



UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**Desarrollo de algoritmo de detección de eventos de riesgo para
grúas horquillas basado en Deep Learning y Transfer Learning**

Programa de Doctorado en Ciencias de la Ingeniería con mención en Ingeniería

Eléctrica

LUCIANO IGNACIO RADRIGAN FIGUEROA

CONCEPCIÓN-CHILE

2024

Profesor Guía: Sebastián Godoy

Dpto. de Ingeniería Eléctrica, Facultad de Ingeniería

Universidad de Concepción

Índice

ACRÓNIMOS	4
Resumen.....	5
CAPITULO 1. Introducción.....	7
1.1 Definición de problema.	7
1.2 Sistemas de monitoreo de eventos peligrosos	10
1.3 Algoritmos de detección de actividades de riesgos	16
1.4 Transfer Learning.....	28
1.5 Caso de aplicación: Grúa Horquillas.....	37
1.6 Discusión General.....	41
CAPITULO 2. Objetivos y plan de trabajo	44
CHAPTER 2.....	44
2.1 Introducción	44
2.2 Hipótesis	44
2.3 Objetivo general.....	44
2.4 Objetivos específicos	44
2.5 Limitaciones.....	45
2.6 Metodología	46

CAPITULO 3. Diseño de modelos.....	56
CHAPTER 3.....	56
3.1 Etiquetado de datos	62
3.2 Algoritmo Propuesto	66
3.3 Algoritmo con Amazon Rekognition Custom Labels	70
3.4 Nvidia DeepStream SDK	74
3.5 Evaluación de modelos.....	75
3.6 Discusión de evaluación de modelos.....	81
CAPITULO 4. Transfer-Learning y generalización de modelos y evaluación de modelos.....	83
CHAPTER 4.....	83
4.1 Transfer Learning.....	83
4.2 Resultados de Algoritmos	87
4.3 Resultados de clasificador binaria	88
4.4 Resultados de clasificador multi-estado.....	97
4.4.1 Análisis de correlación.....	111
4.5 Significancia del modelo	114
CAPITULO 5. Inferencia de modelos en dispositivos Edge y Cloud.	118
CHAPTER 5.....	122
5.1 Funcionamiento de algoritmos en pruebas	122

CAPITULO 6. Discusión y Conclusiones	133
REFERENCIA.....	142

ACRÓNIMOS

I.o.T:	<i>Internet of Things.</i>
I.I.o.T:	<i>Industrial Internet of Things.</i>
MTBF:	<i>Mean time between failures</i>
MTTR:	<i>Mean time to repair</i>
OSHA:	<i>Occupational Safety and Health Administration</i>
CNN:	<i>Convolutional Neural Networks</i>
LSTM:	<i>Long short-term memory</i>
API:	<i>Application Programming Interface.</i>
FFT:	<i>Fast Fourier Transform.</i>
MQTT:	<i>Message Queue Telemetry Transport.</i>
IP:	<i>Internet Protocol.</i>
TCP:	<i>Transmission Control Protocol.</i>
RUL:	<i>Remaining useful life</i>
CM:	<i>Condition Monitoring.</i>
O.M.:	<i>Opportunistic maintenance.</i>

Resumen

Esta investigación tiene como objetivo diseñar e implementar un algoritmo basado en Deep Learning y Transfer Learning para la detección de eventos de riesgo en grúas horquillas, empleando una cantidad limitada de datos. La metodología se estructuró en varias etapas: (1) recopilación y etiquetado de videos según la normativa OSHA 3949; (2) preprocesamiento de datos, incluyendo técnicas de aumento de datos e imágenes; (3) entrenamiento de modelos supervisados para la clasificación de eventos de riesgo; (4) adaptación mediante Transfer Learning de modelos pre-entrenados, optimizando su generalización; y (5) evaluación y validación comparativa frente a herramientas estandarizadas como NVIDIA DeepStream SDK y Amazon Rekognition Custom Labels.

El algoritmo propuesto alcanzó un F1-score superior al 0.95, superando el umbral de 0.85 establecido para su aceptación. En comparación, Amazon Rekognition obtuvo por sobre un un F1-score 0.95, mientras que NVIDIA DeepStream SDK logró un 0.83. El modelo desarrollado demostró una mayor eficiencia en el uso de datos para el reentrenamiento y menor tiempo de inferencia en sistemas embebidos, destacando por su menor peso y capacidad para operar en entornos con recursos limitados.

Un aspecto crítico fue la capacidad del modelo para generalizar, una necesidad fundamental en el contexto industrial, donde la recopilación de datos etiquetados, especialmente sobre actividades de riesgo, es costosa y limitada. Estas actividades no pueden replicarse intencionalmente debido a las

implicaciones de seguridad, lo que plantea desafíos únicos para el diseño de modelos robustos. El uso de Transfer Learning fue clave para superar esta limitación, permitiendo al modelo aprender patrones generales de conjuntos de datos preexistentes y adaptarse eficazmente a nuevas condiciones operativas con pocos datos específicos. Este enfoque no solo mejoró la precisión, sino que también redujo los costos y el tiempo de desarrollo.

Asimismo, las pruebas en sistemas embebidos validaron el desempeño del modelo en entornos reales, asegurando su capacidad para operar en tiempo real. Esto resulta crítico para aplicaciones industriales donde la detección rápida y precisa de eventos de riesgo puede prevenir accidentes y optimizar la seguridad y la eficiencia operativa.

CAPITULO 1. Introducción

1.1 Definición de problema.

La operación segura de maquinaria crítica es un aspecto esencial en la industria, debido a los altos riesgos asociados con el mal uso de estos equipos. De hecho, la mala utilización de equipos críticos puede reducir la disponibilidad de estos hasta en un 37%, incrementando la tasa de fallas y, en consecuencia, la posibilidad de incidentes. Este tipo de mal uso también reduce la vida útil de la maquinaria en un 25% y, en situaciones más críticas, puede llevar a la pérdida total del equipo o incluso a provocar accidentes en los operadores [1]. .

Un enfoque central en los programas modernos de seguridad industrial es la mejora de la *Conciencia Situacional* (Situational Awareness o S.A.), que implica la capacidad de un operador para percibir y comprender el entorno, identificando condiciones cambiantes y anticipando posibles riesgos. La S.A. consiste en recopilar, procesar y comprender las señales ambientales para construir un conocimiento claro de lo que sucede, de modo que el operador pueda prever eventos potencialmente riesgosos [2]. Esta conciencia es cada vez más relevante en la operación de maquinaria pesada, donde un entorno en constante cambio puede incrementar el riesgo de accidentes.

El concepto de Industria 4.0 introduce tecnologías habilitadoras que han transformado la capacidad de monitoreo, conectividad y automatización de las operaciones industriales. Estas tecnologías incluyen la implementación de sensores, redes de comunicación de alta velocidad y sistemas de procesamiento

que permiten la recopilación y análisis continuo de datos sobre las condiciones operativas y ambientales de la maquinaria. De esta forma, las tecnologías de la Industria 4.0 contribuyen a fortalecer la Conciencia Situacional en los operadores, facilitando un monitoreo interconectado y continuo de la maquinaria y sus condiciones de operación, lo cual se traduce en prácticas operativas más seguras y eficientes [3] [4].

La masificación de sistemas de adquisición de datos, posibilitada por la reducción de costos y el aumento en la capacidad de cómputo de los componentes electrónicos, ha permitido que los sistemas de monitoreo de maquinaria industrial ya no se limiten a la recopilación de variables de operación tradicionales, sino que también puedan captar indicadores de riesgo en tiempo real. Estos sistemas son cada vez más críticos en industrias como la minería, la construcción y la logística, donde el uso seguro y eficiente de la maquinaria es un factor determinante en la productividad y seguridad de las operaciones.

En este contexto, el uso de algoritmos de Deep Learning (DL) y Transfer Learning para la detección de eventos de riesgo en maquinaria crítica, como las grúas horquillas, se presenta como una solución innovadora. Sin embargo, la implementación efectiva de estos algoritmos enfrenta un desafío importante: la generalización. Los modelos de DL suelen requerir una gran cantidad de datos para su entrenamiento, especialmente en aplicaciones de visión por computadora y detección de eventos. La capacidad de generalización de un modelo es su habilidad para reconocer eventos de riesgo en datos que no fueron parte del

conjunto de entrenamiento, permitiéndole funcionar eficazmente en diversas condiciones de operación [5].

Este desafío es particularmente significativo en el contexto de los eventos de riesgo. Debido a su propia naturaleza, los eventos de riesgo en maquinaria industrial no pueden replicarse de forma intencionada para generar datos de entrenamiento, ya que esto representaría un peligro para la seguridad del operador y de la máquina. Esta falta de datos representativos afecta la capacidad del modelo para generalizar, ya que los algoritmos de DL, sin un conjunto de datos suficientemente diverso y amplio, tienden a sobreajustarse a los patrones específicos observados en los datos de entrenamiento y pierden efectividad cuando se enfrentan a situaciones nuevas y no vistas. Además, el ruido presente en los datos industriales y la naturaleza no estacionaria de las operaciones industriales añaden complejidad al diseño de modelos de DL robustos [6]. La variabilidad en las condiciones operativas y en los entornos hace que los datos recopilados no siempre representen adecuadamente todas las situaciones de riesgo que pueden surgir. Este problema se ve exacerbado por la limitación en la cantidad de datos de entrenamiento que representan eventos de riesgo, ya que los registros de estos eventos suelen ser escasos debido a su rareza y a las políticas de seguridad que evitan la reproducción de dichas situaciones para fines de entrenamiento.

Ante, este escenario el Transfer Learning emerge como una técnica clave que permite mejorar la capacidad de generalización de los modelos de DL. En lugar de entrenar el modelo desde cero, el Transfer Learning permite aprovechar modelos pre-entrenados en grandes conjuntos de datos de dominio general para adaptarlos a un nuevo contexto con pocos datos específicos. Esta técnica es especialmente relevante en aplicaciones de seguridad industrial, donde los datos etiquetados de eventos de riesgo son limitados y la adquisición de nuevos datos es costosa y riesgosa.

1.2 Sistemas de monitoreo de eventos peligrosos

Respecto a los productos e investigaciones de sistemas de monitoreo de eventos peligrosos, podemos subdividirlos en 2 categorías:

- Monitoreo de la seguridad ocupacional de los operadores de equipos: estas soluciones se centran en analizar las habilidades y variables fisiológicas del operador. La adquisición de estas señales suelen ser cámaras y utilización de algoritmos de visión computacional [7]. Otra alternativa se basa en sistemas de adquisición de variables fisiológicas de forma directa, tales como electroencefalograma (EEG) y electrocardiograma (ECG). Estas señales se procesan para entregar alertas en tiempo real. Propuestas como esta última se encuentran desde soluciones maduras con un nivel TRL 9 como también investigaciones recientes que al hardware incorporan algoritmos de Deep Learning para la estimación de indicadores de riesgos a partir de las variables fisiológicas adquiridas [8] [9]. Estos sistemas presentan una serie de limitantes dentro de las que destacan: (i) Los operadores sienten que su

privacidad es pasada a llevar. (ii) Los sistemas de adquisición basado en imágenes son muy susceptibles a los cambios de iluminación del entorno monitoreado. (iii) Los sistemas de adquisición basado en sensores que miden variables fisiológicas suelen tener una gran cantidad de ruido al momento de adquirir las señales en operación, por lo que es fundamental un pre-procesamiento de la señal y la utilización de filtros. (iv) Los algoritmos que se basan en aprendizaje automático para la estimación de indicadores de actividad de riesgo basadas en estas señales no pueden ser generalizados y por cada implementación se debe realizar un reentrenando [10] .

- Monitoreo de operaciones de los equipos: en el contexto de la industria 4.0, se ha puesto el énfasis en los sistemas de monitoreo en tiempo real basados en sistemas embebidos e IoT. Utilizando esta tecnología es que se han realizado estudios que a través del monitoreo con sensores inalámbricos se estiman indicadores de riesgos de operación en maquinarias críticas. A nivel de investigaciones que incluyen el desarrollo de hardware estos tienen consideraciones de diseños como hacer frente a las duras condiciones ambientales; desafíos de cableado de instrumentación, energía y comunicaciones; y la falta de conexiones de suministro de energía mientras se evita la interferencia con las comunicaciones inalámbricas existentes; asegurando al mismo tiempo una solución rentable en términos de inversión de capital y costo de operación y mantenimiento; Implementación y configuraciones rápidas y sencillas con sensores no invasivos. Estos requisitos representan un conjunto de condiciones límite estrictas para

seleccionar un sistema de monitoreo. La tabla 1 muestra diferentes soluciones e investigaciones de sistemas de monitoreo en la que se especifica: (i) Nivel de madurez tecnológica (TRL) [11], (ii) Descripción de soluciones, (iii) Métricas que el sistema estima.

Tabla 1 Sistemas de monitoreo de condiciones.

Solución	Nivel de madurez tecnológica	Descripción	Métrica
Monitoreo de condición para equipos estacionarios críticos [12] [13] [14] [15]	TRL 6-TRL 9	<p>Uso de sistemas integrados por hardware y software, que monitorea equipos vibratorios. Estos sistemas están compuestos por acelerómetros y algoritmos de análisis específicos para un tipo de equipo y pueden ser OTM u OEM. Los equipos críticos monitoreados corresponden a fajas transportadoras, cribas vibratorias, Trituradoras, entre otros. Estos trabajos tienen en común la sensorización de un equipo, por lo que las métricas solo se prueban en un equipo particularmente crítico en un entorno específico.</p>	Mejora: (i) F1 en el reconocimiento de fallas catastróficas
Monitoreo de condición para equipos no estacionarios [16] [17] [18].	TRL 6-TRL 9	<p>Los sistemas con niveles TRL 8-9 realizan mantenimiento preventivo basado en tres variables principales: (i) tiempo de uso de la máquina (medido por horómetros), (ii) datos georreferenciados (medidos por GPS), (iii) medición de consumo de combustible y emisiones de gases. Por otro lado, las soluciones en TRL 6- TRL7 proponen realizar mediciones de vibraciones durante la operación no estacionaria de equipos críticos para caracterizar modos de operación. Los equipos en operaciones no estacionarias corresponden a excavadoras, retroexcavadoras, volquetas y palas mineras. La principal limitación de estos estudios es que son exploratorios para generar modelos de una máquina específica en un entorno determinado.</p>	Mejore la precisión para el diagnóstico y el horizonte de pronóstico de fallas aleatorias y de degradación realizados por algoritmos de aprendizaje automático
Sistemas de monitorización de estado de equipos críticos basados en el Internet de las Cosas. [19] [20] [21] [22] [23] [24]	TRL 4-TRL 9	<p>Estas soluciones se encuentran en varios niveles de TRL. La investigación que utiliza soluciones de nivel TRL8 -TRL9, tiene como objetivo mejorar estudiar el rendimiento de la arquitectura de la red de sistemas de monitoreo para la clasificación de estados de operación. Investigaciones que desarrollan soluciones a niveles TRL4-TRL7, comparan el rendimiento de las arquitecturas Cloud, Edge y la combinación de ambas para los sistemas de monitoreo</p>	Optimiza diferentes parámetros como el tráfico de datos, el sistema de reportabilidad y el consumo de energía.

