UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN

Extracción de Conocimiento en Bases de Datos

I.3. Reporte de investigación de los tipos de aplicaciones, procesamiento y herramientas para inteligencia artificial, machine learning, data mining y big data.

IDGS91N

PRESENTA:

JORGE ALEJANDRO HERNANDEZ CONTRERAS

DOCENTE:

Enrique Mascote

Chihuahua, Chih.,25 de Septiembre de 2025

Introducción

En la actualidad, la información se genera y se procesa a una velocidad sin precedentes. Tecnologías como la Inteligencia Artificial, el Machine Learning, la Minería de Datos y el Big Data se han convertido en pilares para comprender y aprovechar ese caudal de datos. Este reporte tiene como propósito explorar y comparar estos cuatro dominios: su definición, las aplicaciones más comunes en cada uno, las modalidades de procesamiento que utilizan batch y streaming y las herramientas más representativas.
La meta es ofrecer una visión clara y organizada que ayude a comprender cómo cada uno de estos campos contribuye a resolver problemas del mundo real, desde la recomendación de productos hasta el análisis en tiempo real de millones de eventos. A lo largo del documento se incluyen ejemplos concretos y herramientas ampliamente usadas, para que el lector pueda relacionar la teoría con aplicaciones prácticas y actuales.

Inteligencia Artificial

Definición breve

Inteligencia Artificial es el campo de la informática que diseña sistemas capaces de realizar tareas que, si las realizara un humano, requerirían inteligencia, percepción, razonamiento, planificación, procesamiento del lenguaje natural y toma de decisiones.

Tipos de aplicaciones

Visión por computador

Descripción:
Aplicaciones que extraen información y toman decisiones a partir de imágenes o video.
Ejemplo concreto:
Sistemas de visión para inspección industrial.

Procesamiento de lenguaje natural (PLN / NLP)

Descripción:
Modelos que analizan y generan texto o voz.
Ejemplo concreto:
Asistentes virtuales y chatbots desplegados por empresas para atención al cliente.

Sistemas de recomendación

Descripción:
IA que predice preferencias de usuarios y sugiere productos, contenidos o acciones.

Ejemplo concreto:
Recomendaciones de productos en tiendas en línea.

Agentes autónomos y robótica

Descripción:
Robots o agentes que perciben el entorno y actúan.

Ejemplo concreto:
Robots de limpieza autónomos y vehículos autónomos.

Procesamiento

Batch vs. Streaming en IA

Batch:
Se procesan grandes conjuntos de datos de forma periódica. En IA se emplea habitualmente para entrenamientos de modelos, reentrenamientos periódicos y evaluación masiva.

Cuándo:
Cuando el entrenamiento requiere recorrer todo el dataset, la latencia no es crítica y se benefician optimizaciones por lotes.

Streaming:
Procesa eventos conforme llegan. En IA se emplea especialmente para inferencia en tiempo real, detección de anomalías, o sistemas que adaptan respuestas en milisegundos/segundos.

Cuándo:
Cuando la aplicación necesita baja latencia.

Herramientas

TensorFlow

Funcionalidad:
Librería y ecosistema para construir, entrenar y desplegar modelos de aprendizaje automático y redes neuronales.

Caso de uso:
Entrenamiento de redes profundas para visión por computador; despliegue en producción con TensorFlow Serving y TFX.

PyTorch

Funcionalidad:
Framework flexible y fácil de usar para desarrollo de modelos, soporte para entrenamiento distribuido y TorchServe para producción.

Caso de uso:
Prototipado rápido de modelos NLP y visión; investigación y despliegue en producción.

OpenAI / Modelos de lenguaje

Funcionalidad:
Modelos de lenguaje preentrenados para generación de texto, resumen, y tareas de PLN.

Caso de uso:
Chatbots avanzados, generación automática de contenido y asistentes virtuales.

