentro del contexto del analizador.

Este análisis subraya la adecuación de los hiper-parámetros seleccionados. Con esta configuración, el analizador sintáctico Drain logró procesar un total de 9,772 líneas de un conjunto inicial de 186,591 líneas. Los resultados del histograma, por tanto, validan la elección de los hiper-parámetros de profundidad y similitud seleccionados, facilitando un procesamiento eficiente tanto del volumen de datos como de los recursos computacionales.

Preprocesamiento flujo plantilla Drain.

Se realizó la extracción, en la columna de la plantilla de mensajes generada por el archivo CSV de Drain, se aplica de nuevo la transformación a minúsculas esto para asegurarse que Drain no generara cambios a mayúsculas en el proceso, se elimina mensajes de información duplicada, ya que no se tiene en cuenta en el modelo frecuencia de mensajes, posteriormente, se transforma de extensión CSV(valores separado por comas) a un archivo de extensión de texto Txt, aceptado por el módulo FastText.

Preprocesamiento flujo mensajes de sistema.

Los mensajes del sistema fueron extraídos del archivo de fábrica, estos fueron transformados a minúsculas y guardados en un archivo txt para el procesamiento de FastText.

FastText.

En el contexto del modelo AIRE, se emplean representaciones vectoriales de palabras para el análisis de datos textuales. Estas representaciones, conocidas como matrices vectoriales, asignan a cada palabra o n-grama un vector numérico en un espacio de 300 dimensiones. El número 300 se refiere al número de atributos que cada vector utiliza para representar características lingüísticas de la palabra, permitiendo así una representación rica y detallada del significado y el contexto lingüístico. Este enfoque facilita el procesamiento y la comprensión matemática del texto por parte de sistemas informáticos.

Para este propósito, se ha seleccionado el modelo pre-entrenado cc.en.300.bin de FastText, perteneciente a la colección 'Common Crawl'. Este modelo es adecuado ya que los mensajes de la fábrica y los recolectados por las máquinas de impresión están en inglés. Common Crawl ha sido previamente entrenado en extensos conjuntos de datos web, lo que le permite capturar eficazmente la esencia del idioma inglés y generar vectores para palabras completas y sub-palabras. Utilizando este modelo, se transforman los textos de la plantilla de Drain y los mensajes emitidos por la fábrica en representaciones vectoriales numéricas.

La reconstrucción de las representaciones vectoriales de los mensajes, tanto de la fábrica como de la impresora digital, es fundamental para analizar, comprender los patrones de comunicación y las operaciones dentro de estos entornos. Este proceso involucra un emparejamiento preciso entre el valor vectorial y la subpalabra, tanto en los registros procesados por el analizador sintáctico Drain como en los mensajes emitidos por la fábrica.

Para observar el comportamiento de los dos grupos de mensajes independientemente, se realiza transformación de datos utilizando el método de k-means. Esta técnica identifica centroides y asigna puntos de datos a estos grupos basados en la similitud de sus características vectoriales. Las dimensiones de estas matrices se detallan en la Tabla 4.

Tabla 4. Tamaño de matrices después de reconstrucción vectorial.

Matriz	Tamaño
Plantilla Drain	9772 x 300
Mensajes de Fabrica	4262 x 300

Esta metodología, ilustrada en las Figuras 8 y 9, facilita la visualización y análisis de patrones en los datos. La técnica de k-means permite la aproximación al comportamiento de los datos estáticos, como son los mensajes de la fábrica, y los mensajes de los registros de la máquina que son aleatorios, permitiendo la identificación de agrupaciones. Sin embargo, es importante destacar que, aunque útil para entender el comportamiento de estos datos, la agrupación mediante k-means no constituye el objetivo final del modelo AIRE. En su lugar, el proyecto AIRE se centra en el uso de la similitud cosenoidal para realizar el agrupamiento final, que es fundamental para el análisis y la monitorización efectiva de los eventos registrados por las impresoras, optimizando así la respuesta y el mantenimiento de los sistemas en contextos industriales.

A continuación, el análisis de los agrupamientos por medio de K-means:

La Figura 8 muestra la distribución de los agrupamientos con los centroides de las plantillas de registro del archivo de la máquina. En esta, se observan varios grupos claramente definidos, cada uno representado por un color diferente. Esto indica que el algoritmo ha podido diferenciar entre diferentes tipos de mensajes o situaciones registradas en las plantillas. Dicha diferenciación es crucial, ya que permite identificar tendencias en el comportamiento de la prensa.

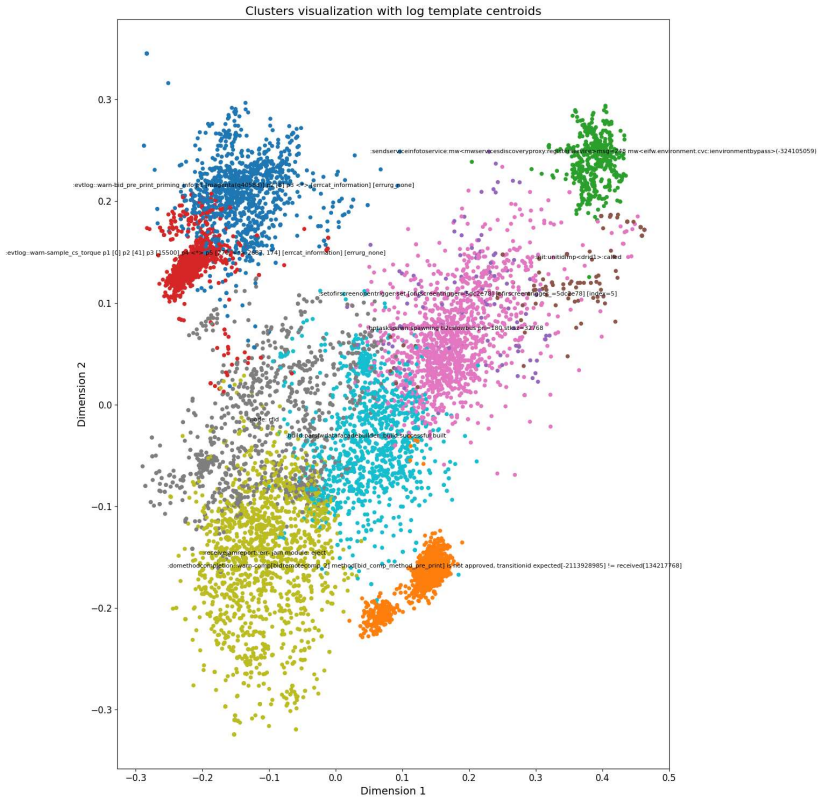
Características Notables:

Separación de agrupamiento: tiene un espacio claro entre ellos, lo que sugiere una buena diferenciación y agrupación de los datos.