<p>Sistema de monitoreo basado en IoT para el reconocimiento de eventos de riesgos [25] [26] [27]</p>	<p>TRL 4-TRL6</p>	<p>Estas investigaciones presentan soluciones que se centran en diseñar arquitecturas IoT con integración TI/TO, para generar alertas y/o alarmas de eventos de situaciones de riesgos. Cabe destacar que las arquitecturas propuestas se abstraen del proceso generador de datos, ya que se asume que la instrumentalización del proceso está realizada. Las arquitecturas propuestas y analizada varían desde Cloud hasta Edge.</p>	<p>Compara los tiempos de respuesta en entregar la alarma de la actividad riesgosa. También reportan los anchos de banda requerido según la arquitectura seleccionada (Cloud o Edge).</p>
<p>Plataformas Cloud con herramientas de reconocimiento de eventos [28] [29] [30].</p>	<p>TRL9</p>	<p>Estas soluciones se centran en procesar datos que pueden ser recopilados con diferentes sistemas (señales de sensores, fotos y/o videos). Estas soluciones Cloud cuentan con herramientas de reconocimientos de eventos en el cual se deben entregar información etiquetada y a través de algoritmos de Deep Learning se genera un modelo de reconocimiento de eventos y/o actividades. Todos los modelos generados en estas plataformas pueden ser utilizado solo en la plataforma, por lo que no se pueden exportar los modelos a dispositivos.</p>	<p>Entrega un algoritmo de reconocimiento de eventos u operación, mostrando el F1 del modelo.</p>

Respecto a las investigaciones y sistemas revisados en la Tabla 1, es posible identificar importantes retos en el desarrollo de instrumentación adecuada para la adquisición de variables que permitan reconocer actividades de riesgo de manera precisa y fiable. El diseño del hardware para estos procesos generadores de datos debe cumplir con varias características fundamentales: (i) bajo costo para facilitar su implementación a gran escala, (ii) capacidad para recopilar información que permita construir modelos de detección de eventos peligrosos que sean robustos y generalizables, (iii) compatibilidad con las tareas de operación de la maquinaria, y (iv) consumo energético reducido para optimizar su operatividad en entornos industriales. Los sistemas de monitoreo existentes han alcanzado un desarrollo considerable en el análisis de vibraciones para maquinaria crítica en operaciones estacionarias. Sin embargo, aún son limitados los estudios que exploran el monitoreo de condiciones en maquinaria durante operaciones no estacionarias, un contexto caracterizado por señales

probabilísticas y altamente no lineales [31]. Estas características representan un desafío significativo en el preprocesamiento de señales y en la selección de algoritmos de detección de eventos riesgosos, ya que requieren una instrumentación adaptable y precisa. Esto también representa una oportunidad para avanzar hacia arquitecturas mixtas, como las combinaciones de procesamiento en la nube y en el edge, que podrían optimizar los sistemas de monitoreo al reducir la latencia y mejorar la eficiencia y flexibilidad en la instrumentación mediante sensores.

Los estudios revisados en la tabla 1 basados en la tecnología de sensores y en algoritmos de aprendizaje automático han permitido desarrollar sistemas de monitoreo más precisos y adaptativos para la detección de eventos peligrosos en entornos industriales, especialmente cuando se combinan arquitecturas de procesamiento híbridas. Sin embargo, estos avances también plantean desafíos relacionados con la privacidad y adaptabilidad de los algoritmos utilizados. Para abordar las preocupaciones sobre la privacidad de los operadores, investigaciones recientes han diseñado soluciones que anonimizan los datos personales antes de procesarlos, protegiendo así la privacidad del operador. Por ejemplo, algunos estudios del área de ciberseguridad proponen sistemas que procesan los datos en dispositivos edge en lugar de enviarlos a la nube, lo cual reduce la exposición de información sensible y mejora la aceptación del usuario [32] [33]. La ventaja de estos sistemas basados en edge reside en su capacidad para procesar datos localmente, lo que contribuye a un mayor control sobre la información sensible y limita la transferencia de datos a servidores externos,

mitigando así las preocupaciones de privacidad. Finalmente, respecto a los sistemas de monitoreo revisados en la tabla 1, podemos ver que el desarrollo de sistemas IoT específicos para maquinaria crítica ha mostrado avances en su utilización, particularmente en operaciones no estacionarias donde la no linealidad de las señales y las condiciones variables representan un reto para los sistemas de monitoreo convencionales. La mayoría de estos sistemas se encuentran en arquitecturas Cloud a pesar de que existen avances emergentes en la utilización de arquitecturas Edge e híbrida (mezcla entre Cloud y Edge) que han demostrado que no solo minimizan la latencia y optimizan el ancho de banda, sino que también proporcionan una mayor flexibilidad y eficiencia en la instrumentación de maquinaria crítica. Estos enfoques permiten a los sistemas de monitoreo responder rápidamente a eventos de riesgo en tiempo real, facilitando una implementación más flexible y eficiente en entornos industriales [34] [35].

1.3 Algoritmos de detección de actividades de riesgos

El paradigma de la Industria 4.0 está impulsando la relevancia de la detección en tiempo real de eventos de riesgo, particularmente en entornos donde las operaciones de maquinaria crítica requieren una supervisión constante y precisa. La integración de capacidades de ciencia de datos en los procesos generadores de datos permite no solo la captura y procesamiento de grandes volúmenes de información, sino también el análisis avanzado de patrones y comportamientos. Esto facilita la identificación de condiciones de operación seguras y potenciales riesgos en tiempo real, contribuyendo a decisiones y acciones óptimas en la operación de equipos críticos. La Figura 1 muestra cómo estos procesos generan información que permite anticipar y mitigar riesgos, optimizando la productividad y reduciendo los tiempos de inactividad al alertar de posibles fallos o situaciones de peligro antes de que ocurran [36].

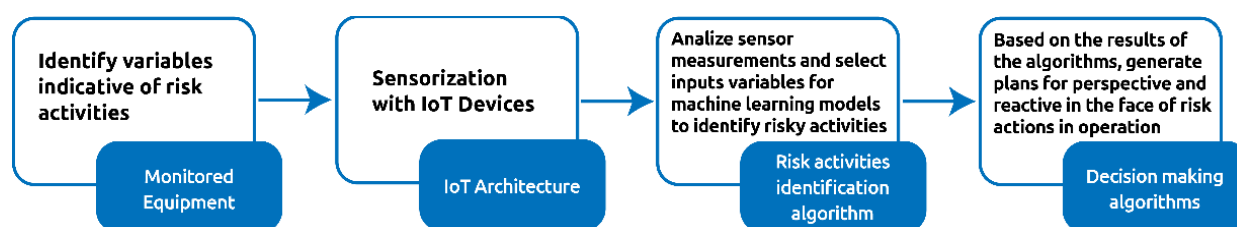


Fig. 1 Procesos para la de detección de eventos de riesgos [36].

La tarea de detección de actividad de riesgo se basa en la clasificación de los datos de entrada, asociando cada ventana de tiempo con un modo específico de operación de la máquina. Este enfoque permite identificar no solo las condiciones normales de funcionamiento, sino también las variaciones que pueden indicar condiciones inusuales o riesgosas. Al categorizar los modos de operación de esta

manera, los algoritmos pueden aprender a reconocer patrones de comportamiento tanto seguros como peligrosos, lo cual es fundamental en entornos industriales donde cualquier retraso en la identificación de un riesgo puede tener consecuencias graves en términos de seguridad y eficiencia operativa [37].

Además, la aplicación de técnicas de machine learning y deep learning en la clasificación de estas ventanas temporales amplía la capacidad de estos sistemas para generalizar a nuevos escenarios y condiciones operativas, adaptándose a contextos variables y dinámicos propios de la Industria 4.0 [38]. Con el uso de estos métodos, los modelos de detección de riesgos pueden entrenarse para aprender de ejemplos previos y adaptar sus predicciones a situaciones desconocidas, mejorando así la respuesta ante eventos críticos en tiempo real. La Tabla 2 muestra una comparación de varios algoritmos utilizados para la detección de actividades de riesgo. Además, se ilustra un resumen de sus ventajas y desventajas.

Tabla 2 Comparación de algoritmos para detección de actividades de riesgo

Familia de algoritmos	Proceso generador de Datos	Ventajas	Desventajas
Modelos de Markov [39]	<ul style="list-style-type: none"> Los modelos de Markov se utilizan con series de tiempo de datos estacionarios y determinísticos. Dentro de estos destacan utilización para estimación de indicadores de mantenimiento tales como MTTR, MTTF y MTBF. No es necesario que el proceso de generador de datos tenga una frecuencia de muestreo alta. 	<ul style="list-style-type: none"> Adecuado para reconocimiento de múltiples modos de conocimientos específicos para el reconocimiento de eventos. Gestión rápida de conjuntos de datos incompletos. 	<ul style="list-style-type: none"> Requiere grandes cantidades de datos de entrenamiento, proporcionales al número de estados ocultos. Computacionalmente ineficiente para grandes datos de entrada. Inexacta para un gran número de condiciones de la máquina.
Máquinas de vectores soporte [40]	<ul style="list-style-type: none"> Los datos suelen ser simulaciones de funcionamiento de equipos en diferentes modos de 	<ul style="list-style-type: none"> Resultados robustos y precisos con gran entrada no lineal. 	<ul style="list-style-type: none"> No hay un método estándar para elegir la función del núcleo. Series temporales multivariadas difíciles de procesar.

	operación, se utilizan un modelo por cada proceso generador de datos. No es necesario que el proceso de generador de datos tenga una frecuencia de muestreo alta.	<ul style="list-style-type: none"> • Buen rendimiento de generalización. • Aplicado con éxito en el reconocimiento de actividades o modos de operación. 	<ul style="list-style-type: none"> • Baja adaptabilidad que requiere reajuste de parámetros. • Altamente dependiente del proceso de extracción de características
LSTM [41]	<ul style="list-style-type: none"> • Los datos requieren una alta tasa de muestreo y además deben caracterizar procesos en series de tiempo. Una de las utilidades de las LSTM es predecir actividades físicas de los usuarios de smartphone a través de los sensores inerciales que estos traen. Estas señales se caracterizan por ser no lineales y probabilísticas. 	<ul style="list-style-type: none"> • Adecuado para secuencias de datos largas. • Aprende características temporales de los datos. 	<ul style="list-style-type: none"> • Entrenamiento lento debido a la baja estructura paralelizable • Requiere grandes cantidades de datos de entrenamiento.
Redes neuronales convolucionales [41]	<ul style="list-style-type: none"> • Los datos requieren una alta tasa de muestreo. Una de las utilidades de las CNN es clasificar estados de operación, analizando variables ya sean de sensores o de videos/imágenes. En el caso de la clasificación de series de tiempo multi canales se debe transformar a una serie de imágenes. Estas señales se caracterizan por ser no lineales y probabilísticas. 	<ul style="list-style-type: none"> • Puede modelar sistemas multidimensionales y no lineales. • Las técnicas de regularización aportan inmunidad al ruido. • Modela fenómenos complejos sin conocimientos previos. • Cómputo rápido y operación en tiempo real debido a la estructura paralelizada. • Manejo rápido de análisis multivariante. • Comportamiento de tolerancia a fallas a través de información redundante. 	<ul style="list-style-type: none"> • Requiere el procesamiento previo de una gran cantidad de datos (representativos). • Funciona como una "caja negra", por lo que no aclara las decisiones tomadas. • Se supone que la falla ocurre después del índice de condición. • No tiene en cuenta los estados operativos anteriores.
Modelos Gaussianos [42]	<ul style="list-style-type: none"> • Estos modelos se utilizan con datos de distribuciones conocidas a priori y que no tienen una alta tasa de muestreo. Suelen ser utilizados para el reconocimiento de modos de funcionamiento de simulaciones multifísicas de procesos. 	<ul style="list-style-type: none"> • Calcula predicciones de degradación no lineales y soluciones a problemas de clasificación complejos. • Predice la incertidumbre a través de la varianza alrededor de la predicción. • Alta adaptabilidad para manejar datos de alta dimensión. • Permite la función de aprendizaje de progresión no paramétrica de datos ruidosos. 	<ul style="list-style-type: none"> • Gran carga computacional, especialmente con grandes conjuntos de datos de entrenamiento. • Supone que el ruido en el entrenamiento es constante. • Altamente dependiente del proceso de extracción de características

En el contexto de análisis de reconocimiento de eventos de “riesgo” en equipos críticos industriales, y específicamente a grúas horquillas, se debe considerar que las señales recopiladas por el proceso generador de datos son dependientes del tiempo, no lineales, probabilísticas y con niveles de ruidos variables en el tiempo. Es debido a la morfología de las señales que los modelos de redes neuronales convolucionales (CNN) y las redes neuronales de memoria a corto plazo (LSTM) son de especial interés. Los modelos basados en CNN muestran más ventajas que desventajas, ya que son rápidos de implementar y robustos frente a datos ruidosos, pero se centran solo en las correlaciones de datos locales, sin considerar los estados anteriores.

Los modelos CNN constan de n -capas convolucionales, las cuales tienen un número decreciente de filtros. De manera similar, el tamaño del kernel también disminuye en las capas. Esta arquitectura permite que la red aprenda patrones más específicos a medida que los datos pasan a través de las capas. Cada capa convolucional tiene una función de activación y es seguida por un bloque de compresión y excitación. Las salidas de cada filtro describen un mapa de características que será procesado por el bloque de compresión y excitación. Este bloque descarta o enfatiza información de cada canal del mapa de características. Luego, los mapas de características se multiplican por un conjunto de sus características recuperadas de un perceptrón de capas totalmente conectadas [41], como se observa en la figura 2.

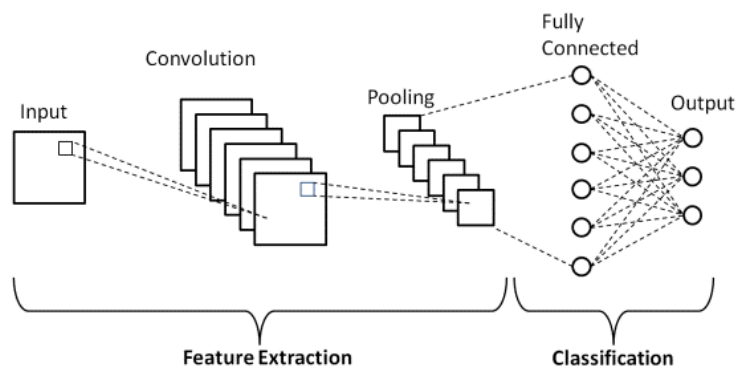


Fig. 2 Diagrama CNN.

Por otro lado, las LSTM extraen características de las correlaciones de datos secuenciales, siendo especialmente útiles para el análisis de eventos temporales. Las LSTM son un tipo de red neuronal recurrente (RNN) y están diseñadas explícitamente para evitar el problema de la dependencia a largo plazo. Todas las RNN tienen la forma de una cadena de módulos repetidos de red neuronal, como se observa en la figura 3.

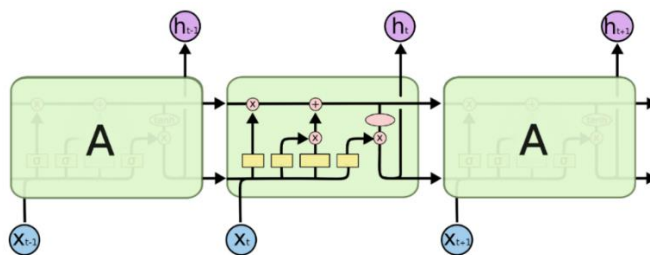


Fig. 3 Diagrama LSTM.

Por lo tanto, la combinación de redes CNN y LSTM en un modelo híbrido brinda una alta precisión para la detección de eventos de riesgos instantáneos y temporales. La implementación de algoritmos de detección de actividades de riesgo basados en Deep-learning ha demostrado ser altamente efectiva en entornos industriales, específicamente en la identificación de movimientos inusuales, mal funcionamiento de equipos y situaciones de riesgo. Estos

algoritmos, respaldados por redes neuronales profundas como las convolucionales (CNN) y las de memoria a corto plazo (LSTM), destacan por su capacidad para capturar y analizar características complejas en datos secuenciales y no lineales, como las señales provenientes de equipos críticos. La combinación de modelos CNN y LSTM en un enfoque híbrido permite una detección precisa, abordando tanto aspectos instantáneos como temporales de los eventos de riesgo [43].

Para abordar la complejidad de casos altamente no lineales o procesos probabilísticos en la detección de eventos de riesgo, se recurre a métodos de ensamble como *Bagging*, *Boosting* y *Stacking*. Estos enfoques combinan diversas técnicas de machine learning para mejorar el rendimiento de los modelos, proporcionando mayor robustez en contextos de detección de riesgo [44] [45]. Los métodos de ensamble son especialmente efectivos para manejar el balance entre sesgo y varianza, dos propiedades clave que afectan el desempeño de los modelos de machine learning.

Los modelos de machine learning para clasificación y regresión aprenden una función de mapeo de las entradas a las salidas, y su rendimiento se evalúa en un conjunto de datos de prueba. El error de estos modelos se caracteriza comúnmente por el sesgo (bias) y la varianza [46]. El sesgo refleja la capacidad del modelo para capturar la función subyacente entre las entradas y las salidas; un alto sesgo suele llevar a un ajuste insuficiente, donde el modelo omite relaciones importantes entre las características y las salidas deseadas. La varianza, en cambio, mide la sensibilidad del modelo a las fluctuaciones de los

datos de entrenamiento, y un alto valor de varianza puede llevar a un sobreajuste, donde el modelo aprende el ruido de los datos en lugar de los patrones generales [47].