Open Neural Network Exchang

Funcionalidad:
Formato y ecosistema para interoperabilidad entre frameworks.

Caso de uso:
Facilitar despliegue multiplataforma de modelos.

TensorFlow Serving

Funcionalidad:
Plataformas para servir modelos ML en producción y responder inferencias en tiempo real.

Caso de uso:
Implementar endpoints de inferencia en aplicaciones web o sistemas embebidos.

Machine Learning

Definición breve

Machine Learning es una subárea de la IA que se centra en algoritmos que aprenden patrones a partir de datos para hacer predicciones o tomar decisiones sin ser programados explícitamente para cada tarea.

Tipos de aplicaciones

Clasificación y detección de objetos

Descripción:
Asignar etiquetas a entradas.

Ejemplo:
Clasificación de correos como spam/no-spam.

Regresión y forecasting

Descripción:
Predicción de valores continuos.

Ejemplo:
Predicción de ventas o consumo energético.

Clustering

Descripción:
Agrupar instancias similares sin etiquetas.

Ejemplo:
Segmentación de usuarios para campañas de marketing.

Detección de anomalías

Descripción:
Identificar patrones inusuales.

Ejemplo:
Sistemas de detección de fraude bancario.

Procesamiento

Batch:
Entrenamiento y evaluación de modelos con datasets históricos. Ideal para procesamiento intensivo y análisis de calidad.

Streaming:
Actualización incremental de modelos e inferencias en tiempo real.

Herramientas

scikit-learn

Funcionalidad:
Biblioteca de algoritmos clásicos de ML fácil de usar en Python.

Caso de uso:
Prototipado rápido, tareas de ML tradicionales en datasets pequeños o medianamente grandes.

XGBoost

Funcionalidad:
Implementaciones optimizadas de árboles de decisión potenciados para alto rendimiento.

Caso de uso:
Competencias de ML y problemas tabulares.

MLflow

Funcionalidad:
Plataforma para gestionar el ciclo de vida de modelos.

Caso de uso:
Seguimiento de experimentos y reproducibilidad en proyectos ML.

Keras

Funcionalidad:
API para construir redes neuronales sobre TensorFlow.

Caso de uso:
Modelos de clasificación, regresión y redes profundas con desarrollo rápido.

Jupyter

Funcionalidad:
Entornos interactivos para desarrollo, experimentación y visualización de modelos.

Caso de uso:
Exploración de datos, demostraciones pedagógicas y prototipado.

Data Mining

Definición breve

Data Mining o minería de datos consiste en descubrir patrones útiles, relaciones y estructura en grandes conjuntos de datos mediante técnicas estadísticas, de ML y visualización.

Tipos de aplicaciones

Asociación y reglas de asociación

Descripción:
Encontrar relaciones frecuentes entre elementos.

Ejemplo:
Recomendaciones en retail basadas en coocurrencia de productos.

Clasificación y predicción

Descripción: Aplicar modelos para predecir categorías basadas en atributos.

Ejemplo: Predicción de clientes que abandonarán un servicio.

Segmentación y clustering

Descripción:
Agrupamiento de clientes/instancias para análisis exploratorio y marketing.

Ejemplo:
Segmentación de clientes por comportamiento de compra.

Detección de anomalías y reglas de interés

Descripción:
Identificación de patrones raros o sospechosos.

Ejemplo:
Identificación de transacciones anómalas.

Procesamiento

Batch:
Muy común en data mining tradicional, donde se analiza histórico para descubrir patrones y construir modelos.

Streaming:
Cada vez más usado para minería en tiempo real, se emplea cuando los patrones deben detectarse de inmediato.

Herramientas

Weka

Funcionalidad:
Suite de herramientas Java para preprocesamiento, algoritmos de minería y visualización.

Caso de uso:
Enseñanza y prototipado de técnicas de minería de datos.

RapidMiner

Funcionalidad:
Plataforma visual para flujo de trabajo de data mining y despliegue, integra preparación, modelado y validación.