Distribución de los Centroides: Los centroides están ubicados cerca del centro de sus respectivos agrupamientos, lo cual es típico en un buen resultado de k-means.

Densidad Variable: Algunos grupos parecen más densos que otros, lo que podría indicar una frecuencia más alta de ciertos tipos de mensajes.

Figura 8. Representación gráfica de los grupos por Kmeans, de los vectores de los registros de la máquina.



La gráfica 9, aunque similar a la primera, muestra centroides basados en los mensajes de fábrica. Aunque la diferenciación entre los grupos sigue siendo clara, se observa un cambio en la ubicación y composición de estos comparado con la gráfica anterior. Estos cambios son consecuencia del filtrado de mensajes que la fábrica ha realizado, utilizando diversas versiones del software instalado en la máquina de impresión a lo largo del tiempo.

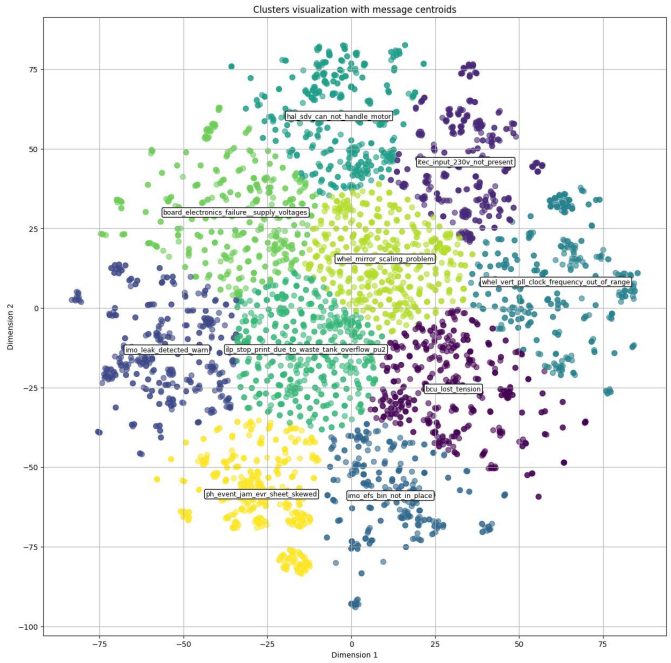
Características Notables:

Centroides de Mensajes: Los centroides son los puntos que están destacados dentro de cada grupo.

Agrupación: Aunque los conjuntos están bien definidos, la proximidad entre algunos de ellos es mayor en comparación con la gráfica 7, lo indica similitudes más cercanas entre ciertos tipos de mensajes.

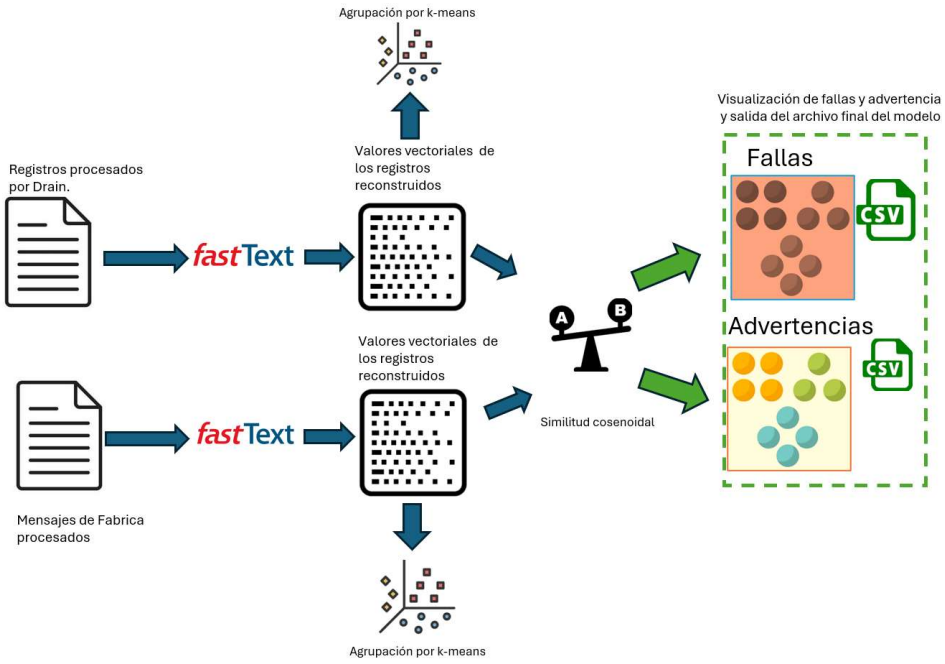
Distribución espacial: Reflejan la diversidad y/o similitud de los mensajes dentro de cada grupo.

Figura 9. Representación gráfica de los cluster Kmeans, de los vectores de los mensajes del sistema compartidos por la fábrica.



Para observar el flujo interno del modelo se presenta en la gráfica 10, la representación de las etapas desde que se reciben los registros de la maquina procesados por Drain y los mensajes de la máquina y su debido procesamiento por fastText, también se observa etapas que se explicaran más adelante.