Encontrar el equilibrio entre sesgo y varianza es un desafío común. Modelos con alta varianza pueden ajustar bien el conjunto de entrenamiento, pero corren el riesgo de sobreajustarse al ruido, mientras que los algoritmos de alto sesgo tienden a producir modelos más simples que pueden no capturar patrones importantes. Los métodos de ensamble ayudan a manejar este compromiso entre sesgo y varianza, combinando múltiples modelos para mejorar tanto la precisión como la generalización.

A continuación se detallan algunos de los métodos de ensamble más utilizados para mejorar la precisión y reducir la varianza y el sesgo:

- Bagging: Este método considera modelos de aprendizaje débiles y homogéneos, entrenados de manera independiente entre sí en paralelo, y los combina promediando sus predicciones. Bagging ayuda a reducir la varianza al promediar los resultados de múltiples modelos, lo cual es útil para reducir el riesgo de sobreajuste en modelos de alta varianza como los árboles de decisión [48]. El proceso de Bagging implica entrenar varios modelos en diferentes subconjuntos del conjunto de datos, lo cual mejora la estabilidad y generalización del modelo final.
- Random Forests: Una mejora del Bagging que introduce aleatoriedad en la selección de características durante la construcción de árboles. Al reducir la correlación entre los árboles, Random Forests optimiza aún más

la precisión y mantiene la capacidad interpretativa de los árboles de decisión individuales [49]. Este método ofrece una mayor diversidad entre los modelos al seleccionar subconjuntos aleatorios de características para cada árbol, lo cual contribuye a un rendimiento robusto en tareas de clasificación complejas.

- **Boosting:** Este método consiste en construir secuencialmente una serie de modelos de aprendizaje débiles, donde cada modelo se entrena para corregir los errores del anterior. La combinación de estos modelos produce un modelo final más fuerte y adaptativo. Boosting reduce el sesgo al enfocarse en los errores del conjunto de datos y mejorar progresivamente el ajuste del modelo. En el contexto de la detección de riesgos, Boosting es útil para mejorar la precisión en escenarios donde los eventos de riesgo son poco frecuentes o complejos de clasificar [49].
- **Stacking:** Este enfoque combina modelos de rendimiento variado y luego entrena un metamodelo para generar predicciones basadas en las predicciones de estos modelos débiles. Stacking permite que el metamodelo aproveche la diversidad de los modelos base para mejorar el rendimiento general [50]. A diferencia de otros métodos de ensamble, Stacking permite alinear la variabilidad entre los modelos y adaptar la predicción final a diferentes subgrupos de datos, maximizando la precisión y flexibilidad en la detección de riesgos.

Además de estos métodos, la reducción de dimensionalidad y la selección de características son técnicas que ayudan a reducir la varianza simplificando el modelo. En algunos casos, aumentar el tamaño del conjunto de datos de entrenamiento o agregar predictores adicionales puede disminuir el sesgo, aunque esto puede introducir más varianza.

La importancia de gestionar el sesgo y la varianza en los modelos de machine learning también incluye la identificación y corrección de sesgos en los datos de entrada. El sesgo de los datos (data bias) puede surgir debido a diversos factores, como el sesgo de selección, sesgo de exclusión y sesgo de medición, entre otros. Un conjunto de datos sesgado puede llevar a modelos con resultados inexactos y bajos en precisión, afectando la eficacia de los sistemas de detección de riesgo. Los métodos de ensamble, al combinar predicciones de múltiples modelos, también ayudan a mitigar los efectos del sesgo al mejorar la representatividad de las predicciones finales [51] [52].

Un *stacking* permite una implementación de un modelo híbrido que combina CNN y LSTM ha avanzado significativamente. Las CNN se encargan de extraer características locales y específicas de los datos, identificando patrones relevantes asociados a eventos de riesgo. Por otro lado, las LSTM capturan las dependencias temporales en los datos, permitiendo la detección de patrones que evolucionan a lo largo del tiempo [53].

La integración de características extraídas de ambas arquitecturas es crucial para lograr una detección precisa de eventos de riesgo. Después de la extracción de características por parte de las CNN y LSTM, se combinan en capas

posteriores del modelo, permitiendo al modelo híbrido aprovechar la información tanto local como temporal para tomar decisiones más informadas [54].

Este enfoque híbrido ha demostrado ser altamente eficiente en numerosos estudios, donde se destaca la mejora significativa en la capacidad de detección y la adaptabilidad a escenarios industriales específicos. Este modelo híbrido se posiciona como una herramienta valiosa para la seguridad y la prevención de riesgos en entornos industriales, al combinar la fortaleza de las CNN en la extracción de características locales con la capacidad temporal de las LSTM para capturar patrones en evolución [55] [56] [57].

La adaptabilidad continua de los algoritmos de detección de actividades de riesgo basados en deep learning en entornos industriales es un aspecto crítico para garantizar su eficacia a medida que las condiciones y patrones cambian con el tiempo. Además, la adaptabilidad del modelo está directamente vinculada a la disponibilidad de datos representativos, siendo la falta de diversidad o la presencia de sesgos en los datos desafíos a superar. Otro desafío es el riesgo de *overfitting* en actualizaciones frecuentes, lo que requiere una gestión cuidadosa de la actualización continua para evitar hacer un sobreajuste a datos específicos y garantizar la generalización del modelo a nuevas condiciones [58] [59] [60].

Sin embargo, a pesar de las ventajas que muestran algoritmos de detección de eventos de riesgo basado en CNN, LSTM y métodos de ensamble, existen aspectos asociados a la cantidad de datos de entrenamiento. El transfer learning, o aprendizaje por transferencia, es una técnica en el campo del aprendizaje

automático y la inteligencia artificial en la que se aprovecha el conocimiento adquirido por un modelo en una tarea para mejorar el desempeño en otra tarea relacionada, esto tiene como objetivo disminuir el impacto que tiene un pequeño o bajo conjunto de datos de entrenamiento [61]. Esto se logra entrenando un modelo desde cero para cada tarea específica, luego el modelo se entrena primero en una tarea relacionada y luego se adapta para la tarea de interés. La idea es que los modelos de aprendizaje automático pueden aprender patrones generales en los datos que se pueden transferir a otras tareas [62]. En general, el uso de transfer learning en algoritmos de aprendizaje automático basados en video puede mejorar la precisión, reducir el tiempo y el costo de entrenamiento, mejorar la generalización, manejar conjuntos de datos pequeños y permitir la adaptabilidad a nuevas tareas, tal como muestra la Tabla 3.

Tabla 3 Ventajas, desventajas y limitaciones de transfer learning

Proceso generador de Datos	Ventajas	Desventajas
<ul style="list-style-type: none"> • Dependencia del modelo pre-entrenado: el rendimiento del modelo depende en gran medida de la calidad y la relevancia del modelo pre-entrenado utilizado. • Riesgo de sobreajuste: es posible que el modelo pre-entrenado tenga características específicas que sean relevantes para la tarea original, pero que no sean relevantes para la tarea específica que se está resolviendo. • Dificultad para ajustar parámetros: debido a la complejidad del modelo pre-entrenado, puede ser difícil ajustar los parámetros para la tarea específica que se está resolviendo. 	<ul style="list-style-type: none"> • Reducción del tiempo y del costo de entrenamiento: al utilizar un modelo pre-entrenado, se reduce el tiempo y el costo que se necesitaría para entrenar un modelo desde cero. • Mejora de la precisión: al utilizar un modelo pre-entrenado, se puede mejorar la precisión del modelo para la tarea que se está resolviendo. • Adaptabilidad: el transfer learning permite adaptar modelos pre-entrenados a tareas específicas con diferentes datos de entrada. • Generalización: el modelo pre-entrenado ya ha aprendido a generalizar, por lo que puede aplicarse a nuevas situaciones que no se han visto durante el entrenamiento 	<ul style="list-style-type: none"> • Restricciones de dominio: el modelo pre-entrenado debe ser relevante para la tarea específica que se está resolviendo. • Diferentes distribuciones de datos: si los datos utilizados para entrenar el modelo pre-entrenado son diferentes de los datos utilizados para la tarea específica, la precisión del modelo puede verse afectada. • Transferencia limitada: es posible que el conocimiento adquirido durante la resolución de una tarea no sea útil para la resolución de otra tarea. • Dificultad para interpretar el modelo.

La integración de algoritmos de detección de actividades de riesgo basados en deep learning en los sistemas industriales actuales es un proceso crítico que implica consideraciones técnicas y económicas. Estos algoritmos, respaldados por redes neuronales profundas, deben incorporarse de manera armoniosa en la infraestructura existente, interactuando con otros sistemas y sensores para proporcionar una detección integral de riesgos. Estudios como el trabajo de Ramin Moradi et al. resaltan la importancia de una integración cuidadosa para maximizar la eficacia de estos algoritmos [63]. La evaluación del impacto de la implementación incluye análisis de costos y beneficios, considerando aspectos como la mejora de la seguridad y la eficiencia operativa. Se espera un retorno de inversión positivo, ya que la detección precisa de riesgos puede prevenir accidentes costosos y tiempos de inactividad no planificados. Sin embargo, es crucial realizar un análisis exhaustivo para garantizar una implementación rentable y sostenible en entornos industriales específicos, como sugiere el informe de la Asociación de Tecnologías Industriales. La relevancia del transfer learning en este contexto es fundamental, ya que permite superar la limitación de conjuntos de datos pequeños en entornos industriales. Al aprovechar modelos pre-entrenados en tareas relacionadas, se logra mejorar la capacidad de generalización de los algoritmos, adaptándose a condiciones operativas cambiantes y garantizando una detección continua y precisa a medida que evolucionan los patrones de riesgo [43].

1.4 Transfer Learning

La problematización en el ámbito de los modelos de Machine Learning y Deep Learning se centra en su capacidad para generalizar patrones a partir de datos de entrenamiento y aplicar estos conocimientos a nuevos datos. A menudo, los modelos pueden enfrentar desafíos significativos en términos de generalización, lo que puede dar lugar a resultados ineficaces o inexactos. Un ejemplo de esta problemática es la clasificación de imágenes en un modelo de reconocimiento de objetos. Supongamos que entrenamos un modelo de Deep Learning para reconocer gatos y perros en imágenes. Si bien el modelo puede funcionar excepcionalmente bien en datos de entrenamiento y validación, puede enfrentar dificultades al generalizar cuando se enfrenta a imágenes de gatos y perros en situaciones inusuales o con condiciones de iluminación adversas. En tales casos, el modelo puede cometer errores significativos y, en última instancia, no ser confiable en aplicaciones del mundo real. Si bien en el ejemplo de la detección de gatos y perros puede ser un impacto trivial, si el mismo sistema de detección está diseñado para identificar operaciones de riesgo en industria su impacto puede ser fatal. Por lo que si el algoritmo identifica un evento como un riesgo cuando en realidad no lo es (o viceversa), puede tener importantes implicaciones negativas en términos de costos, recursos y seguridad. Los falsos positivos en este escenario pueden causar pánico innecesario, activación de alarmas de forma indebida o recursos desperdiciados en la respuesta a eventos no reales, lo que puede ser costoso. Por otro lado, los falsos negativos comprometen la seguridad y son potencialmente peligroso [63].

Para abordar esta limitación en la generalización, se pueden emplear varias técnicas y enfoques. Una estrategia es aumentar la cantidad y diversidad de datos de entrenamiento, lo que permite al modelo aprender patrones más robustos y variados. Además, se pueden aplicar técnicas de regularización, como la disminución de la complejidad del modelo o la incorporación de técnicas como el dropout. El uso de arquitecturas de modelos más avanzadas, como redes neuronales convolucionales (CNN) pre-entrenadas, mediante la transferencia de aprendizaje, también puede ayudar a mejorar la generalización. Además, la optimización de hiperparámetros y la validación cruzada son prácticas importantes para garantizar un mejor rendimiento general en una variedad de escenarios.

El Transfer Learning permite aprovechar el conocimiento previo adquirido por modelos de Deep Learning previamente entrenados en conjuntos de datos masivos y tareas relacionadas. Esto resulta especialmente beneficioso en situaciones donde la disponibilidad de datos de entrenamiento es limitada, como es común en aplicaciones de seguridad industrial. Para la detección de actividades de riesgo, el Transfer Learning es particularmente ventajoso, ya que facilita que los modelos logren una mayor generalización con conjuntos de datos de entrenamiento acotados y reduzcan su dependencia de situaciones de riesgo que son difíciles de replicar o etiquetar. Esta capacidad para adaptarse a tareas relacionadas permite que los modelos sean operativos y efectivos incluso con pocos ejemplos de eventos riesgosos específicos, aprovechando características previamente aprendidas en tareas análogas.

Además, el uso de Transfer Learning reduce considerablemente la necesidad de recopilar y etiquetar grandes conjuntos de datos desde cero. Esto implica un ahorro significativo en términos de tiempo y recursos, ya que el modelo puede ser adaptado a datos específicos de grúas horquillas con un número mucho menor de muestras de entrenamiento en comparación con el entrenamiento desde cero. Otra ventaja clave es la mejora en la velocidad de entrenamiento. Dado que gran parte del modelo ya está pre-entrenada en características generales, el proceso de fine-tuning o ajuste fino se vuelve más rápido y eficiente. Esto es especialmente valioso cuando se busca implementar rápidamente un sistema de detección de eventos de riesgo [64].

El Transfer Learning es también un método eficaz para extender la vida útil y relevancia de un modelo en un entorno industrial en constante cambio. A medida que las condiciones y patrones de operación evolucionan, el modelo puede adaptarse a los nuevos datos de manera continua, asegurando que su capacidad para identificar actividades de riesgo se mantenga al día. Este aspecto dinámico es fundamental en la detección de eventos en tiempo real, ya que un modelo que se ajusta de forma incremental a nuevos datos y mejora la respuesta frente a posibles riesgos en el entorno de trabajo [66] [67]. La capacidad del Transfer Learning para generalizar a nuevas situaciones y adaptarse a cambios incrementales resulta en una mejora considerable en la seguridad y eficiencia operativa. Al integrar datos recientes, los modelos pueden detectar patrones emergentes de riesgo que no estaban presentes en el conjunto de entrenamiento original, lo que contribuye a una respuesta proactiva en lugar de reactiva [65].

Además, el Transfer Learning puede contribuir a una mejor generalización del modelo a nuevos datos y situaciones. Los modelos pre-entrenados tienden a haber aprendido características valiosas y genéricas a partir de una variedad de tareas, lo que facilita que el modelo se adapte y funcione bien en la detección de eventos de riesgo para grúas horquillas, incluso en condiciones no presentes en el conjunto de entrenamiento. El abordar un problema en el contexto industrial con un enfoque de Transfer Learning permite una actualización continua del modelo a medida que se obtienen más datos específicos que se pueden ir generando durante la misma operación del sistema. Esto garantiza que el sistema esté al tanto de las condiciones cambiantes y los eventos de riesgo, lo que contribuye a una mayor precisión y relevancia en la detección [66] [67].

En el estado del arte, podemos observar diferentes técnicas para utilizar implementar el Transfer Learning, dentro los cuales destacan [68] [69] [70]:

- Feature Transfer: La Transferencia de Características implica tomar las características extraídas por un modelo pre-entrenado en una tarea y utilizar esas características como entrada para un nuevo modelo diseñado específicamente para la tarea objetivo. Esta técnica es eficiente en términos de recursos, ya que solo se transfieren las características relevantes, lo que puede ser útil cuando se tiene un conjunto de datos limitado. Sin embargo, las características pre-entrenadas pueden no ser perfectamente adecuadas para la nueva tarea, lo que puede requerir ajustes adicionales. La Transferencia de Características es adecuada cuando se busca una inicialización efectiva para un modelo personalizado. Por ejemplo, se puede

utilizar una red neuronal convolucional (CNN) pre-entrenada en clasificación de imágenes y transferir sus características para la detección de objetos en grúas horquillas.