Caso de uso:
Usuarios no necesariamente programadores que necesitan pipelines de minería de datos.

KNIME

Funcionalidad:
Plataforma modular de análisis y minería de datos con nodos visuales e integración con Python.

Caso de uso:
Preparación de datos, ETL y modelos colaborativos en entornos empresariales.

Orange

Funcionalidad:
Entorno visual para análisis, con widgets para visualización y aprendizaje automático.

Caso de uso:
Experimentos y enseñanza en minería de datos.

R

Funcionalidad:
Lenguaje y ecosistema de paquetes para análisis estadístico y minería.

Caso de uso:
Análisis estadístico profundo, reglas de asociación y modelado.

Big Data

Definición breve

Big Data se refiere a conjuntos de datos cuyo volumen, velocidad y variedad exceden la capacidad de las tecnologías tradicionales para capturarlos, almacenarlos, gestionarlos y analizarlos con eficiencia. Cuando se explotan correctamente, estos datos permiten obtener insights a escala empresarial, soporte para decisiones y automatización basada en eventos en tiempo real.

Tipos de aplicaciones

Almacenes de datos a escala

Descripción:
Ingesta, normalización y consolidación de datos procedentes de múltiples orígenes hacia un repositorio central para análisis posterior.

Ejemplo concreto:
Una compañía de e commerce que recoge logs de pedidos, eventos web y telemetría de sus microservicios para alimentar informes de negocio y modelos de atribución.

Analítica en tiempo real y streaming

Descripción:
Procesamiento continuo de flujos de eventos para generar resultados con baja latencia.

Ejemplo concreto:
Sistema de recomendaciones en tiempo real que ajusta la oferta a cada usuario según su clickstream y su sesión actual.

Procesamiento distribuido y cómputo en batch

Descripción:
Ejecución de procesos por lotes en clústeres para transformar y analizar grandes volúmenes.
Ejemplo concreto:
Procesamiento nocturno de todos los logs de una plataforma para generar métricas agregadas y datasets de entrenamiento para modelos.

Machine learning a gran escala

Descripción:
Entrenar modelos con datos distribuidos y servir scoring en batch o en línea.

Ejemplo concreto:
Motor de recomendación que reentrena un modelo de factores en un clúster y publica resultados para uso en la app.

Procesamiento

Batch:
Procesamiento por lotes para operaciones que admiten mayor latencia. Se usa cuando se necesita recorrer grandes volúmenes completos.

Streaming:
Procesamiento continuo de eventos conforme llegan; es la elección cuando se necesita baja latencia, detección inmediata o acciones en tiempo real. La línea entre ambos se ha ido difuminando, frameworks modernos permiten unificar lógica para batch y streaming.

Herramientas

Apache Hadoop

Funcionalidad:
almacenamiento distribuido y procesamiento por lotes en clúster; útil para almacenar grandes volúmenes en nodos económicos y ejecutar jobs MapReduce o batch.
Caso de uso:
archivo de logs históricos y ETL por lotes.

Apache Spark

Funcionalidad:
Motor unificado para procesamiento en batch y streaming, con APIs para SQL, MLlib y procesamiento distribuido, ampliamente usado para ETL, análisis interactivo y entrenamiento escalable.
Caso de uso:
Pipelines ETL, feature engineering y entrenamiento distribuido.

Apache Kafka

Funcionalidad:
Plataforma de mensajería/streaming para construir pipelines de eventos.
Caso de uso:
Bus de eventos entre microservicios, ingestión de clickstreams y fuente para consumidores en tiempo real.

Apache Flink

Funcionalidad:
Motor de procesamiento de streams con capacidad de estado, tolerancia a fallos y procesamiento en tiempo real con garantías. Ideal para análisis complejo de eventos y windowing avanzado.
Caso de uso:
Detección y correlación de eventos en tiempo real con tolerancia a datos tardíos.