Figura 10. Flujo de información FastText de modelo Aire



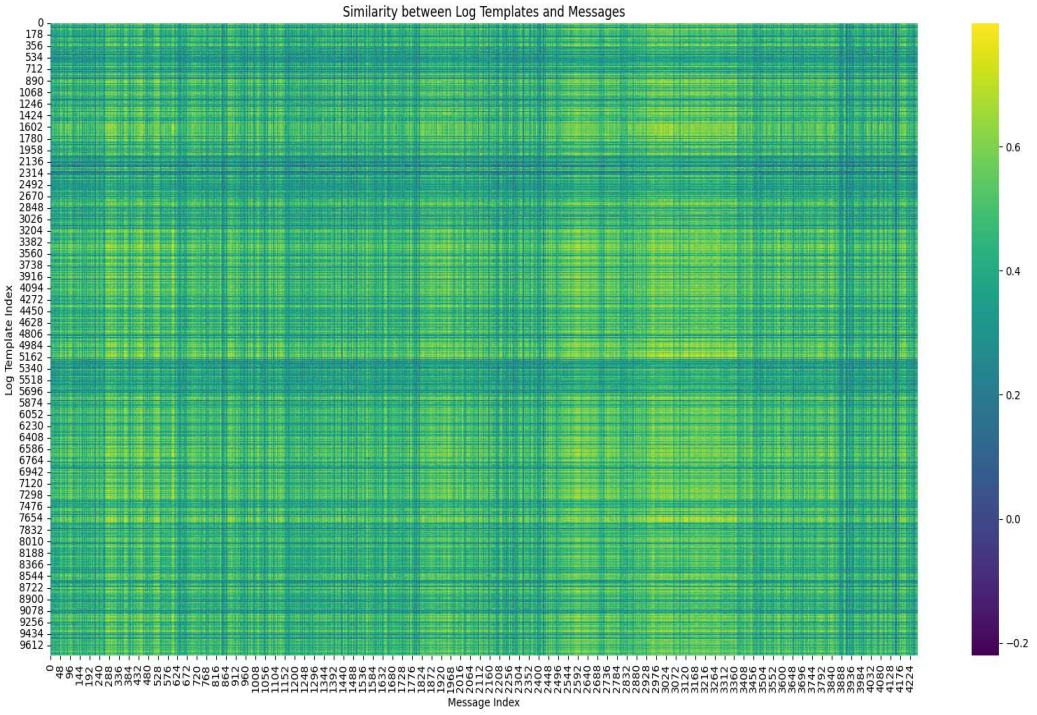
Agrupamiento por similitud cosenoidal.

El principal objetivo de esta sección es mejorar la eficiencia en la monitorización y el mantenimiento automatizado de sistemas en contextos industriales, permitiendo una respuesta más rápida y efectiva a los eventos registrados. Para alcanzar este objetivo, se ha empleado el modelo FastText para determinar los valores vectoriales de los mensajes de la fábrica y los registros de la máquina, facilitando así la reconstrucción de las líneas de texto y la ejecución de un análisis de similitud cosenoidal entre estos mensajes de fábrica versus registros de la maquina procesados. Este análisis vectorial es crucial porque permite agrupar mensajes y registros de manera eficiente basándose en su similitud cosenoidal, utilizando una escala de 0 a 1 donde 1 indica que los vectores son completamente iguales.

Un umbral de similitud de 0.65 se estableció para identificar y descartar aquellos mensajes con baja o nula similitud, garantizando que solo se consideren para el análisis los mensajes suficientemente similares. Los mensajes de la fábrica y los registros de la máquina, una vez identificados y reconstruidos vectorialmente, se

agruparon utilizando técnicas de agrupamiento. Aunque los registros en la matriz de Drain-FastText (registros de la maquina) contienen información adicional crucial para el análisis por parte de expertos, esto no impide la comparación efectiva con los mensajes de la fábrica.

Figura 11. Mapa de calor entre las dos matrices vectoriales mensajes de log (Drain) y los mensajes del sistema.



En la gráfica número 11 que representa el mapa de calor con ejes descritos como registros de la maquina (índices en el eje vertical) y mensajes del sistema (índices en el eje horizontal). Esta visualización es útil para identificar rápidamente las similitudes entre los registros de texto a través de sus representaciones vectoriales. La escala de color varía de -0.2 a 0.6, con tonos más fríos (morados) que indican baja similitud y tonos más cálidos (amarillos) que señalan alta similitud. Predominan los tonos verdes y amarillos, indicando una similitud moderada a alta entre muchas

combinaciones de plantillas y mensajes. No se observan áreas predominantemente moradas, lo cual evitaría una falta total de similitud.

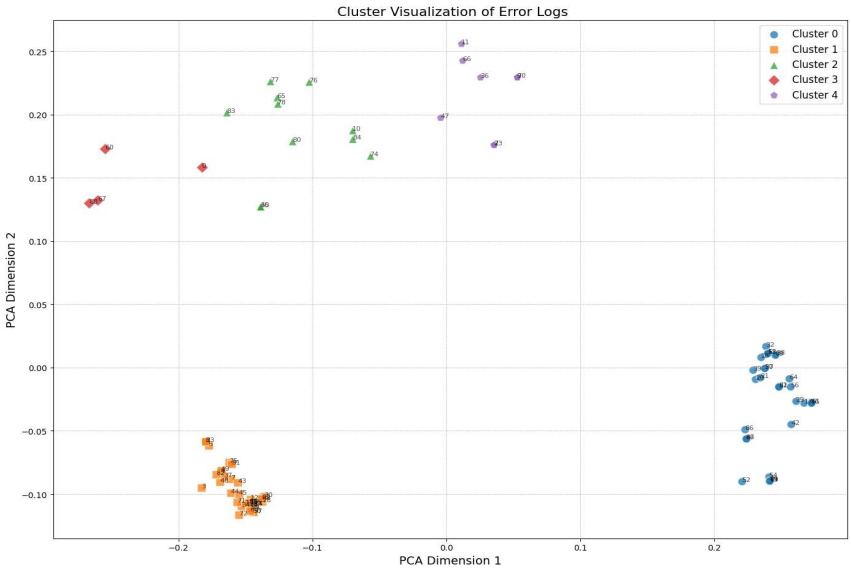
Almacenamiento y reporte de resultados.

Los resultados de los análisis de grupos de mensajes se almacenan y reportan en un formato accesible, con cada grupo detalladamente descrito en archivos CSV que el modelo proporciona como salida final. Esta organización de datos facilita no solo el acceso y la revisión por parte de los técnicos y el personal de soporte técnico, sino que también proporciona una base sólida para análisis detallados y la implementación de acciones correctivas. La transparencia y accesibilidad de estos datos son fundamentales para una gestión efectiva y oportuna de las operaciones de la impresora.

Al comparar los vectores de La técnica de similitud cosenoidal ha sido fundamental en el agrupamiento de los mensajes de error y advertencia, basándose en su similitud textual. texto de los registros, el modelo facilita la identificación y categorización de anomalías en distintos subsistemas de la impresora, tales como el manejo del papel y la generación de imágenes. Esta metodología no solo agrupa mensajes similares, sino que también revela sus contextos específicos, ofreciendo así una comprensión más profunda de las condiciones bajo las cuales se generan estos registros.