- Full Model Transfer: En este enfoque, se utiliza un modelo pre-entrenado en su totalidad como punto de partida para la nueva tarea sin modificar su arquitectura. Esto puede ser eficaz cuando la arquitectura del modelo pre-entrenado se ajusta bien a la nueva tarea y se busca una transferencia completa de conocimiento y arquitectura. Sin embargo, puede requerir una gran cantidad de datos de entrenamiento específicos y recursos computacionales considerables. La Transferencia del Modelo Completo es útil cuando se necesita una adaptación precisa y completa de un modelo. Por ejemplo, se puede tomar un modelo pre-entrenado en inglés y transferirlo a una tarea de traducción de texto en español.
- Fine-tuning: El Ajuste Fino implica tomar un modelo pre-entrenado y ajustar sus pesos y parámetros específicamente para la nueva tarea. Esto permite una adaptación precisa y es eficiente en términos de recursos cuando se dispone de suficientes datos de entrenamiento. Sin embargo, requiere una cantidad significativa de datos de entrenamiento para la tarea objetivo y un ajuste cuidadoso de hiperparámetros. El Ajuste Fino es útil cuando se necesita una adaptación precisa y se cuenta con suficientes datos. Por ejemplo, se puede tomar una red neuronal pre-entrenada en reconocimiento de voz y ajustarla finamente para la identificación de comandos de voz específicos de grúas horquillas.

- Zero-shot Learning: El Aprendizaje de Cero Disparos permite que un modelo generalice a nuevas clases o tareas sin haber sido entrenado específicamente en ellas. Es valioso en escenarios donde no se tienen datos de entrenamiento para todas las clases o tareas. Permite una adaptación a tareas nuevas sin necesidad de reentrenamiento completo. Sin embargo, puede requerir modelos y técnicas avanzadas, como modelos generativos o métodos basados en atributos, lo que puede ser más complejo de implementar. El Aprendizaje de Cero Disparos es adecuado cuando se necesita adaptarse a nuevas clases o eventos de riesgo sin datos de entrenamiento específicos. Por ejemplo, se puede utilizar para reconocer un nuevo tipo de evento de riesgo en grúas horquillas sin haberlo entrenado previamente.

El Transfer Learning demuestra su eficacia en diversos dominios mediante la utilización de arquitecturas pre-entrenadas. En el ámbito de la visión por computadora, ResNet y VGG son ampliamente aplicadas. Por ejemplo, adaptar una ResNet pre-entrenada para la detección de anomalías en imágenes médicas permite identificar patologías en radiografías. En procesamiento de lenguaje natural, BERT y GPT son versátiles, como el caso de ajustar BERT para clasificación de texto en el ámbito legal. Además, en la detección de fraudes financieros, se emplean modelos pre-entrenados, como WaveNet, para identificar patrones anómalos en transacciones a lo largo del tiempo [71] [72] [73].

Estos ejemplos subrayan la versatilidad del Transfer Learning al utilizar arquitecturas pre-entrenadas y adaptarlas para tareas específicas en diversos

dominios, mejorando la eficiencia y rendimiento de los modelos en situaciones con conjuntos de datos limitados. El Transfer Learning desempeña un papel crucial en la detección de actividades de riesgo, permitiendo que modelos pre-entrenados aporten conocimientos valiosos a nuevas tareas, incluso en presencia de diferencias significativas entre dominios de datos. Frente a este desafío de transferencia de dominio, se sugiere emplear métodos de adaptación, como la adaptación adversarial, que ajusta el modelo para desempeñarse bien en el nuevo dominio. Además, estrategias de fine-tuning en capas específicas, especialmente las finales, abordan disparidades en dominios y mejoran la capacidad de generalización [74].

La optimización de hiperparámetros y la validación cruzada son prácticas esenciales para el éxito del Transfer Learning. La optimización ajusta la configuración del modelo para adaptarse mejor a la nueva tarea, y la validación cruzada evalúa su capacidad de generalización en diversos conjuntos de datos. Estudios respaldan la eficacia de estas prácticas, enfatizando su importancia en el rendimiento del modelo [75].

La adaptabilidad continua del modelo es esencial para su efectividad. El Transfer Learning permite actualizaciones sin entrenamiento completo desde cero, adaptándose a nuevos datos específicos. La capacidad de aprendizaje incremental juega un papel crucial, permitiendo la incorporación progresiva de nuevos datos. Sin embargo, se requiere monitoreo constante para identificar cambios significativos y evitar la obsolescencia del modelo [76].

El Transfer Learning enfrenta desafíos importantes en la transferencia de dominio, especialmente cuando existen discrepancias significativas entre los datos de entrenamiento y los datos de aplicación real. Sin embargo, el uso de técnicas como la adaptación adversarial, que ajusta el modelo para que pueda reconocer patrones relevantes en datos de dominio nuevo, y el fine-tuning, que permite la actualización específica de ciertos parámetros del modelo para la nueva tarea, han demostrado ser altamente efectivos en mejorar la capacidad de generalización y rendimiento del modelo. Estas técnicas permiten que el modelo mantenga la precisión en escenarios donde los datos de entrenamiento pueden ser limitados o representar condiciones distintas a las encontradas en la práctica.

Además, prácticas como la optimización de hiperparámetros y la validación cruzada son esenciales para garantizar un ajuste adecuado del modelo a la tarea específica. La optimización de hiperparámetros ayuda a encontrar la configuración óptima que maximiza el rendimiento del modelo, mientras que la validación cruzada asegura que el modelo sea capaz de generalizar adecuadamente a distintos subconjuntos de datos, reduciendo el riesgo de sobreajuste y mejorando la robustez.

La adaptabilidad continua del modelo es un aspecto crucial en entornos dinámicos, donde las condiciones de operación pueden cambiar con frecuencia. No obstante, esta adaptabilidad requiere un monitoreo constante y ajustes periódicos para evitar la obsolescencia, especialmente cuando los datos o patrones cambian significativamente con el tiempo. Esto puede incluir la implementación de mecanismos de aprendizaje incremental, donde el modelo se

actualiza regularmente con nuevos datos sin necesidad de reentrenamiento completo, lo cual optimiza los recursos y mantiene la relevancia del modelo en el tiempo [77] [78] [79].

1.5 Caso de aplicación: Grúa Horquillas

En la industria existen diferentes tipos de maquinaria, dentro de la categoría principal se encuentran los “equipos críticos”. Estos equipos se caracterizan por el fuerte impacto que su detención tiene en la producción. Por esta razón la importancia de monitorearlos y optimizar sus detenciones, para así disminuir al máximo las detenciones no programadas. Bajo este contexto el monitoreo de actividades de “riesgo” es sumamente relevante, ya que permite disminuir la probabilidad de un accidente y por ende de una falla aleatoria. Los sistemas de monitoreo comerciales tienen un nivel de madurez tecnológico alto en caracterización de equipos en operaciones estacionarias [80]. Sin embargo, los equipos en operaciones no estacionarias, es decir, cuyos datos se distribuyen con una probabilidad que varía de forma no constante, solo tiene soluciones altamente especializadas. Cabe destacar que los procesos no estacionarios pueden convertirse en procesos estocásticos estacionarios mediante la transformación de serie [81]. Debido a esto, monitorear equipos como las grúas horquillas es una tarea compleja, ya que además de tener un modo de trabajo nominal que puede ir variando por cada operador, se desenvuelven en ambientes con alta variabilidad.

Las grúas horquillas son maquinaria móvil utilizadas para transportar, levantar o descender objetos sin esfuerzo del operador, y funcionan a través de un sistema hidráulico que permite elevar y descender cargas a una gran altura, el equipo cuenta con un contrapeso que le permite realizar las actividades mencionadas. La horquilla es un equipo fundamental en el sector industrial en áreas de

almacenaje y centros de distribución, ya que debido a las características de: (i) Capacidad de carga, (ii) Tamaño, (iii) Fácil operatividad y mantenimiento, (iv) Bajo nivel de contaminación acústica, (v) Puede operar en espacios cerrados como abiertos. En el mercado existen diferentes tipos de grúas, una de las principales características para clasificarlas es según el tipo de energía que utiliza para realizar su operación, ya que puede variar desde gasolina, gas hasta electricidad. Si bien, independiente de la fuente de energía que ocupe en su funcionamiento todas cumplen las mismas tareas (apilar, arrastrar, trasladar, cargar y descargar objetos), su diferencia radica en su desempeño en diferentes entornos de operación, dentro de las que se destacan: (i) Grúas horquillas a gasolina se utilizan cuando la actividad se realiza en ambientes abiertos con suficiente espacio para ejecutar las maniobras y donde las emisiones que genera el equipo no representan un riesgo para la salud de los operadores, (ii) Grúas horquillas a gas licuado se utilizan cuando la actividad se realiza en ambientes cerrados debido a que tienen una baja emisión (iii) Grúas horquillas a diésel suelen utilizarse para ambientes en exteriores con carga de gran volumen que requiere ser elevada, trasladada o arrastrada. También se les denomina grúas horquillas 4x4 debido a que son ideales para el trabajo en terrenos irregulares, con pendientes, rocas o alta humedad. (iv) Grúas horquillas eléctricas pueden utilizar baterías de plomo ácido o de litio. La principal diferencia entre ambas es que la primera es económica y dura aproximadamente 12 horas en operación.

Por otro lado, las grúas con baterías de Litio son más costosas, utilizan sistemas de carga rápida y la batería es de libre mantenimiento [82]. La figura 4 muestra los componentes principales de las grúas horquillas.

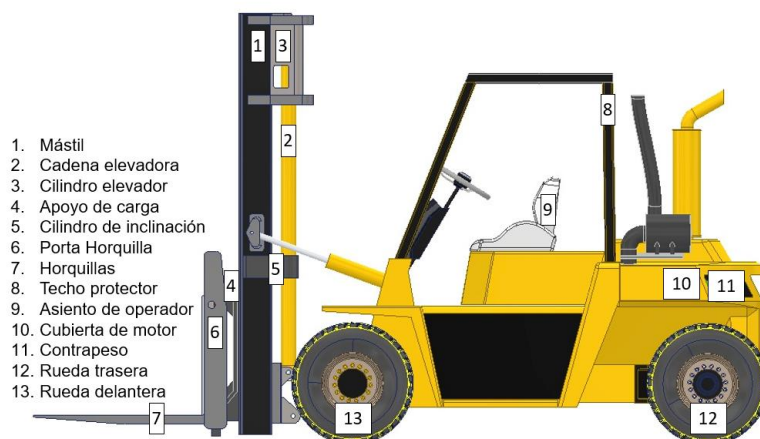


Fig. 4 Componentes de grúa horquilla.

Respecto a las investigaciones y estudios realizados respecto a grúas horquillas se muestra una sistematización de estas en la Tabla 4.

Tabla 4 Investigaciones de grúas horquillas.

Titulo de estudio	Resumen	Principal aporte
Testing of data analysis process in hydraulic Forklift [83].	<ul style="list-style-type: none"> Análisis de datos en una grúa horquilla con situaciones de fallas autoinfligidas, se utilizaron técnicas de aprendizaje no supervisado para agrupar los datos. Dentro de estas fallas destacan la fuga interna del cilindro de elevación y retrasos en las válvulas. Los datos son recopilados por el sistema de monitoreo de la misma grúa. 	<ul style="list-style-type: none"> Reporta que la agrupación de los datos tiene correlación con los datos agrupados según sus modos de fallas. Sin embargo, no se entrega una métrica cuantitativa que permita evaluar de forma objetiva la problemática analizada.
IoT Based Forklift Realtime Monitoring System Development [84]	<ul style="list-style-type: none"> Desarrollo de sistema de monitoreo de condiciones basado en IoT. El sistema toma las variables de la ECU (Unidad de control de motor), y lo envía a través Bluetooth 4.0 a una base de datos SQLite a un teléfono móvil. 	<ul style="list-style-type: none"> Se logran ahorros de combustible de alrededor del 30 % en comparación con las grúas no instrumentadas. Con el monitoreo en tiempo real se minimiza el riesgo de accidentes mediante el control y la supervisión de las principales funciones de operación.
Forklift safety, traffic engineering and intelligent transport systems: a case study [85]	<ul style="list-style-type: none"> Se realiza una intervención en 2 fábricas. En la primera fábrica se opera de forma normal bajo la supervisión de un experto. En la segunda fábrica se utilizan protocolos de seguridad delimitando las zonas de operación de las grúas con respecto a las de los trabajadores. Se comparan ambos resultados y se analiza el impacto de la intervención de la segunda fábrica respecto a la disminución de 	<ul style="list-style-type: none"> Al realizar la Inter versión se reportó una disminución de hasta 4 veces el riesgo en términos de gravedad nominal de las interacciones entre grúa horquilla y peatones.

	los accidentes asociados a la operación de la grúa.	
VR-Based investigation of forklift operator situation awareness for preventing collision accidents [86]	<ul style="list-style-type: none"> • Utilizan un entorno de realidad virtual (VR) como entorno experimental, en el que los sujetos realizan una serie de subtareas, como conducir, girar, dar marcha atrás, cargar y descargar, con un modelo de simulación de grúa horquilla VR 	<ul style="list-style-type: none"> • Al realizar ciertas tareas consideradas de riesgo previamente en un entorno de realidad virtual se puede aumentar hasta en un 30% la correcta respuesta del operador ante ciertas situaciones se conciencia situacional SA.
A Situationally Aware Voice-commandable Robotic Forklift Working Alongside People in Unstructured Outdoor Environments [87]	<ul style="list-style-type: none"> • Se desarrolló una grúa horquilla manejada por comando de voz que permite operar junto con personas y otros vehículos, manipulando materiales paletizados dentro de las instalaciones de almacenamiento y al aire libre. El sistema tiene un sistema de aprendizaje de tareas a través del procesamiento de imágenes con algoritmos de aprendizaje automático. activas existentes. 	<ul style="list-style-type: none"> • Se introducen modelos de aprendizaje automático con visión computacional en la utilización de la grúa Horquilla. Además, se evalúa el funcionamiento "autónomo" en un entorno con operadores y otras máquinas.
Development of automated monitoring safety system of the forklift DP 3510 [88]	<ul style="list-style-type: none"> • Se desarrolla un modelo de evaluación de seguridad instrumental a través del monitoreo con un sistema de software y hardware de Omron. El sistema monitorea variables sobrecarga y presión de bomba de aceite. 	<ul style="list-style-type: none"> • Se presenta el modelo de evaluación que funciona estimando la diferencia entre los valores máximos (de carga y presión de la bomba del aceite) de una operación nominal y los datos adquiridos por el sistema de monitoreo.

Las investigaciones expuestas en la Tabla 4 tienen como foco en el centro la correcta operación de las grúas horquillas. Es por esto que trabajos de agencias como Administración de Seguridad y Salud Ocupacional (OSHA), son sumamente relevantes al entregar normas internacionales de la correcta operación y de clasificación de actividades de riesgos. La Osha 3949 especifica la correcta operación y actividades de riesgos que pueden ocurrir durante el proceso de operación de las grúas horquillas. La norma especifica ciertos aspectos de seguridad que no tienen directa relación con el equipo tales como si el operador utiliza sus equipos de protección personal, que tenga la correcta licencia y/o capacitación para la operación del equipo, entre otros. Sin embargo, la norma especifica una serie de situaciones de riesgos, que corresponden a actividades que si se ejecutan aumenta la probabilidad de un accidente, el cual

puede tener resultados de daño para la grúa, la carga, el operador u otro trabajador que se encuentre en el espacio de trabajo.

Dentro de las actividades de riesgo que menciona la norma esta: (i) Mover en cargas de gran altura con carga desequilibrada, (ii) Bajar pendientes de frente con la grúa cargada, (iii) Subir o bajar la carga mientras se transporta, (iv) Transportar una carga alta, (v) Conducir en pendientes con una inclinación superior a 15°, (vi) Subir a operadores en las horquillas, (vii) Incumplimiento del área de trabajo definida para transitar, (viii) Girar en pendientes, (ix) transportar sobre carga en las horquillas.