Delta Lake

Funcionalidad:
Formatos y plataformas que añaden garantías ACID, manejo eficiente de metadatos y unificación de batch y streaming sobre data lakes.
Caso de uso:
Mantener tablas transaccionales en S3/ADLS con consultas consistentes y soporte para pipelines unificados.

Conclusión

Los cuatro dominios abordados en este trabajo, Inteligencia Artificial , Machine Learning, Data Mining y Big Data, conforman un ecosistema complementario que permite transformar datos en valor. Cada dominio aporta una pieza distinta: Big Data provee la infraestructura y las técnicas para almacenar y procesar volúmenes masivos de información; Data Mining aporta métodos estadísticos y exploratorios para descubrir patrones; Machine Learning entrega algoritmos que, a partir de ejemplos, generan predictores y automatizan decisiones; e Inteligencia Artificial integra modelos y sistemas para realizar tareas que simulaban razonamiento humano.

La elección entre batch y streaming es una decisión arquitectónica clave: el modo batch facilita procesamiento exhaustivo y optimizado, mientras que el streaming es imprescindible cuando la latencia y la reacción inmediata son requisitos del negocio. Muchas arquitecturas modernas adoptan un enfoque híbrido: usar batch para entrenamientos y resúmenes, y streaming para inferencias en línea y alertas en tiempo real. Finalmente, seleccionar la herramienta adecuada depende de los requisitos de latencia, volumen, consistencia y del equipo disponible. Frameworks como Apache Spark y soluciones de lakehouse permiten unificar flujos; Kafka y Flink son preferidos en escenarios de streaming intensivo; y soluciones administradas en la nube aceleran la puesta en producción. Evaluar trade-offs y mantener prácticas de observabilidad, gobernanza y reproducibilidad son factores determinantes para que los proyectos transformen datos en resultados reales y sostenibles.

Bibliografía

Apache Software Foundation. (n.d.). Apache Flink. https://flink.apache.org/

Apache Software Foundation. (n.d.). Apache Hadoop. https://hadoop.apache.org/

Apache Software Foundation. (n.d.). Apache Kafka. https://kafka.apache.org/

Apache Software Foundation. (n.d.). Apache Spark. https://spark.apache.org/

CatBoost developers. (n.d.). CatBoost. https://catboost.ai/

Databricks. (n.d.). What is Delta Lake? Databricks documentation. https://docs.databricks.com/

Databricks. (n.d.). Batch vs. streaming data processing. Databricks documentation. https://docs.databricks.com/data-engineering/batch-vs-streaming.html

Google LLC. (n.d.). TensorFlow. https://www.tensorflow.org/

Google Research. (n.d.). Colaboratory (Colab). https://colab.research.google.com/

IBM Corporation. (n.d.). What is big data? https://www.ibm.com/think/topics/big-data

Jupyter Project. (n.d.). Project Jupyter. https://jupyter.org/

KNIME. (n.d.). KNIME. https://www.knime.com/

LightGBM contributors. (n.d.). LightGBM documentation. https://lightgbm.readthedocs.io/

MLflow Project. (n.d.). MLflow. https://mlflow.org/

ONNX. (n.d.). ONNX – Open Neural Network Exchange. https://onnx.ai/

Orange Data Mining. (n.d.). Orange. https://orangedatamining.com/

PyTorch. (n.d.). PyTorch. https://pytorch.org/

RapidMiner (Altair). (n.d.). RapidMiner. https://rapidminer.com/

scikit-learn developers. (n.d.). scikit-learn. https://scikit-learn.org/

TensorFlow (Google). (n.d.). TensorFlow Serving. https://www.tensorflow.org/tfx/guide/serving

TorchServe / PyTorch. (n.d.). TorchServe documentation. https://docs.pytorch.org/serve/

XGBoost developers. (n.d.). XGBoost. https://xgboost.ai/