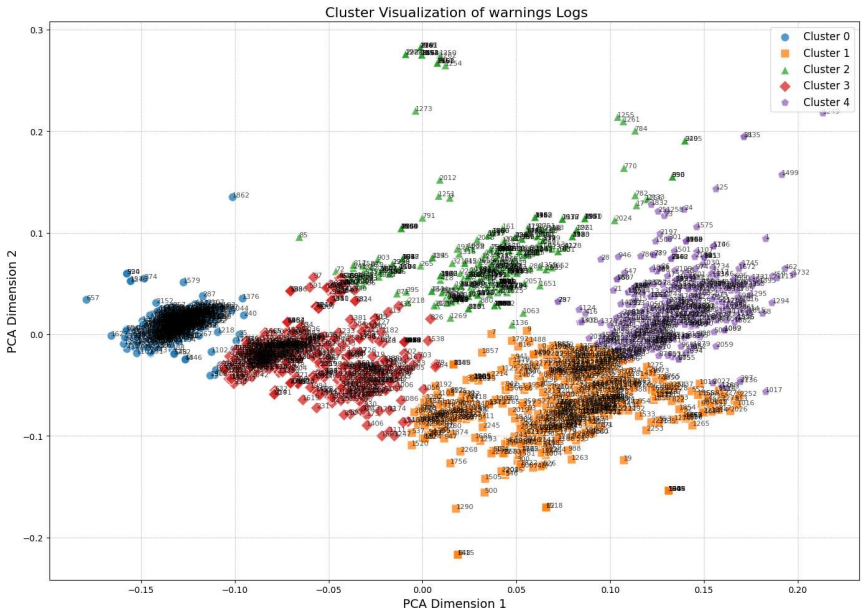
En la gráfica 12, donde se analizan las fallas de la máquina, la clara separación entre los cinco grupos resalta la eficacia del modelo en categorizar variados tipos de errores. Notablemente, el grupo más aislado, el grupo 4, sugiere un tipo de error que es significativamente diferente de los otros, posiblemente en términos de gravedad o naturaleza del fallo. Esta diferenciación clara es crucial para comprender las posibles fallas en el sistema, permitiendo la implementación de medidas preventivas más dirigidas y una mejor gestión del mantenimiento.

Figura 12. Agrupación realizada por Kmeans de los errores de la prensa.



Respecto a la gráfica 13, que aborda los grupos de advertencias, aunque estos grupos están densamente poblados y presentan una ligera superposición, cada uno representa diferentes categorías de advertencias que son vitales para la operatividad de las impresoras. La variabilidad observada en el eje de la dimensión 1 del PCA indica diferencias en la gravedad y el impacto de las advertencias. Este análisis detallado ayuda a priorizar las respuestas a las advertencias según su impacto potencial en la funcionalidad de la impresora. Es importante mencionar que una advertencia sobresale por ser un evento que supera el punto de referencia establecido, aunque sin llegar a exceder los umbrales máximos.

Figura 13. Agrupación realizado por Kmeans de las advertencias de la prensa.



Las técnicas de análisis como PCA y K-Means han sido eficaces para visualizar y comprender la dinámica de estos eventos, mientras que la aplicación de la similitud cosenoidal ha enriquecido el proceso de agrupación, permitiendo discernir complejas relaciones textuales entre los registros tanto de advertencias como de fallas. Este análisis estructurado no solo proporciona una comprensión detallada de los eventos registrados, sino que también resalta la importancia de las técnicas avanzadas de minería de datos en el mantenimiento y la operación de equipos industriales.

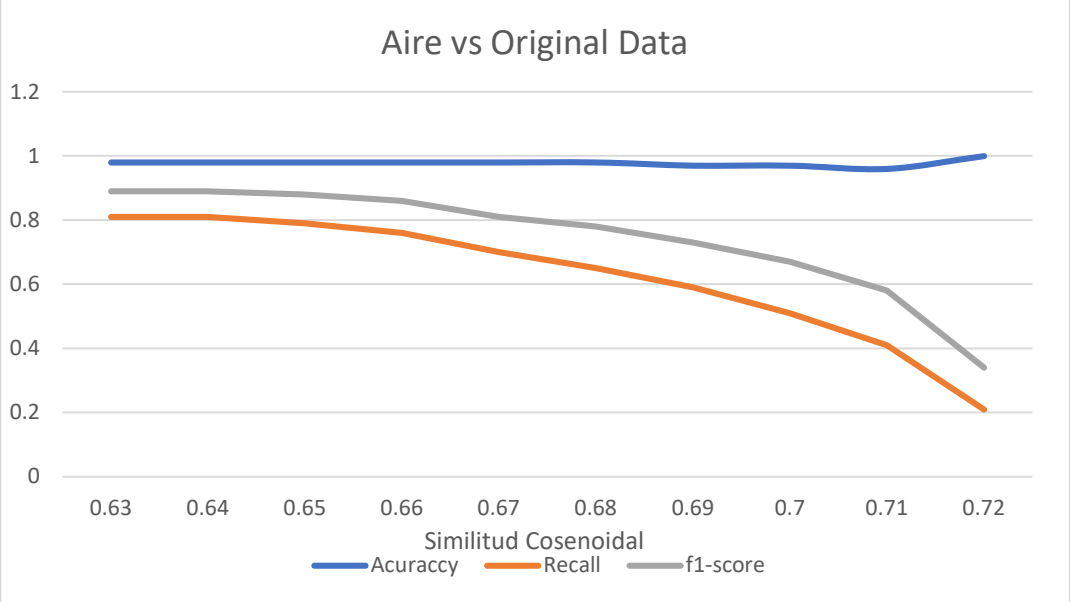
Resultados.

El modelo AIRE se evaluó comparando la información generada por el mismo con los datos originales, es decir, se clasificaron manualmente los errores en los archivos de registro antes de procesarlos con el modelo. Posteriormente, se analizó la salida del modelo utilizando métricas como accuracy, recall y F1-score.

Para esta evaluación, se capturó información durante tres días consecutivos, generando más de 185,000 líneas en la prensa HP Indigo 10000. (Véase Anexo 2 para un modelo

evaluado con datos de un día y de siete días de trabajo de la prensa, para comparar los resultados).

Figura 14. Evaluación del modelo por medio de accuracy, recall y f1-score, con un total de 186100 de la prensa HP índigo 10000.



Con respecto a la Figura 14, la evaluación por similitud cosenoidal osciló entre 0.63 y 0.68, con un accuracy cercano a uno. Este alto valor de accuracy se debe a que los mensajes de la fábrica, que están incluidos en el modelo, son similares a los del modelo FastText-Drain. Además, el modelo Drain-FastText no solo incluye mensajes directos de la fábrica, sino que también integra información adicional del proceso que se lleva a cabo, lo cual enriquece la contextualización de los datos analizados. Sin embargo, al incrementar el umbral de similitud cosenoidal, se empiezan a excluir de la salida del modelo aquellos mensajes que no cumplen con el factor de similitud, reduciendo la inclusividad del modelo y afectando la precisión global de los resultados.