1.6 Discusión General

Respecto a la revisión bibliográfica realizada en los tópicos, se encuentran oportunidades de investigación dentro de las que destacan:

- En las investigaciones analizadas existe una clara disociación entre las investigaciones que se dedican a los procesos generados de datos con las investigaciones de generación de modelos a través del aprendizaje automático. Por ende, en el estado del arte no se encuentran propuestas metodológicas para abordar esta problemática de forma integral.
- En el estado del arte no se encuentra reporte metodológico para realizar evaluaciones de algoritmos de detección de eventos de riesgo con set variado de videos y ejecutados en arquitectura Edge.
- Las investigaciones indican que la detección de actividades de riesgo para las grúas horquillas es fundamental para la seguridad ocupacional y para evitar fallas imprevistas. Aun así, los trabajos presentan sistemas de

monitoreo que ocupan como principal fuente generadora de datos la ECU, siendo totalmente insuficiente para la detección de actividades de riesgo. Por otro lado, ninguna de las investigaciones estudiadas utilizaba la totalidad de eventos de riesgo que presenta la OSHA 3949, por lo que se presentaban soluciones parciales.

- Para cualquier sistema de monitoreo de actividades de riesgo, el tiempo de detección y de generación de alarma es clave, sin embargo, este parámetro nunca se evaluó en los sistemas de monitoreo de la Tabla 4. Considerando este elemento, se debiese optar por arquitecturas Edge o Cloud-Edge ya que tienen una respuesta más rápida al momento de realizar las notificaciones.

En el escenario de aplicaciones industriales, el cual se caracteriza por una escasez de datos etiquetados y con dificultades en recopilar grandes conjuntos de datos el Transfer Learning ofrece una solución al permitirnos iniciar con modelos pre-entrenados en datos más generales, lo que reduce el proceso de entrenamiento, lo cual es una ventaja crucial en entornos industriales donde la velocidad de respuesta es esencial, especialmente en la detección de eventos de riesgo en tiempo real. Además, la generalización es esencial en entornos industriales dinámicos y variados. En el estado del arte, se ha demostrado que estos modelos poseen una notable capacidad para generalizar a nuevas situaciones, gracias a su habilidad para aprender representaciones útiles en distintos contextos. Esto los hace más adaptables y resistentes a la variabilidad

característica de los entornos industriales, reduciendo significativamente la dependencia de grandes volúmenes de datos industriales recopilados desde cero. Feature Transfer se presenta como una de las mejores opciones en situaciones donde el tiempo de entrenamiento debe ser acotado y los datos son escasos. Esto se debe a que permite tomar un modelo pre-entrenado y ajustar sus pesos y parámetros específicamente para la nueva tarea, sin necesidad de entrenar todo el modelo desde cero. Por lo que acorta el tiempo de entrenamiento y tiene un eficiente desempeño con datos escasos.

CAPITULO 2. Objetivos y plan de trabajo

2.1 Introducción

Este trabajo diseña algoritmos para la detección de actividades de “riesgo” en un caso de aplicación de grúa horquilla. En este capítulo se definirá la hipótesis, objetivo general, objetivos específicos, alcances, limitaciones, metodología, plan de trabajo y cronograma de este proyecto de investigación.

2.2 Hipótesis

El uso de modelos de Deep Learning y Transfer Learning permite construir un modelo robusto y generalizado para la detección de ‘eventos de riesgo’ en grúas horquillas mediante una cantidad limitada de videos, aprovechando modelos previamente entrenados en tareas similares y adaptándolos al contexto específico de identificación de eventos de riesgo.

2.3 Objetivo general.

Desarrollar, implementar y evaluar un algoritmo de detección de eventos de riesgo efectivo y generalizable para aplicaciones en grúas horquillas utilizando técnicas de Deep Learning y Transfer Learning a partir de un numero limitados de datos.

2.4 Objetivos específicos

Los objetivos específicos son cuatro:

- 1. Recopilación y Preprocesamiento de Datos:** Este objetivo específico se alinea con la implementación de conjuntos de datos etiquetados para el entrenamiento, testeo y validación de los modelos. Implica la adquisición y preparación de datos de videos de grúas horquillas, garantizando la calidad

y la homogeneidad de los datos, así como la etiquetación precisa de eventos de riesgo.

2. **Desarrollo de Modelos de Deep Learning Supervisado:** En este objetivo específico, se busca diseñar e implementar algoritmos de Deep Learning supervisado para la clasificación de eventos de riesgo y operación "normal" según la norma OSHA 3949. Esto implica la construcción de modelos de clasificación basados en Deep Learning.
3. **Implementación de Transfer Learning:** Este objetivo específico corresponde a la implementación de la metodología de reentrenamiento a través de Transfer Learning en los algoritmos de Deep Learning. Aquí se adapta y ajusta la arquitectura de modelos pre-entrenados a la tarea específica de detección de eventos de riesgo en grúas horquillas.
4. **Evaluación y Validación de Modelos:** Finalmente, este objetivo se relaciona con el análisis y la evaluación de los algoritmos basados en Transfer Learning y Deep Learning. Implica la medición del rendimiento de los modelos desarrollados en términos de su capacidad para detectar eventos de riesgo en videos de grúas horquillas, utilizando métricas de evaluación apropiadas.

2.5 Limitaciones

Los algoritmos de detección de eventos peligrosos se entrenarán con datos etiquetados según la norma OSHA 3949. Por lo que actividades de riesgos que estén fuera de esta norma quedan fuera del alcance de la presente investigación.

Cabe señalar que el evento de riesgo “Llevar sobrecarga en la grúa horquilla” no se puede realizar debido a que es una actividad indetectable en video.

El desempeño del algoritmo propuesto se va evaluar respecto a 2 plataformas estandarizadas, AWS Reckognition y DeepStream.

2.6 Metodología

En este trabajo de investigación se propone la implementación de un algoritmo de detección de eventos de riesgo para grúas horquillas utilizando la adquisición videos y clasificadores basado en Deep Learning. Debido a la revisión del estado del arte reportado en grúas horquillas, el enfoque de la estrategia metodológica de esta investigación será de carácter experimental, exploratorio y cuantitativo. El proceso de desarrollo considera los objetivos planteados en esta propuesta, que permite establecer una herramienta adaptativa y basado en las condiciones de operación para la detección de evento de riesgo. La metodología se divide en 2 fases, la primera se enfoca en diseñar y validar el algoritmo diseñado para la detección de eventos de riesgo.

La segunda etapa es comparar el diseño del algoritmo con herramientas o frameworks estandarizados.

La figura 5 muestra la fase del diagrama metodológico, la cual se compone de 3 etapas y 3 condiciones, que se detalla a continuación:

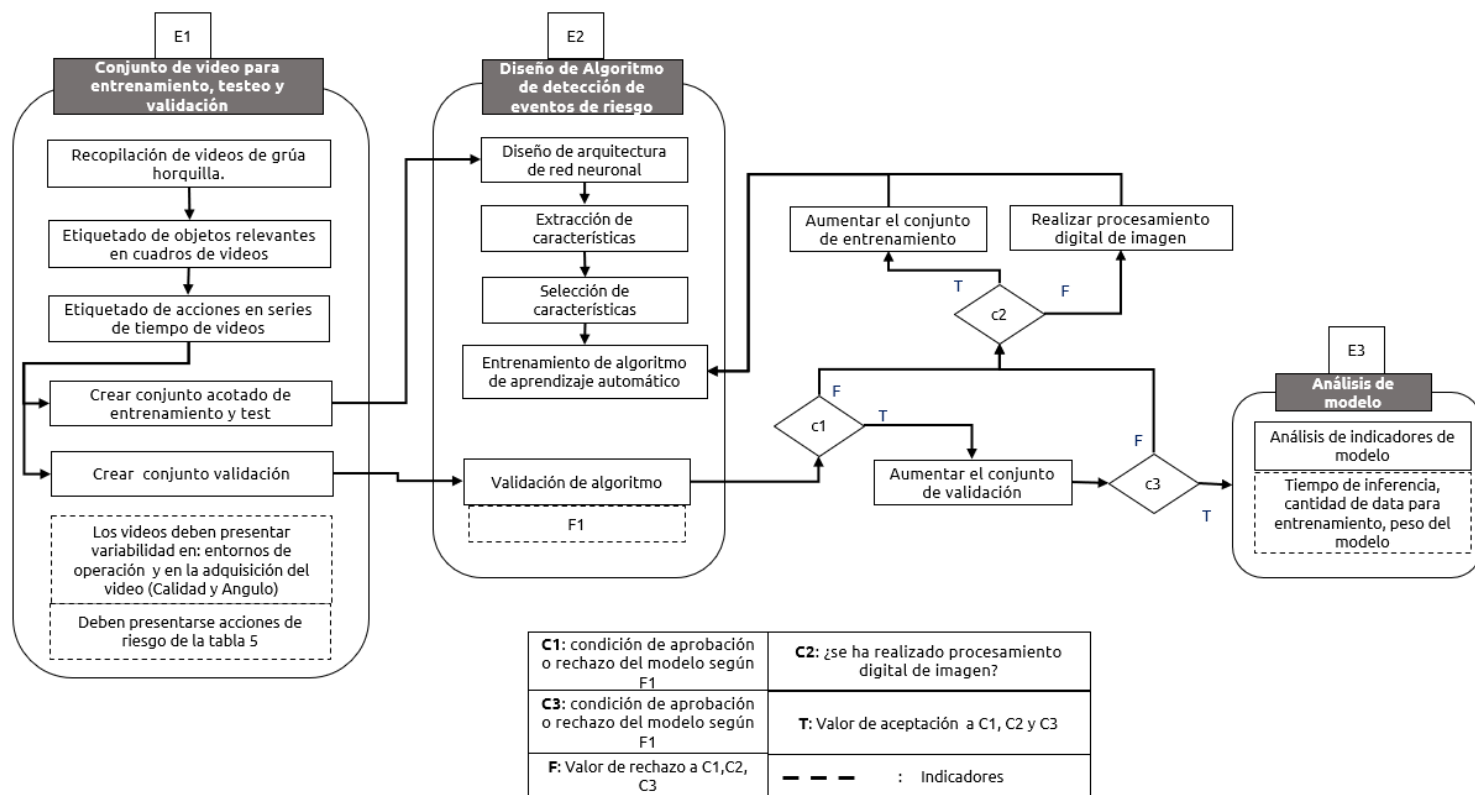


Fig. 5 Metodología.

- (E1) *Conjunto de video para entrenamiento, testeo y validación*: consiste en recopilar un conjunto de videos que se utilizan en las etapas de entrenamiento, test y validación. Los videos de grúas horquillas que se recopilan tienen diferentes ambientes y condiciones de operaciones. Si bien las características de los ambientes de operación pueden ser variados, para la presente investigación se consideran dos. El primero es el lugar de operación de la grúa horquilla. Esta característica es sumamente relevante debido a que la luminosidad es un factor crítico al momento del entrenamiento de los algoritmos de Deep Learning para la detección de eventos de riesgo basado en imágenes. Dentro de los posibles ambientes de operación destacan: (i) Exterior de día, (ii) Exterior

de noche, (iii) Interior con iluminación artificial. El segundo ambiente de condición relevante tiene relación con el proceso de adquisición de datos. La recopilación de los videos debe ser variada en la calidad de los videos (resolución) y ángulos de adquisición.

Respecto a las condiciones de operación es sumamente relevante que los videos contengan las diferentes actividades que el algoritmo quiere identificar. Para definir los eventos que los videos deben mostrar se utiliza la OSAH 3949. Cabe señalar que los eventos de "Cargar sobre carga" no se puede detectar con algoritmos basados en videos, por lo que este evento queda fuera de los alcances del algoritmo propuesto. Por otro lado, es muy importante que el aprendizaje automático entrene la definición de "funcionamiento normal". OSHA 3949 define muchas operaciones como "normales" o "ejecutadas correctamente". En los videos recolectados la definición de operación normal se limita a la correcta manipulación de la carga sobre un palet. Esta definición se resume en los siguientes puntos:

- (i) Posicionar el montacargas frente a la tarima y fijar la altura e inclinación de las horquillas, (ii) enganchar las horquillas a la tarima, (iii) elevar la tarima cargada a una altura de 10 a 20 cm en el suelo, (iv) desplazar la grúa por las zonas de delimitación hasta el punto de entrega de la carga, (v) bajar la carga en el suelo, retroceder y desenganchar las horquillas del pallet. La Tabla 5 muestra un resumen de los eventos que el algoritmo propuesto debe detectar.

Tabla 5 Eventos para detectar.

N°	Eventos a detectar
1	Operación normal
2	Mover en cargas de gran altura con carga desequilibrada.
3	Bajar pendientes de frente con la grúa cargada.
4	Subir o bajar la carga mientras se transporta.
5	Transportar una carga alta.
6	Conducir en pendientes con una inclinación superior a 15°.
7	Subir a operadores en las horquillas.
8	Incumplimiento del área de trabajo definida para transitar.
9	Girar en pendientes

Una vez recopilados todos los videos, se procede a etiquetar. Se realizan 2 diferentes tipos de etiquetado. El primer etiquetado se basa en la identificación de objetos “relevantes” que se encuentran en los videos. Debido a que a priori no se conoce cuáles son los “objetos relevantes”, es que se etiquetan los componentes más relevantes de la grúa horquilla (ver figura 4) y algunos del ambiente de operación (operador, personas en el lugar de trabajo, delimitación de lugar de trabajo y carga).

El segundo etiquetado es la detección del “evento de riesgo”, para esto se selecciona una sección de video donde la grúa realiza uno de los eventos de la tabla 5. Luego del proceso de etiquetado se generan grupos para el entrenamiento, testeo y validación del algoritmo. Cabe destacar que cada grupo tendrá todos los eventos a detectar, pero se subdividen en los diferentes ambientes de operación. Esto tiene como objetivo poder determinar la cantidad y variabilidad de datos que necesita el algoritmo para poder “generalizar”.

- (E2) *Diseño de algoritmo de detección de evento de riesgo*: El algoritmo de reconocimiento de eventos, se basa en una *staging* de *Deep Learning*. El desempeño de la red diseñada se medirá con el F1-score (F1 en adelante). El F1 es una medida comúnmente utilizada en Machine Learning para evaluar la precisión de un modelo en términos de la proporción de verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN) en un conjunto de datos. Es una combinación de la precisión y el *recall* del modelo, tal como muestra la ecuación (1) [89].

$$F1_{score} = 2 \frac{(precision)(recall)}{(precision + recall)} \quad (1)$$

Donde la precisión se define como la proporción de verdaderos positivos respecto al total de predicciones positivas (VP + FP), y el *recall* se define como la proporción de verdaderos positivos respecto al total de casos positivos (VP + FN). El F1 es un buen indicador porque es una métrica balanceada que considera tanto la precisión como el *recall*, lo que significa que tiene en cuenta tanto los falsos positivos como los falsos negativos.

Una alta precisión significa que el modelo es muy bueno en predecir positivos reales y minimiza los falsos positivos, mientras que un alto *recall* significa que el modelo es muy bueno en detectar todos los casos positivos, incluso si hay falsos positivos. Por lo tanto, el F1 proporciona una evaluación más completa del modelo que simplemente mirar la precisión o el *recall* por separado [90] [91].

- (C) *Cumplimiento de condición*: En el diagrama metodológico se pueden ver 3 instancias donde se pregunta el cumplimiento de una condición (C1, C2 y C3). C1 consulta si la validación del modelo se cumple, es decir, si el modelo en la etapa de validación tiene F1 mayor a 0.85. Si se cumple la condición de aceptación se aumenta el conjunto de validación y se vuelve la entrada de la instancia de pregunta C3. Si no se cumple la condición de aceptación vuelve la entrada de la instancia de pregunta C2. C2 consulta si el conjunto de imágenes con el que el algoritmo fue entrenada ha tenido al procesamiento digital de imagen. Si no se cumple esta condición, se debe modificar con PDI el set de entrenamiento, debido a que esto puede ayudar en [92]: (i) Mejorar la calidad de las imágenes con un pre-procesamiento tales como aplicar técnicas de filtrado para reducir el ruido en las imágenes, la normalización para ajustar el contraste y la iluminación, y la segmentación para separar los objetos de interés del fondo, (ii) Extraer características de las imágenes que son relevantes para el modelo de aprendizaje profundo. Por ejemplo, se pueden utilizar técnicas de detección de bordes, detección de características y clasificación para extraer información importante de las imágenes. Estas características pueden utilizarse como entrada para el modelo de aprendizaje profundo y pueden mejorar su capacidad para identificar y clasificar objetos en las imágenes, (iii) Aumento de datos, utilizando técnicas de rotación, cambio de tamaño y distorsión para

generar variaciones de las imágenes originales. Estas imágenes generadas pueden utilizarse para aumentar el tamaño del conjunto de datos y mejorar el rendimiento del modelo de Deep Learning.

Si se cumple la condición de C2, se aumenta el conjunto de entrenamiento y se vuelve al “Entrenamiento de algoritmo” de E2. C3 consulta si la validación del modelo se cumple, es decir, si el modelo en la etapa de validación tiene F1 mayor a 0.85, luego de haber aumentado el conjunto de datos de validación.

- (E3) *Análisis de modelo*: Se realiza un análisis para tener una descripción acabada del modelo. Para esto además de utilizar el F1 como métricas de iteres, también se suma: (i) La cantidad de cuadros con el que el modelo debió ser entrenado para llegar a ser “generalizado”, (ii) El peso del modelo, (iii) el tiempo en que se demora en realizar la inferencia en un sistema embebido.

Por otro lado, al realizar el análisis del modelo y caracterizar su desempeño (como se muestra en E3), queda la interrogante, ¿otras herramientas estandarizadas pueden tener un desempeño similar. Ante esta interrogante es que se decide utilizar 2 distintos *frameworks* que utilizan Deep Learning, para comparar el algoritmo propuesto. La figura 6, muestra el diagrama metodológico de cómo se compara el algoritmo diseñado con las 2 distintas herramientas estandarizadas. Y se termina en la etapa metodológica 4 (o E4), en la cual se

aprueba o rechaza la pregunta de investigación planteada. Los 2 *frameworks* a utilizar son:

- DeepStream SDK es un conjunto de herramientas de desarrollo de software creado por NVIDIA que permite a los desarrolladores crear aplicaciones de análisis de video de alta velocidad y baja latencia utilizando hardware acelerado por GPU. DeepStream SDK funciona mediante la creación de un *pipeline* de procesamiento de video que consta de varios módulos de procesamiento que se ejecutan en paralelo. Cada módulo de procesamiento se encarga de una tarea específica, tales como: (i) recopilación de fuente de datos de video, (ii) pre-procesamiento se encarga de realizar tareas como la conversión de formato de video, la escala de imagen y el recorte de video, (iii) inferencia es responsable de ejecutar modelos de aprendizaje profundo para detectar, segmentar o clasificar objetos en el video, (iv) post-procesamiento se encarga de realizar tareas como la eliminación de falsos positivos, la eliminación de objetos redundantes y la agrupación de objetos, (v) salidas de video en diferentes formatos tales como pantallas de visualización, archivos de video y flujos de video en vivo [93].
- Amazon Rekognition Custom Labels es un servicio de AWS que permite entrenar modelos de aprendizaje profundo personalizados para reconocer objetos y escenas específicas en imágenes y videos. El servicio funciona en tres pasos principales: (i) se entrenamiento del modelo con un conjunto de imágenes etiquetadas que representan los objetos o actividades que

desean detectar en los videos, (ii) evalúa en un conjunto de datos separado del entrenamiento para determinar su precisión y calidad. Si el modelo no alcanza un nivel de precisión aceptable, el desarrollador puede ajustar el modelo y volver a entrenarlo (se utiliza el F1 como principal métrica de evaluación), (iii) utilización para el reconocimiento de objetos y actividades específicas en imágenes y videos, la utilización se hace en la nube y se puede interconectar con otros dispositivos a través de APIs [94].

En resumen, para llevar a cabo la metodología que nos permite responder nuestra pregunta de investigación, podemos simplificarlos en los pasos de la figura 6. En la cual se muestra que el primer paso es la recopilación de videos, etiquetado y generación del banco de datos. El detalle de esta sección de los tipos de videos, el cómo se etiquetan y agrupan se muestra con detalle en el capítulo llamado “Conjunto de video para entrenamiento, testeo y validación.”. Luego se diseña el Algoritmo propuesto basado en Deep Learning para la detección de los eventos de riesgo. Para evaluar y comparar el resultado del AP se utilizan 2 *framework*. Es importante seleccionar la arquitectura correcta en ambos *frameworks*, que nos permitan tener buenos resultados en la detección de actividades de riesgo. Para eso en el caso de AWS se debe seleccionar y configurar una arquitectura Cloud que permite tener, almacenamiento de datos, computo para la utilización de los algoritmos, y sistema de comunicación para reportar los resultados. Para NVIDIA *DeepStream*, es más sencillo ya que su configuración depende de la tarjeta de video donde este siendo implementada.

El detalle de los algoritmos se muestra con detalle en el capítulo llamado “Diseño de modelos”.

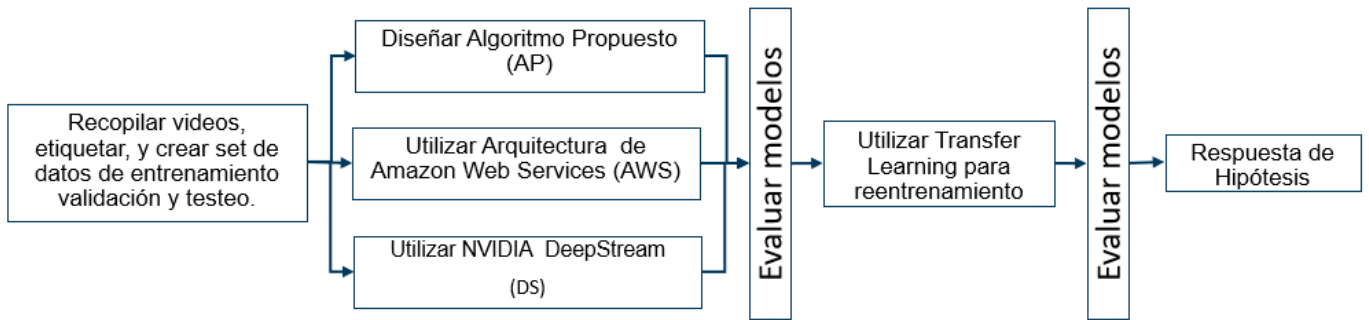


Fig. 6 Flujo de modelos a utilizar.

CAPITULO 3. Diseño de modelos

Para que cualquier algoritmo basado en Deep Learning, tenga éxito en su operación (ya sea de regresión o clasificación), es extremadamente relevante que el conjunto de datos con el que el modelo es entrenado sea variado y de un volumen “considerable”. Por esta razón, es que la presente investigación, además de generar un conjunto de videos en base a fuentes de información de terceros, también incorpora información generada en un entorno controlado en banco de pruebas.

El banco de pruebas diseñado tiene como objetivo replicar las actividades de interés a detectar (que se muestran en la tabla 5). El banco de pruebas tiene las siguientes características: (i) Área delimitada en el suelo como área de operación, (ii) Tarimas con diferentes cargas, (iii) Una rampa o área con grados de inclinación y (iv) Sistema de adquisición de video. La figura 7 muestra la configuración del banco de pruebas, que tiene 3 puntos. El primer punto (P1) es donde comienza la carga de la grúa. El segundo punto (P2) es donde se deja la carga. El tercer punto (P3) tramo con inclinación, donde se realizan maniobras con carga. La figura 8 muestra la implementación del banco de pruebas. Se utilizan 3 distintas cámaras.

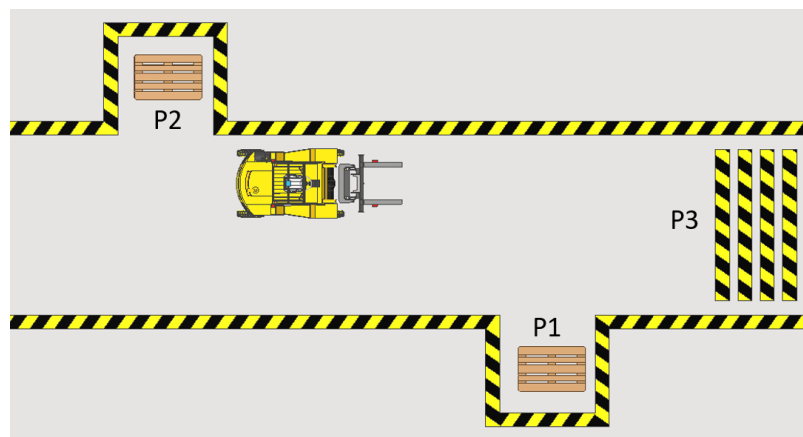


Fig. 7 Plano del entorno del banco de pruebas.



Fig. 8 Entorno de banco de pruebas.

Del banco de ensayo se logra obtener información sobre todos los eventos de interés (ver tabla 5) en un ambiente exterior de buena iluminación. Por otro lado, también se realiza una búsqueda de diferentes videos de operación de grúas horquillas en diferentes entornos, con diferentes ángulos (en la captura de video) y diferente iluminación, la tabla 7 muestra una descripción de todos los videos.

Tabla 6. Cantidad y descripción de videos por tipo de operación.

Tipo de operación	N° de videos Totales	N° video en ambiente exterior	N° video en ambiente Interior	Otras características de videos
Normal	15	10	5	La mayoría de los videos que representan una correcta utilización de la grúa horquilla, son videos de capacitaciones técnicas. Tanto para ambientes interiores como exteriores tienen una excelente luminosidad.
Mover en cargas de gran altura con carga desequilibrada.	12	8	4	Estos videos corresponden a grabaciones de cámaras de seguridad a una resolución promedio de 420x594. Las grabaciones no tienen grados de libertad y suelen tener una vista isométrica.
Bajar pendientes de frente con la grúa cargada.	11	8	3	Estos videos corresponden a grabaciones de cámaras de seguridad a una resolución promedio de 420x594. Las grabaciones no tienen grados de libertad y suelen tener una vista isométrica.
Subir o bajar la carga mientras se transporta.	15	8	7	Estos videos tienen dos principales fuentes de datos. La primera corresponde a grabaciones de videos de seguridad y la segunda a grabaciones realizadas desde celulares. Estas últimas no tienen una perspectiva fija, suelen seguir a la grúa y contienen movimientos (ya que no están estabilizados)
Transportar una carga alta.	15	8	7	Estos videos son los que se encuentran en formatos más variados. Pueden ser capturados por videos de capacitaciones, cámaras de seguridad o por cámaras de celulares de trabajadores que están el mismo lugar. Por lo que la variedad en vistas, luminosidad y calidad del video es muy variable.
Conducir en pendientes con una inclinación superior a 15°.	10	5	5	Estos videos tienen dos principales fuentes de datos. La primera son videos de capacitaciones técnicas y la segunda grabación de cámaras de seguridad.
Subir a operadores en las horquillas.	15	5	10	Estos videos tienen dos principales fuentes de datos. La primera corresponde a grabaciones de videos de seguridad y la segunda a grabaciones realizadas desde celulares. Estas últimas no tienen una perspectiva fija, suelen seguir a la grúa y contienen movimientos (ya que no están estabilizados)
Incumplimiento del área de trabajo definida para transitar.	11	6	5	Estos videos tienen dos principales fuentes de datos. La primera son videos de capacitaciones técnicas y la segunda grabación de cámaras de seguridad. Cabe destacar que el "área definida de trabajo", varía entre los videos. Ya que se utilizan cono se seguridad, cinta de seguridad o una mezcla de ambas.
Girar en pendientes	11	6	5	Estos videos corresponden a grabaciones de cámaras de seguridad a una resolución promedio de 420x594. Las grabaciones no tienen grados de libertad y suelen tener una vista isométrica.

A continuación, se muestran ejemplos de ciertos eventos en los escenarios que contempla el conjunto de datos utilizados.



Fig. 9 Escenario 1, correspondiente a banco de ensayo en un evento de riesgo (mover en cargas de gran altura con carga desequilibrada).



Fig. 10 Escenario 2, realizando una operación normal.



Fig. 11 Escenario 4, realizando un evento de riesgo (subir a operadores en las horquillas).



Fig. 12 Escenario 3, realizando un evento de riesgo (mover en cargas en altura).



Fig. 13 Escenario 5, realizando un evento de riesgo (bajar pendientes de frente con la grúa cargada.)



Fig. 14 Escenario 6, realizando un evento de riesgo (conducir en pendientes con una inclinación superior a 15°)



Fig. 15 Escenario 7, realizando un evento de riesgo (bajar la carga mientras se transporta).

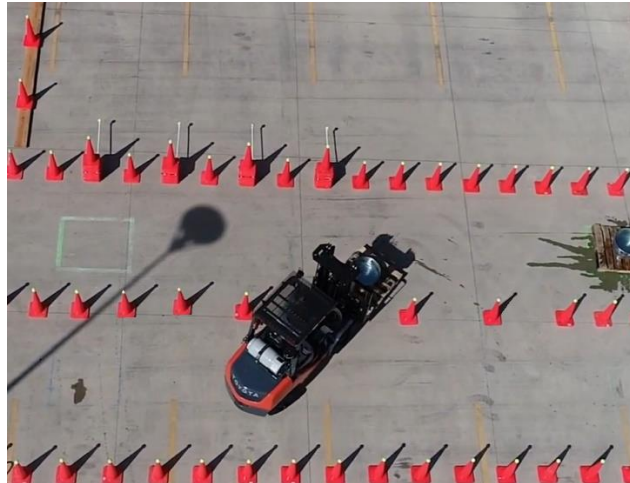


Fig. 16 Escenario 8, realizando un evento de riesgo (incumplimiento del área de trabajo definida para transitar.).

3.1 Etiquetado de datos

El algoritmo de reconocimiento de eventos de riesgo diseñado se subdivide en dos clasificaciones. El primer algoritmo de clasificación realiza el reconocimiento de objetos, como la grúa horquilla, carga, horquillas, cinta delimitadora de seguridad entre otros. Este algoritmo requiere segmentar el video en cuadros. Para esto se dividen los videos en cuadros, en cada cuadro se enmarca el objeto de interés con una etiqueta, para el proceso de transformación de video a cuadros se usa un pipeline de *GStreamer* [95]. Para esto se utiliza una herramienta de anotación de imágenes gráficas, la cual está programada en Python y Qt para su interfaz gráfica, llamada *Labellmg* [96]. Esta herramienta nos genera 2 salidas, la imagen etiquetada y un archivo json en el cual se especifica la posición de cada marco con su respectiva etiqueta. La figura 17 muestra la interfaz *Labellmg*.

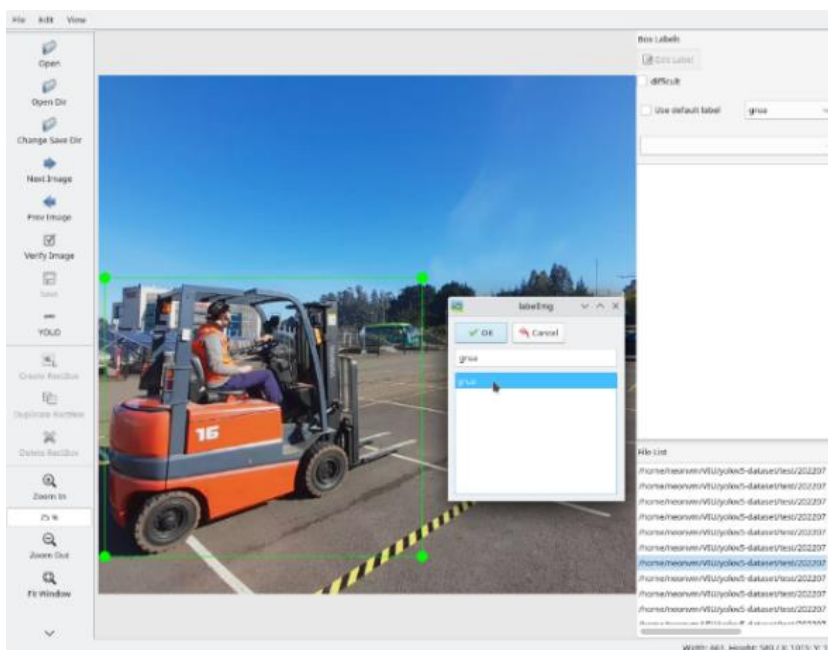


Fig. 17 Labellmg UI.

El segundo algoritmo de clasificación toma el objeto y clasifica el evento de riesgo que está realizando. Para ello, se genera un conjunto de entrenamiento compuesto por secciones de video. Por lo que los *frames* anteriormente etiquetados se renderizan para generar un video con etiquetas de objetos (en la misma imagen). Luego lo videos renderizados son categorizados según la acción que se ejecuta. Cabe destacar que los videos utilizados tienen de 60 hasta 120 *fps* y con una calidad de *full hd* (1080p). Sin embargo, para alivianar la carga computacional en la etapa de procesamiento de video, es que se reducen todos los videos a 15 *fps* y a una calidad de *hd* (720p).