Además, el algoritmo Drain, que ha demostrado una precisión del 84% en la clasificación, puede ser un factor determinante en la caída de la clasificación de AIRE cuando se incrementa el umbral de similitud cosenoidal. Esto se debe a que Drain tiende a fusionar varios errores en una misma línea de árbol dentro de su proceso de clasificación. Esta fusión puede llevar a una generalización excesiva, lo que resulta en un modelo menos capaz de diferenciar entre errores únicos cuando la similitud se incrementa, afectando negativamente la funcionalidad de AIRE. Esta generalización puede ser una causa directa del decaimiento observado en la clasificación por parte de AIRE, especialmente a medida que el modelo se vuelve más restrictivo con un umbral de similitud más alto.

El modelo enfrenta limitaciones al aumentar la rigidez del umbral de similitud cosenoidal. Es evidente que hay una disminución del recall a partir de un umbral de aproximadamente 0.65, momento en el cual el modelo comienza a perder la capacidad de clasificar correctamente los datos y puede fallar en distinguir entre errores y no errores, como lo demuestra la caída del F1-score que indica que el algoritmo comienza a perder eficacia y efectividad.

Herramientas utilizadas.

En el desarrollo del modelo se utilizaron herramientas clave como Python 3.10, un lenguaje de programación versátil y de alto nivel, adecuado para diversas aplicaciones incluyendo ciencia de datos y desarrollo web. Pandas fue crucial para la manipulación y análisis de datos, facilitando la extracción de información y el procesamiento de resultados en CSV, así como la creación de archivos CSV para los clústeres del modelo. Numpy se destacó en el manejo de matrices, especialmente las generadas por FastText. La biblioteca 're' fue importante para la manipulación de texto en los logs, mientras que Matplotlib ayudó en la visualización de datos. LogparserDrain procesó mensajes de log en plantillas CSV, y 'sys' y Zipfile manejaron la integración y gestión de archivos, incluyendo los extraídos de la página de HP(Hewlett Packard) en formato Zip.

FastText se utilizó para el análisis de datos textuales y aprendizaje automático, optimizando la representación de palabras y oraciones. Sklearn (Scikit-learn) se aplicó en varias áreas, utilizando sub-librerías como Sklearn.metric.pairwise para similitud coseno, Sklearn.decomposition para PCA, y Sklearn.cluster con Kmeans para la agrupación de datos. Estas herramientas fueron esenciales para el análisis y segmentación de los datos. Visual Studio Code 1.84.2, un entorno de desarrollo integrado de Microsoft proporcionó un soporte robusto para Python, facilitando la escritura, depuración y gestión del código en el proyecto.

Conclusiones.

Este estudio ha demostrado la efectividad de aplicar técnicas avanzadas de procesamiento de lenguaje natural (NLP) para el tratamiento y análisis de archivos de registros semi-estructurados, específicamente en el contexto de las prensas HP Indigo. A través del desarrollo e implementación del modelo AIRE, se logrados objetivos cruciales que reflejan avances significativos en el diagnóstico de anomalías.

Alcance y Eficacia del Modelo AIRE

La integración de modelos NLP y la selección precisa de herramientas como Drain y FastText han permitido la clasificación eficaz y precisa de textos semi-estructurados derivados de los registros de las máquinas. Esta capacidad de procesamiento y clasificación no solo facilita el análisis de datos, sino que también mejora la identificación de anomalías y el comportamiento del sistema, cumpliendo así con el objetivo general y el primer objetivo específico de recolectar y aplicar el modelo de NLP más adecuado.

Adaptabilidad y Flexibilidad del Modelo

El modelo AIRE ha demostrado ser notablemente adaptable y actualizable, característica esencial ante cualquier cambio o actualización por parte del fabricante. Esta adaptabilidad, destacada el segundo objetivo específico, asegura que el modelo pueda evolucionar continuamente para satisfacer las necesidades de diagnóstico de fallas, reflejando la naturaleza dinámica de las prensas HP Indigo y sus aplicaciones.

Avances en el Diagnóstico y Mantenimiento

Se ha logrado un avance significativo en proporcionar un método que permite no solo el análisis de anomalías sino también la propuesta de planes de mantenimiento

correctivo, preventivo y predictivo. Este enfoque se alinea con el tercer y cuarto objetivo específico, destacando la capacidad del modelo para incrementar el análisis de anomalías y facilitar la toma de decisiones basada en datos concretos y contextualizados. La flexibilidad en la entrada de datos ha permitido analizar diferentes rangos de uso diario de las prensas, ofreciendo insights valiosos para la optimización de las operaciones y el mantenimiento.

Viabilidad del Modelo

La viabilidad del modelo AIRE para su implementación se confirma a través de la eficiencia computacional alcanzada en las etapas de preprocesamiento. La utilización de Drain para filtrar mensajes de log mantiene la relevancia del proceso sin sacrificar información crítica. Simultáneamente, FastText minimiza la pérdida de vocabulario, optimizando el rendimiento y la precisión del modelo. Estos hallazgos subrayan la contribución del modelo a reducir los requerimientos computacionales mientras se mantiene la eficacia en el diagnóstico de fallas.

En conclusión, el modelo AIRE representa un avance significativo en el campo del análisis de logs y diagnóstico de anomalías en sistemas complejos. Su desarrollo, fundamentado en objetivos claros y precisos, ha culminado en una herramienta robusta, adaptable y eficaz que no solo cumple con las expectativas planteadas, sino que también establece un nuevo estándar para futuras investigaciones y aplicaciones prácticas en la industria de la impresión digital y más allá.

Trabajos futuros.

Los resultados alcanzados por el modelo AIRE en el análisis de registros para la detección de advertencias y fallas en sistemas de prensas HP Índigo abren diversas posibilidades para la expansión de esta investigación. A continuación, se muestran campos clave para el desarrollo futuro del modelo.

Integración en Sistemas Operativos en Tiempo Real: La implementación en línea del modelo AIRE directamente en los sistemas operativos, ofrece un avance significativo hacia el monitoreo continuo y la gestión proactiva de anomalías. Esta adaptación permitiría a los usuarios evaluar y responder a los informes de estado sin necesidad de herramientas externas, promoviendo un ambiente de mantenimiento preventivo y correctivo más ágil y eficiente.

Expansión a Diversos Modelos de Prensas HP Índigo: Dada la versatilidad demostrada por AIRE, explorar su aplicabilidad a una variedad más amplia de modelos de prensas HP índigo y potencialmente a otras series se presenta como un paso lógico. Adaptar el modelo para satisfacer las especificaciones y requerimientos únicos de cada tipo de prensa podría mejorar significativamente la precisión en la detección de anomalías, beneficiando una gama más extensa de aplicaciones industriales, siempre y cuando se solicite la autorización a la compañía para la utilización de los datos.

Optimización de Parámetros y Ajustes Configurables: Profundizar en la investigación sobre la influencia de los hiperparámetros de Drain y los parámetros de similitud cosenoidal en la efectividad del modelo para diversas condiciones de operación representa una oportunidad para afinar aún más su rendimiento. Ajustar estos parámetros podría llevar a mejoras en la precisión y eficiencia del modelo, optimizando los recursos computacionales y adaptando el sistema a necesidades específicas de diagnóstico.

Desarrollo de Interfaz de Usuario Avanzada para Análisis y Diagnóstico: El diseño de una interfaz de usuario intuitiva y herramientas de visualización avanzadas facilitaría la interpretación de los resultados del análisis de logs por tanto en esta interfaz podría transformar la forma en que se manejan las operaciones de mantenimiento, elevando la eficiencia operativa y la toma de decisiones basada en datos.

Aplicación para el Análisis Predictivo de Fallas: La capacidad del modelo AIRE de procesar y analizar datos de logs en diferentes rangos temporales sugiere un potencial sin explotar para la predicción de fallas. Desarrollar y validar algoritmos que puedan prever anomalías antes de que ocurran, basándose en patrones detectados en los datos históricos, reduciendo el tiempo de inactividad y aumentando la productividad.

Evaluación de Portabilidad y Escalabilidad del Modelo: Finalmente, evaluar la portabilidad del modelo AIRE para su integración en diferentes entornos operativos y su escalabilidad para manejar volúmenes crecientes de datos son aspectos cruciales para garantizar su aplicabilidad a largo plazo. Esta evaluación ayudaría a identificar los desafíos y oportunidades para su implementación en una variedad de contextos industriales.

Negociaciones para Acceso a Datos: Se recomienda que futuros investigadores interesados en continuar o replicar este estudio establezcan inicialmente acuerdos de colaboración o negocien el acceso a los datos con Hewlett Packard y la Universidad de la Sabana. Es esencial que estos acuerdos cumplan con las normas de confidencialidad y las restricciones legales establecidas tanto por la compañía como por la institución académica. Esto permitirá acceder a la plataforma necesaria para realizar las réplicas requeridas del Modelo Aire de manera ética y conforme a la ley.

Referencias bibliográficas.

1. Ryciak, P., Wasielewska, K., & Janicki, A. (2022). Anomaly detection in log files using selected natural language processing methods. *Appl. Sci.*, 12(1), 5089. https://doi.org/10.3390/app12105089.
2. Gillespie, M., & Givre, C. (2021). *Understanding log analytics at scale* (2nd ed.). O'Reilly.
3. Abdelakfi, M., Mbarek, N., & Bouzguenda, L. (2021). Mining organizational structures from email logs: An NLP based approach. In *Proceedings of the 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*. Elsevier. CC BY-NC-ND license.
4. Matteo Boffa, Luca Vassio, Marco Mellia, Idilio Drago, Giulia Milan, Zied Ben Houidi, Dario Rossi. 2022. On Using Pretext Tasks to Learn Representations from Network Logs. In Native Network Intelligence (NativeNI '22), December 9, 2022, Roma, Italy. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3565009.356952
5. Ramos-Gutiérrez, B., Varela-Vaca, Á. J., Ortega, F. J., Gómez-López, M. T., & Wynn, M. A NLP-oriented methodology to enhance event log quality. *IDEA & ITALICA Research Groups, Universidad de Sevilla*. http://www.idea.us.es
6. Bertero, C., Roy, M., Sauvanaud, C., & Tredan, G. (2017). Experience Report: Log Mining Using Natural Language Processing and Application to Anomaly Detection. In *Proceedings - International Symposium on Software Reliability Engineering, ISSRE*, 351-360
7. Wang, J., Tang, Y., He, S., Zhao, C., Sharma, P., Alfarraj, O., & Tolba, A. (2020). LogEvent2vec: LogEvent-to-vector based anomaly detection for large-scale logs in internet of things. *Sensors (Switzerland)*, 20(9).
8. Wang, W., Wang, J., Peng, X., Yang, Y., Xiao, C., Yang, S., Wang, M., Wang, L., Li, L., & Chang, X. (2023). Exploring best-matched embedding

model and classifier for charging-pile fault diagnosis. *Cybersecurity, 6*(7). https://doi.org/10.1186/s42400-023-00138-z

9. Sehwani, N. S. (2022). No Features Needed: Using BPE Sequence Embeddings for Web Log Anomaly Detection. In *Proceedings of the 2022 ACM International Workshop on Security and Privacy Analytics (IWSPA '22)*, April 24–27, 2022, Baltimore, MD, USA ACM, New York, NY, USA. https://doi.org/10.1145/3510548.3519375
10. Meghanani, A., Anoop, C.S., & Ramakrishnan, A.G. (2021). Recognition of Alzheimer's dementia from the transcriptions of spontaneous speech using fastText and CNN models. *Frontiers in Computer Science, 3*, 624558. https://doi.org/10.3389/fcomp.2021.624558
11. Khomsah, S., Ramadhani, R. D., & Wijayanto, S. (2022). The Accuracy Comparison Between Word2Vec and FastText on Sentiment Analysis of Hotel Reviews. *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6(3), 352-358. http://jurnal.iaii.or.id
12. Novotný, V., Ayetiran, E. F., Baˇcovský, D., Lupták, D., Štefánik, M., & Sojka, P. (2021). One Size Does Not Fit All: Finding the Optimal Subword Sizes for FastText Models across Languages. In *Proceedings of Recent Advances in Natural Language Processing* (pp. 1068-1074). Masaryk University. https://doi.org/10.26615/978-954-452-072-4_120
13. Chen, S., & Liao, H. (2022). BERT-Log: Anomaly Detection for System Logs Based on Pre-trained Language Model. *Applied Artificial Intelligence*, 36(1), e2145642. https://doi.org/10.1080/08839514.2022.2145642.
14. Huang, S., Liu, Y., Fung, C., He, R., Zhao, Y., Yang, H., & Luan, Z. (2020). HitAnomaly: Hierarchical Transformers for Anomaly Detection in System Log. *IEEE Transactions on Network and Service Management*, 17(4), 2064.
15. Wang, J., Zhao, C., He, S., Gu, Y., Alfarraj, O., & Abugabah, A. (2021). LogUAD: Log Unsupervised Anomaly Detection Based on Word2Vec.