Para las herramientas estandarizadas presentada en la metodología de evaluación de modelos de Deep Learning de *DeepStream SDK* se utilizan secciones de videos con la etiqueta de las acciones y no de los objetos.

A pesar de muchas veces tener un conjunto de datos variados, los modelos nos exigen una mayor cantidad de datos para poder tener un buen desempeño. Para aumentar los datos para el entrenamiento, se utilizan técnicas de *procesamiento de imágenes digitales* para generar imágenes "sintéticas" con ruido en los objetos de interés y actividades. Generar imágenes sintéticas es una técnica fundamental en el ámbito del aprendizaje profundo, ya que proporciona una serie de ventajas significativas para mejorar el proceso de entrenamiento de una red neuronal. Uno de los beneficios clave radica en el aumento de datos disponibles para el entrenamiento por lo que la red neuronal puede aprender de una mayor variedad de ejemplos y permite que la red neuronal se vuelva más versátil y capaz de reconocer patrones en una variedad de situaciones donde los

escenarios pueden variar considerablemente. Por otro lado, la generación de imágenes sintéticas también puede ayudar a abordar desafíos como el desequilibrio de clases en el conjunto de datos, lo cual es muy común en datos reales, debido a que las situaciones de riesgo ocurren en mucho menor medida que las operaciones normales. Entre los procesos realizados para generar imágenes sintéticas se encuentran (i) la rotación de la imagen, (ii) la adición de ruido de sal y pimienta a la imagen, (iii) el cambio de la saturación de la imagen y (iv) la generación de collages de imágenes con objetos de interés.

La figura 19 muestra un conjunto de imágenes sintéticas generadas por una imagen del banco de ensayo.



Fig. 19 Imágenes sintéticas generadas con PDI.

Para entrenar modelos de Deep Learning es fundamental conocer con precisión la cantidad de imágenes etiquetadas por cada uno de los estados que se quiere clasificar. Esto es especialmente importante en el contexto del aprendizaje supervisado, donde el modelo se entrena para aprender a reconocer patrones y características en los datos. Las etiquetas proporcionan la "supervisión" necesaria al indicar al modelo qué características debe buscar y

cómo debe categorizar los datos y evita el sobreajuste. Además, la cantidad adecuada de imágenes etiquetadas contribuye a la precisión del modelo cuando la clasificación es multi-estado. Al tomar los videos expuestos en la Tabla 6, y etiquetarlos procedemos a ordenar por categoría la cantidad de *frames* etiquetados por evento.

Tabla 7 Porcentaje de datos por tipo de evento para clasificador multiclase

Evento	Datos por evento
Operación normal	15 %
Mover en cargas de gran altura con carga desequilibrada.	5 %
Bajar pendientes de frente con la grúa cargada.	10 %
Subir o bajar la carga mientras se transporta.	15 %
Transportar una carga alta.	15 %
Conducir en pendientes con una inclinación superior a 15°.	15 %
Subir a operadores en las horquillas.	15%
Incumplimiento del área de trabajo definida para transitar.	5%
Girar en pendientes	5%

3.2 Algoritmo Propuesto

Para reconocer eventos de riesgo, se utiliza un apilamiento de dos redes convolucionales. La primera red tiene como objetivo identificar los objetos de interés (grúa horquilla, carga y cinta de seguridad delimitando el espacio de trabajo).

La entrada al modelo son cuadros de video capturados a 15 fps. Cada fotograma se introduce como una imagen RGB. Por lo que el modelo tiene como entradas las imágenes captadas por 3 cámaras RGB, es decir una entrada de 9 canales. El primer algoritmo de clasificación consiste en un modelo CNN que consta de cuatro capas convolucionales y una capa completamente conectada. Las capas convolucionales tienen un número decreciente de filtros de 128 a 8 y luego a 4; De manera similar, el tamaño del *kernel* también disminuye en las capas, comenzando con dos (5,5), luego (3,3) y finalmente (1,1). Esta arquitectura permite que la red aprenda patrones más específicos a medida que los datos pasan a través de las capas. Cada capa convolucional tiene una función de activación ReLU y es seguida por un bloque Squeeze and excitar. Las salidas de cada filtro describen un mapa de características que será procesado por el bloque de compresión y excitación. Este bloque descarta o enfatiza información de cada canal del mapa de características. Estos bloques comienzan con una normalización por lotes y una agrupación promedio para la reducción de la dimensionalidad. Luego, los mapas de características se multiplican por un conjunto de sus características recuperadas de un perceptrón de dos capas totalmente conectados. Las funciones de activación son ReLU y sigmoid, respectivamente. Los bloques Squeeze & Excite modelan dependencias

dinámicas no lineales entre canales, lo que reduce la complejidad del modelo. Esto permite mantener un alto rendimiento de clasificación usando no más de 3 capas convolucionales [97]. Después de estas tres capas, la clasificación se realiza mediante una capa densa totalmente conectada de 128 neuronas con activación ReLU y una capa de salida con activación SoftMax. Una representación de esta arquitectura se muestra en la figura 20.A

El segundo algoritmo de clasificación combina capas CNN y LSTM. El diseño propuesto combina la información de ambos algoritmos concatenando los datos recuperados por las redes trabajando en paralelo. Las arquitecturas de red CNN y LSTM se simplifican para evitar obtener un modelo excesivamente complejo. La red convolucional tiene solo dos capas con sus respectivos bloques Squeeze & excitar y el mapa de características de salida se aplana antes de la concatenación. Las capas LSTM necesitan una remodelación en la entrada porque las imágenes de señal vienen como datos 2d. Por lo tanto, esta rama se inicia con una transformación de remodelación. Luego, una sola capa LSTM procesa esta información utilizando una técnica de abandono, lo que reduce las posibilidades de sobreajuste. Este modelo híbrido es como otras arquitecturas que muestran un buen rendimiento para la clasificación de series temporales [11]. Finalmente, las salidas de ambas ramas se concatenan y pasan por una capa de salida con una función de activación SoftMax. Una representación de esta arquitectura se muestra en la figura 20.B

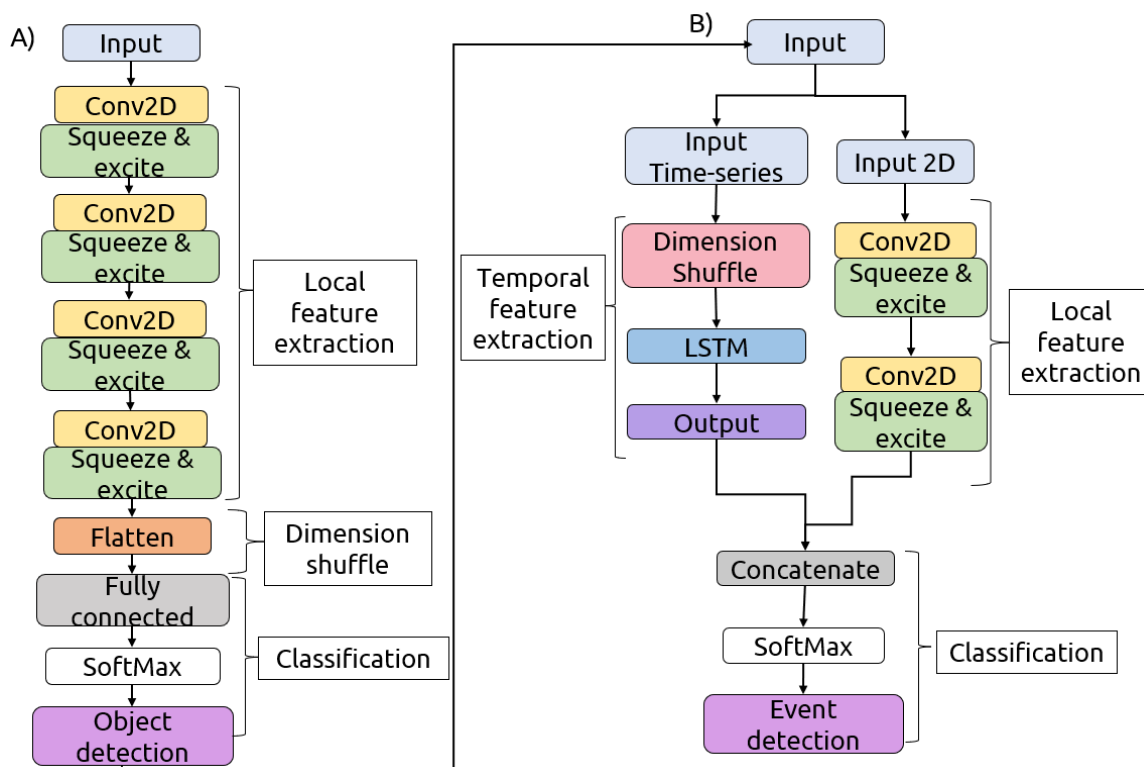


Fig 20. Algoritmo de detección de eventos de riesgo basado en video.

El conjunto de datos de entrenamiento del algoritmo 20.A consiste en *frames* de videos etiquetados con los objetos de interés (chasis de grúa horquilla, horquillas, carga) a una resolución de 15 *fps*. La salida del algoritmo 10.A corresponde a dos archivos, el primero un *JSON*, en el cual se especifica el *frame* con la ubicación de los objetos detectados y el indicado F1, y el segundo *frame* con el objeto de interés etiquetado en la imagen. Utilizando los *frames* de salida del algoritmo 20.A se genera un video que se utiliza como la entrada del algoritmo 20.B. Como el algoritmo 20.B consta de dos partes. Una red LSTM y una CNN, las cuales difieren levemente en su tipo de entrada. La primera tiene una entrada de videos de 1 segundo de operación compuesto por 15 *frames* y la segunda solo tiene un *frame* de la misma ventana. La salida del algoritmo 20.B corresponde a

dos archivos, el primero un o JSON, en el cual se especifica el *frame* con el evento de riesgo detectado y su indicador F1. El segundo archivo de salida corresponde a los cuadros con la etiqueta del reconocimiento de la acción.

Como el algoritmo propuesto es un *staging* entre 2 redes neuronales (la primera que detecta el objeto y la segunda que detecta el evento), es sumamente importante sintonizar los modelos cada uno por si solo y luego en conjunto, debido a que una degradación del F1-score en la primera red podría afectar la detección del evento de riesgo.

3.3 Algoritmo con Amazon Rekognition Custom Labels

Amazon Rekognition Custom Labels permite entrenar rápidamente sus propios modelos personalizados para detectar objetos y actividades de escenas específicos. La principal ventaja de esta herramienta es que se basa en modelos pre-entrenados existente en la herramienta *Amazon Rekognition*, que ha sido entrenada con decenas de millones de imágenes de muchas categorías y por usuarios de todo el mundo. Por lo que permite entrenar modelos con un conjunto más limitado de imágenes, pero que deben ser específicas para el caso de uso.

La figura 21 muestra la arquitectura de la solución, este flujo de trabajo permite generar un modelo de detección de comportamientos específicos en un vídeo está automatizado, con un único parámetro de entrada el cual corresponde a un archivo de vídeo. El flujo de trabajo contiene los siguientes pasos: Se carga un archivo de vídeo en *Amazon Simple Storage Service*, que invoca a *AWS Lambda*, que a su vez llama a un punto final de inferencia de etiquetas personalizadas de *Amazon Rekognition* y a *Amazon Simple Queue Service*. *Amazon SQS* que utiliza una función *Lambda* para realizar una comprobación del estado del punto de enlace de inferencia y lanza *Amazon Elastic Compute Cloud (Amazon EC2)* si el estado es *Running*. *Amazon CloudWatch Events* detecta el estado *Running* de *Amazon EC2* e invoca una función *Lambda*, que ejecuta un *script* (personalizado) en *Amazon EC2* mediante la ejecución de *AWS Systems Manager*. En *Amazon EC2*, el *script* llama a la inferencia de *Amazon Rekognition Custom Labels* para detectar comportamientos específicos en el vídeo cargado en *Amazon S3* y escribe los resultados inferidos en el vídeo en *Amazon S3*. Cuando el archivo de

resultados inferidos se carga en *Amazon S3*, se lanza una función *Lambda* para detener *Amazon EC2* y a la inferencia de *Amazon Rekognition Custom Labels*. Cabe destacar, que para utilizar estas herramientas es fundamental realizar la configuración de *AWS CloudFormation*. El cual es un servicio de infraestructura como código que permite automatizar el proceso de aprovisionamiento de recursos y la gestión de la infraestructura en la nube de *AWS*. *CloudFormation* facilita la creación, la actualización y la eliminación de los recursos de *AWS*, lo que ayuda a reducir el tiempo y los costos asociados con la configuración manual de la infraestructura [98].

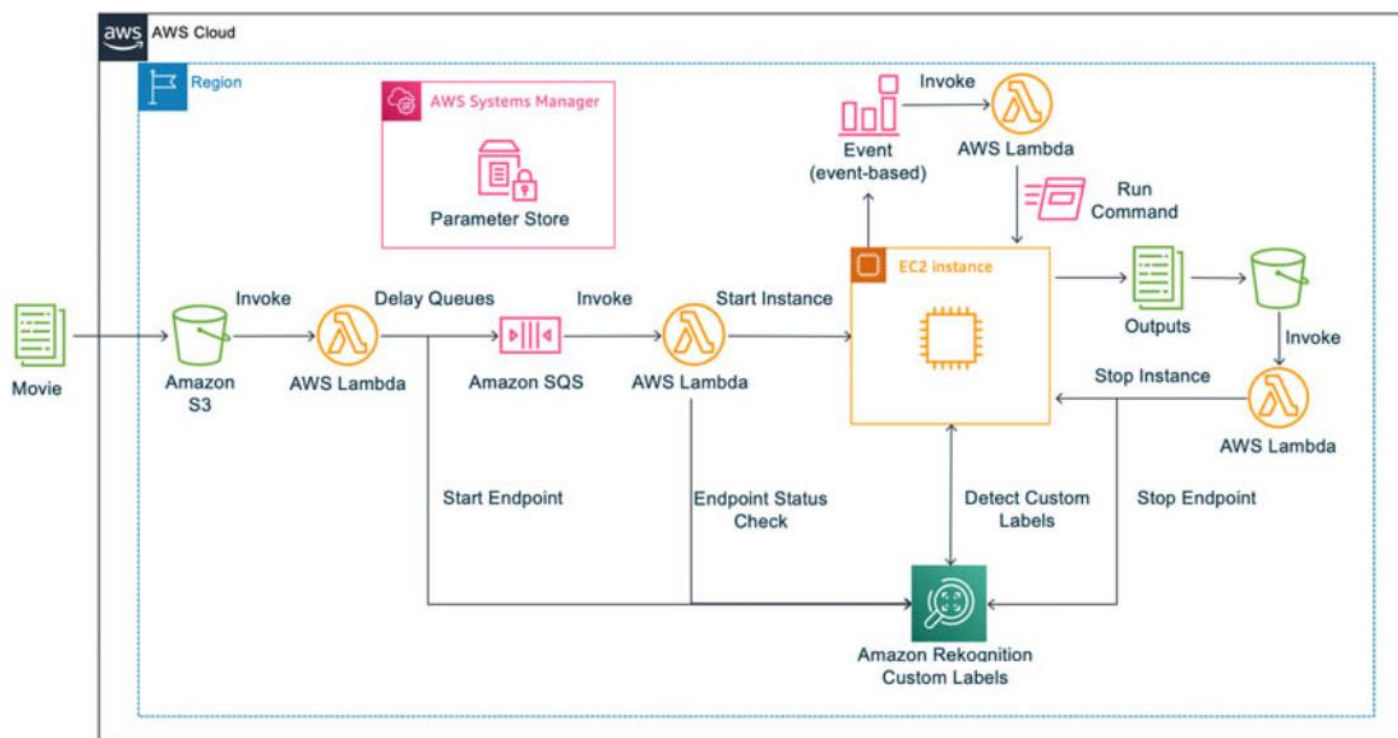


Fig 21. Arquitectura de algoritmo en AWS.

Cabe destacar que a pesar de que la herramienta es estandarizada, en las instrucciones se hace referencia a un script personalizado en cual hace un llamado secuencial de ciertas funciones y configura la salida de nuestro modelo.