Computer Systems Science & Engineering. https://doi.org/10.32604/csse.2022.022365

16. Xiao, T., Quan, Z., Wang, Z.-J., Zhao, K., & Liao, X. (2020). LPV: A Log Parser Based on Vectorization for Offline and Online Log Parsing. *2020 IEEE International Conference on Data Mining (ICDM)*. College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; College of Computer Science, Chongqing University, Chongqing, China; School of Computer Science, University of Auckland, Auckland, New Zealand; College of Computer Science and Technology, National University of Defense Technology, Changsha, China.
17. Wang, Z., Tian, J., Fang, H., Chen, L., & Qin, J. (2021). LightLog: A lightweight temporal convolutional network for log anomaly detection on the edge. *College of Information Engineering, Dalian University, Dalian, China; School of Computer Science, Loughborough University, Loughborough, UK; School of Computing, Ulster University, Belfast, UK; School of Software Engineering, Dalian University, Dalian, China*. https://doi.org/10.1016/j.comnet.2021.108616
18. Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. *School of Computing, University of Utah*. http://dx.doi.org/10.1145/3133956.3134015
19. Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., Sun, P., & Zhou, R. (2019). LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. Tsinghua University; University of Toronto; Nankai University; Huawei; Beijing National Research Center for Information Science and Technology (BNRist)
20. Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y., Xie, C., Yang, X., Cheng, Q., Li, Z., Chen, J., He, X., Yao, R., Lou, J.-G., Chintalapati, M.,

Shen, F., & Zhang, D.-M. (2019). Robust log-based anomaly detection on unstable log data. In *Proceedings of the 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '19)*, August 26–30, 2019, Tallinn, Estonia. ACM, New York, NY, USA. https://doi.org/10.1145/3338906.3338931

21. Hamooni, H., Debnath, B., Xu, J., Zhang, H., Jiang, G., & Mueen, A. (2016). LogMine: Fast pattern recognition for log analytics. In *Proceedings of the ACM Conference* (pp. 1-10). ACM. https://dx.doi.org/10.1145/2983323.2983358
22. Qi, J., Huang, S., Luan, Z., Fung, C., Yang, H., & Qian, D. (Year of publication). LogGPT: Exploring ChatGPT for log-based anomaly detection.
23. Huang, S., Liu, Y., Fung, C., Qi, J., Yang, H., & Luan, Z. (2023, March). LogQA: Question answering in unstructured logs. *ACM Computing Surveys*.
24. Lv, D., Luktarhan, N., & Chen, Y. (2021). ConAnomaly: Content-Based Anomaly Detection for System Logs. *Sensors, 21*(6125). https://doi.org/10.3390/s21186125
25. Zhang, C., Peng, X., Sha, C., Zhang, K., Fu, Z., Wu, X., Lin, Q., & Zhang, D. (2022). DeepTraLog: Trace-Log Combined Microservice Anomaly Detection through Graph-based Deep Learning. In *Proceedings of the 44th International Conference on Software Engineering (ICSE '22)* (pp. 1-12). ACM. https://doi.org/10.1145/3510003.3510180
26. Zhang, S., Meng, W., Bu, J., Yang, S., Liu, Y., Pei, D., Xu, J., Chen, Y., Dong, H., Qu, X., & Song, L. (2017). Syslog processing for switch failure diagnosis and prediction in datacenter networks. *Proceedings of the IEEE Conference*. IEEE. https://doi.org/978-1-5386-2704-4
27. Li, X., Chen, P., Jing, L., He, Z., & Yu, G. (2021). SwissLog: Robust anomaly detection and localization for interleaved unstructured logs. *Journal of LaTeX Class Files, 14*(8).

28. Pi, A., Chen, W., Zeller, W., & Zhou, X. (2019). It can understand the logs, literally. *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. https://doi.org/10.1109/IPDPSW.2019.00084
29. He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. *Proceedings of the IEEE International Conference on Web Services (ICWS)*. https://doi.org/10.1109/ICWS.2017.13
30. Du, M., & Li, F. (2017). Spell: Streaming parsing of system event logs. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'17)*. IEEE.
31. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.
32. Wu, H. C., Luk, R. W. P., Wong, K. F., & Kwok, K. L. (Year). Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems, 26*(3), 1–37. https://doi.org/10.1145/1361684.1361686
33. Devlin, J., M. W. Chang, K. Lee, and K. Toutanova. 2018. BERT: Pre-training of deep bidirec-tional transformers for language understanding. arXiv preprin arXiv:1810.04805, Oct 11
34. M. Gallé," Investigating the Effectiveness of BPE: The Power of Shorter Sequences", Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.
35. J. Shlens, "A Tutorial on Principal Component Analysis", https://arxiv.org/pdf/1404.1100.pdf, April 7th, 2014.
36. B. Trevett, D. Reay and N. K. Taylor, "The Effectiveness of Pre-Trained CodeEmbeddings", 16th International Conference on Data Science (ICDATA'20), Jul 2020.

37. Hagiwara, M. (2021). Tokenization. In *Real-World Natural Language Processing*. Manning Publications.
38. Dirección de Investigación Criminal e Interpol. (2023). *Balance de ciberseguridad 2022*. Policía Nacional de Colombia. https://caivirtual.policia.gov.co/sites/default/files/observatorio/Balance%20anual%202022.pdf
39. CrowdStrike. (2024). *Global threat report 2024*. CrowdStrike. https://go.crowdstrike.com/rs/281-OBQ266/images/GlobalThreatReport2024.pdf
40. Hagiwara, M. (2021). Natural Language Processing. Manning Publications.
41. Zhang, T., Qiu, H., Castellano, G., Rifai, M., Chen, C. S., & Pianese, F. (n.d.). System Log Parsing: A Survey. Nokia Bell Labs.

Anexos.

Anexo 1.

Descripción de las bases de datos utilizadas por los modelos del estado del arte.

BGL Dataset:

- Contiene más de 4.7 millones de mensajes de registro de un sistema supercomputador Blue Gene/L.
- Con 348,460 (7.34%) marcados como anómalos.
- Es valioso para evaluar modelos en entornos de alto rendimiento y sistemas altamente paralelos.

Spirit Dataset:

- Proviene de un cluster de producción Linux y contiene más de 272 millones de mensajes de registro.
- Con 63.47% etiquetados como anomalías del sistema.
- Util para evaluar el desempeño de modelos en la detección de anomalías a gran escala.

HDFS Dataset:

- Generado a partir de trabajos basados en Hadoop sobre más de 200 nodos de Amazon EC2.
- Contiene más de 11 millones de registros, con aproximadamente 2.9% anómalos.
- Permite evaluar modelos en entornos de procesamiento distribuido y detección de anomalías.

OpenStack Dataset:

- Creado en un entorno de CloudLab con datos de un sistema operativo en la nube.
- Tiene 1,335,318 entradas de registro, con cerca del 7% anómalas.
- Sirve para evaluar la eficacia de los modelos en entornos de computación en la nube y detección de anomalías inyectadas.

Hadoop-blk Dataset:

- Compuesto por registros de un clúster Hadoop de 5 nodos.
- Contiene etiquetas de anomalías secuenciales y temporales, útil para confirmar la capacidad de detección de anomalías en series temporales de registros.

Anexo 2.

Figura 15. Evaluación del modelo por medio de accuracy, recall y f1-score, con un total de 89768 líneas evaluados por Drain.

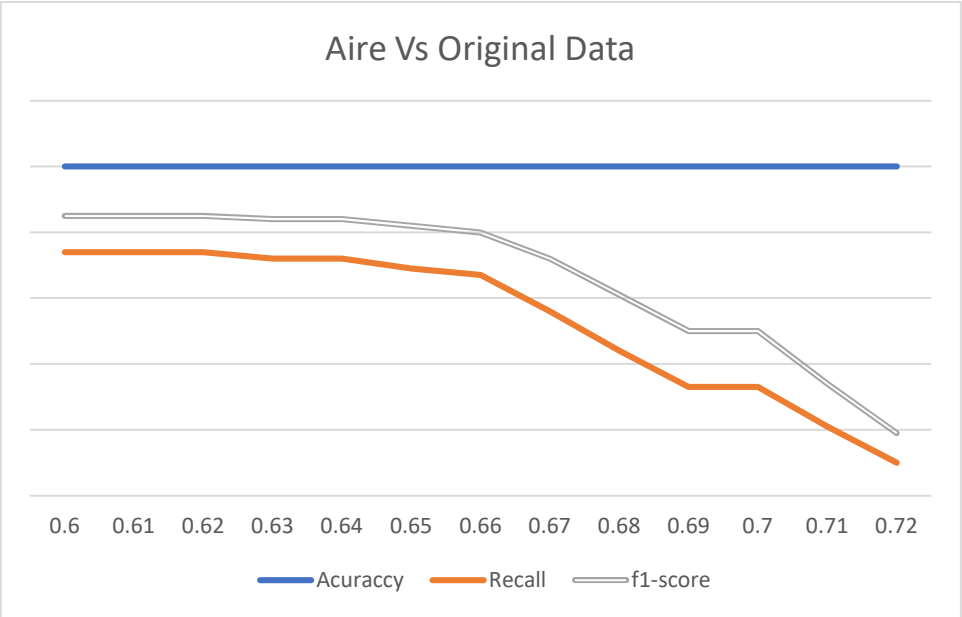


Tabla 5. Métricas con un total de 89768 líneas evaluados por Drain.

Similarity	Accuraccy	Recall	f1-score
0.6	1	0.74	0.85
0.61	1	0.74	0.85
0.62	1	0.74	0.85
0.63	1	0.72	0.84
0.64	1	0.72	0.84
0.65	1	0.69	0.82
0.66	1	0.67	0.8
0.67	1	0.56	0.72
0.68	1	0.44	0.61
0.69	1	0.33	0.5
0.7	1	0.33	0.5
0.71	1	0.21	0.34
0.72	1	0.1	0.19

Figura 16. Evaluación del modelo por medio de accuracy, recall y f1-score, con un total de 436064 líneas.

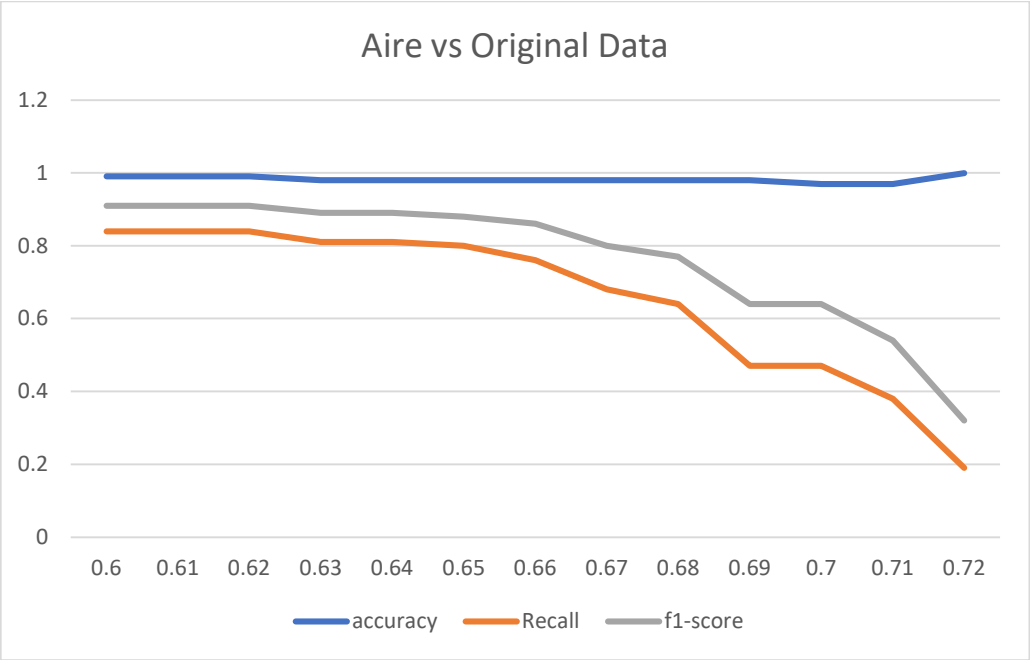


Tabla 6. Métricas de la prensa con un total de 436064 líneas evaluados por Drain.

Similarity	Acuraccy	Recall	f1-score
0.6	0.99	0.84	0.91
0.61	0.99	0.84	0.91
0.62	0.99	0.84	0.91
0.63	0.98	0.81	0.89
0.64	0.98	0.81	0.89
0.65	0.98	0.8	0.88
0.66	0.98	0.76	0.86
0.67	0.98	0.68	0.8
0.68	0.98	0.64	0.77
0.69	0.98	0.47	0.64
0.7	0.97	0.47	0.64
0.71	0.97	0.38	0.54
0.72	1	0.19	0.32