El flujo de trabajo del script es el siguiente:

- Se importan las librerías necesarias.
- Se define una función "get_parameters" que recibe un parámetro "param_key" y utiliza la biblioteca Boto3 para obtener un valor de un parámetro almacenado en AWS Systems Manager (SSM).
- Se define una función "analyzeVideo" que utiliza las bibliotecas Boto3, *OpenCV* y *FFmpeg* para procesar un video.
- Se llama a la función "get_parameters" para obtener el nombre del bucket de S3 y el nombre del archivo de video a procesar.
- Se descarga el archivo de video del *bucket de S3* y se crea un objeto de recurso de S3.
- Se inicializan variables para capturar el ancho, la altura y la velocidad de fotogramas del video.
- Se inicializa un objeto de escritura de video con los parámetros del video de entrada y se define un nombre de archivo de salida.
- Se inicia un bucle *while* que recorre todos los fotogramas del video.
- Se obtiene cada fotograma del video y se convierte a un objeto de imagen utilizando la biblioteca *OpenCV*.
- Se llama al servicio *Rekognition de AWS* para detectar etiquetas personalizadas en el fotograma de la imagen y se procesan los resultados.

- Se actualizan las etiquetas personalizadas y se almacenan en una lista.
- Se dibujan rectángulos y etiquetas en el fotograma de la imagen utilizando la biblioteca *OpenCV* según los resultados de la detección de etiquetas personalizadas.
- Se escribe el fotograma procesado en el archivo de salida de video.
- Se guarda la lista de etiquetas personalizadas en un archivo *JSON* y se carga en un *bucket de S3*.
- Se crea una transcodificación del archivo de salida de video en formato MP4 y se carga en un *bucket de S3*.
- Se libera el objeto de escritura de video y se cierra el objeto de captura de video.

3.4 Nvidia DeepStream SDK

Para la implementación NVIDIA *DeepStream SDK* se utilizó como hardware una tarjeta de evaluación *Jetson AGX*, debido a que este hardware permite realizar tanto entrenamiento como inferencia en videos en tiempo real. La primera etapa de la metodología implica el almacenar los videos de forma local (debido a que este servicio es 100% *edge*). Los videos se deben almacenar etiquetados y servirán como conjunto de datos para entrenar y validar el modelo de detección de actividades de riesgo. Para el entrenamiento del modelo se emplea un modelo de Deep Learning, como una CNN, para la detección y clasificación de actividades de riesgo. Se utiliza el conjunto de datos de entrenamiento etiquetado para entrenar el modelo. La arquitectura de la CNN se ajusta y se entrena utilizando el NVIDIA *DeepStream SDK*, que aprovecha la aceleración de hardware de la *Jetson AGX* para un procesamiento eficiente en tiempo real. Para la utilización de la Aplicación con *DeepStream SDK*, se utiliza el flujo recomendado del fabricante [99]. La cual cuenta con modelos de detección y clasificación previamente entrenados. Para realizar la validación y evaluación, el mismo flujo recomendado nos entrega herramientas para utilizar videos de prueba que no se utilizaron en el entrenamiento. Finalmente, *DeepStream SDK* nos retorna un archivo en el formato *json* en el cual aparece cada *frames* con el estado de clasificación asignado por el algoritmo. Adicionalmente se puede hacer una renderización del video en el cual se le adiciona el estado de operación (esto

se puede hacer en tiempo real o posteriormente).



Fig 22. Arquitectura de algoritmo en DeepStream.

3.5 Evaluación de modelos

Para evaluar el desempeño de la red diseñado se realizaron los siguientes pasos:

1. Se realizar un entrenamiento de clasificación binaria para el algoritmo propuesto, y para ambos *frameworks*, es decir, estos modelos solo tendrán la capacidad de clasificar los videos en “Operación Normal” y en “Operación de Riesgo”. El entrenamiento de estos modelos solo se utilizarán datos de un escenario (solo de un entorno).
2. Luego de obtener el indicador F1-score se estimará el F1-score realizando inferencia de un estado conocido, pero del escenario 2. Esto nos permitirá cuantificar que tan sobre ajustado están los modelos, antes de realizar el transfer Learning.
3. Utilizando los modelos de clasificación binaria se utiliza la técnica de significancia con labores *shap* en el algoritmo propuesto para identificar cuál de los objetos identificados aportan más al modelo de detección de eventos de riesgo.
 4. Se realizan los pasos 1 y 2 pero con una clasificación multi-estado que se muestra en la Tabla 5.

La Tabla 8 muestra la cantidad de datos según su etiqueta que se usaron para el entrenamiento, testeo y validación de los algoritmos, utilizando el 70% , 20% y 10% de los datos respectivamente.

Tabla 8 Cantidad de datos según evento, para entrenamiento de algoritmos

Clasificador binario		Clasificador multiclase	
Evento	Datos por evento	Evento	Datos por evento
Evento normal	60 %	Operación normal	15 %
Evento de riesgo	40 %	Mover en cargas de gran altura con carga desequilibrada.	5 %
		Bajar pendientes de frente con la grúa cargada.	10 %
		Subir o bajar la carga mientras se transporta.	15 %
		Transportar una carga alta.	15 %
		Conducir en pendientes con una inclinación superior a 15°.	15 %
		Subir a operadores en las horquillas.	15 %
		Incumplimiento del área de trabajo definida para transitar.	5%
		Girar en pendientes	5%

El primer análisis se realiza entrenando los algoritmos de detección de eventos de riesgo para clasificar de forma binaria. La Tabla 9 compara la clasificación usando el algoritmo diseñado con la herramienta estandarizada de *Amazon Rekognition Custom Labels*. La tabla muestra F1 del rendimiento como indicador, debido a que se utiliza el mismo conjunto de datos para evaluar ambos modelos, por lo cual la comparación del F1 se considera significativa. Además, se compara el entrenamiento que no utilizan los datos sintéticos con los que si los utilizan. Análogamente la Tabla 10 muestra la misma comparación de parámetros, pero con la clasificación multi-estado.

Tabla 9 Resultados de clasificación binaria.

	Algoritmo diseñado sin datos sintéticos	Algoritmo diseñado con datos sintéticos	Amazon Rekognition Custom Labels sin datos sintéticos	Amazon Rekognition Custom Labels con datos sintéticos
	F1	F1	F1	F1
Operación normal	0.902	0.952	0.951	0.975
Evento de riesgo	0.871	0.93	0.92	0.942

Tabla 10 Resultados de clasificación multi-clase

	Algoritmo diseñado sin datos sintéticos	Algoritmo diseñado con datos sintéticos	Amazon Rekognition Custom Labels sin datos sintéticos	Amazon Rekognition Custom Labels con datos sintéticos
	F1	F1	F1	F1
Operación normal	0.901	0.924	0.949	0.961
Mover en cargas de gran altura con carga desequilibrada.	0.911	0.925	0.951	0.959
Bajar pendientes de frente con la grúa cargada.	0.903	0.934	0.951	0.948
Subir o bajar la carga mientras se transporta.	0.889	0.921	0.963	0.966
Transportar una carga alta.	0.91	0.922	0.941	0.939
Conducir en pendientes con una inclinación superior a 15°.	0.853	0.897	0.901	0.905
Subir a operadores en las horquillas.	0.915	0.961	0.974	0.978
Incumplimiento del área de trabajo definida para transitar.	0.878	0.903	0.941	0.939
Girar en pendientes	0.813	0.85	0.891	0.895

Comparando los F1 de la clasificación binaria del algoritmo diseñado sin datos sintéticos respecto al que, si utilizo datos sintéticos, se obtiene una mejora aproximada de un 3% en el F1, para ambos estados. En cambio, al comparar el desempeño del F1 con el algoritmo basado en *Amazon Rekognition Custom Labels* con datos sintéticos respecto al sin datos sintéticos, se observa una variación menor a 0.3%. Esta variación se considera dentro de la banda de ruido del algoritmo. Cabe destacar que con utilizando el conjunto de entrenamiento del banco de ensayo el algoritmo basado en *Amazon Rekognition Custom Labels* obtiene mejores resultados que el algoritmo diseñado para la clasificación binaria

como para la multi estado. En la clasificación multi-estado el algoritmo diseñado con *Amazon Rekognition Custom Labels* tiene un coeficiente de variación de 2.9% en el F1 entre los eventos, respecto a el coeficiente de variación de 3.33% del algoritmo diseñado. Por otro lado si tomamos el valor F1-score de la clasificación binaria del “Evento de riesgo” y lo comparamos con el promedio de los eventos de riesgo de la clasificación multi-estado para el algoritmo propuesto solo se tiene una mejora del 1.6% y para el algoritmo de AWS la diferencia se encuentra dentro del margen de ruido (variación menor al 0.1%).

Cabe destacar que utilizando el conjunto de entrenamiento del banco de ensayo el algoritmo basado en *Amazon Rekognition Custom Labels* se obtienen mejores resultados que con el algoritmo diseñado, tanto para la clasificación binaria como para la multi-estado. Sin embargo, un buen resultado con un conjunto de datos de un mismo ambiente, no nos permite evaluar los modelos de forma integral, ya que no se puede asegurar que sean generalizables. Es por esto que con los modelos entrenados (con datos sintéticos) se procede a realizar una validación con el conjunto de datos del “escenario 2”, como se muestra en las Tablas 11 y 12.

Tabla 11 Resultados de clasificación binaria entrenado en escenario 1 y realizando inferencia sobre datos de escenario 2

	Algoritmo diseñado con datos sintéticos	Amazon Rekognition Custom Labels con datos sintéticos
	F1	F1
Operación normal	0,632	0,655
Evento de riesgo	0,591	0,602

Tabla 12 Resultados de clasificación multi-evento entrenado en escenario 1 y realizando inferencia sobre datos de escenario 2

	Algoritmo diseñado con datos sintéticos	Amazon Rekognition Custom Labels con datos sintéticos
	F1	F1
Operación normal	0,629	0,589
Mover en cargas de gran altura con carga desequilibrada,	0,527	0,519
Bajar pendientes de frente con la grúa cargada,	0,548	0,558
Subir o bajar la carga mientras se transporta,	0,591	0,572
Transportar una carga alta,	0,621	0,582
Conducir en pendientes con una inclinación superior a 15°,	0,512	0,506
Subir a operadores en las horquillas,	0,634	0,604
Incumplimiento del área de trabajo definida para transitar,	0,557	0,625
Girar en pendientes	0,512	0,585

Observando los resultados de la Tabla 11 y 12 se aprecia que ambos modelos bajaron su desempeño para realizar la detección de eventos de riesgo con un video el cual nunca fue parte de su set de entrenamiento y que cambia en sus características ambientales, tales como el color de la grúa horquilla, la luminosidad del área de trabajo, el cambio en el ángulo de la cámara, la carga, la forma de operación entre otras.

Análogamente se repite el mismo análisis para comparar el Algoritmo Propuesto vs *DeepStream SDK*. La Tabla 13 y Tabla 14 muestra la comparación del F1-score entre ambos algoritmos. Se puede observar que el desempeño entre el algoritmo propuesto y el *DeepStream SDK* sin datos sintéticos es similar (con un leve mejor desempeño el *DeepStream SDK* tanto para la clasificación binaria como para la multi-clase. Sin embargo, al utilizar datos sintéticos el desempeño del *DeepStream SDK* , mejora en un 7% para la clasificación binaria y en un 4% para la clasificación multi-clase. Al igual que la comparación anterior, estos resultados no nos permiten evaluar los modelos de forma integral, ya que no se puede asegurar que sean generalizables. Es por esto que con los modelos

entrenados (con datos sintéticos) se procede a realizar una validación con el conjunto de datos del “escenario 2”, como se muestra en la Tabla 15 y 16.

Tabla 13 Resultados de clasificación binaria.

	Algoritmo diseñado sin datos sintéticos	Algoritmo diseñado con datos sintéticos	DeepStream SDK sin datos sintéticos	DeepStream SDK con datos sintéticos
	F1	F1	F1	F1
Operación normal	0.902	0.952	0.909	0.971
Evento de riesgo	0.871	0.93	0.881	0.951

Tabla 14 Resultados de clasificación multi-clase

	Algoritmo diseñado sin datos sintéticos	Algoritmo diseñado con datos sintéticos	DeepStream SDK sin datos sintéticos	DeepStream SDK con datos sintéticos
	F1	F1	F1	F1
Operación normal	0.901	0.924	0.912	0.966
Mover en cargas de gran altura con carga desequilibrada.	0.911	0.925	0.908	0.961
Bajar pendientes de frente con la grúa cargada.	0.903	0.934	0.909	0.951
Subir o bajar la carga mientras se transporta.	0.889	0.921	0.912	0.970
Transportar una carga alta.	0.91	0.922	0.903	0.945
Conducir en pendientes con una inclinación superior a 15°.	0.853	0.897	0.882	0.917
Subir a operadores en las horquillas.	0.915	0.961	0.913	0.972
Incumplimiento del área de trabajo definida para transitar.	0.878	0.903	0.893	0.945
Girar en pendientes	0.813	0.85	0.853	0.912

Tabla 15 Resultados de clasificación binaria con test en escenario 2

	Algoritmo diseñado con datos sintéticos	DeepStream SDK con datos sintéticos
	F1	F1
Operación normal	0,632	0,544
Evento de riesgo	0,591	0,531

Tabla 16 Resultados de clasificación multi-clase con test en escenario

	Algoritmo diseñado con datos sintéticos	DeepStream SDK con datos sintéticos
	F1	F1
Operación normal	0,629	0,519
Mover en cargas de gran altura con carga desequilibrada,	0,527	0,509
Bajar pendientes de frente con la grúa cargada,	0,548	0,542
Subir o bajar la carga mientras se transporta,	0,591	0,569
Transportar una carga alta,	0,621	0,571
Conducir en pendientes con una inclinación superior a 15°,	0,512	0,510
Subir a operadores en las horquillas,	0,634	0,593
Incumplimiento del área de trabajo definida para transitar,	0,557	0,561
Girar en pendientes	0,512	0,509

3.6 Discusión de evaluación de modelos

Los resultados presentados en las Tablas 9 a 16 ofrecen una visión detallada del desempeño de diferentes algoritmos de detección de eventos de riesgo en grúas horquillas en diversas configuraciones y condiciones. En este análisis, se compararon tres enfoques: Algoritmo Propuesto, *Amazon Rekognition Custom Labels* y *DeepStream SDK*. Estas tres herramientas se probaron con y sin datos sintéticos y con dos tipos de clasificaciones binaria y multi-clase.

Por un lado, se observa que el uso de datos sintéticos en el entrenamiento del algoritmo diseñado llevó a una mejora significativa en el rendimiento, con un aumento de aproximadamente un 3% en el valor F1 tanto en la clasificación binaria como en la multi-estado. Esto indica que la inclusión de datos sintéticos en el proceso de entrenamiento puede ser beneficioso para mejorar la capacidad de detección y clasificación de eventos de riesgo.

Sin embargo, al comparar el algoritmo diseñado con el enfoque de *Amazon Rekognition Custom Labels*, se observa que este último obtiene resultados similares, con una variación menor al 0.3% en el valor F1 en ambas configuraciones, incluso sin el uso de datos sintéticos. Esto sugiere que *Amazon Rekognition Custom Labels* puede proporcionar un desempeño comparable al algoritmo diseñado en esta tarea específica.

Por otro lado, en un análisis más profundo, se destaca un desafío crítico en el entrenamiento de algoritmos de detección de eventos de riesgo la generalización. A pesar de las mejoras logradas mediante el uso de datos sintético, los modelos aún enfrentan dificultades para generalizar en situaciones no vistas previamente. El sobreajuste se hace evidente al comparar el rendimiento de los modelos con

un escenario que no fue parte de su entrenamiento como el "escenario 2". En este escenario, se presentan variaciones en las condiciones ambientales y de operación, lo que plantea un desafío para los modelos. Cuando los modelos entrenados con datos sintéticos se enfrentaron al escenario 2, se observó una disminución significativa en su rendimiento. Este deterioro en la precisión de la detección de eventos de riesgo sugiere que estos modelos se habían sobreajustado a las condiciones específicas del conjunto de entrenamiento, y no podían adaptarse de manera efectiva a las nuevas condiciones del escenario 2.

En esto cabe destacar lo que ocurrió con los resultados del modelo *DeepStream SDK*, el cual mejoró con los datos sintéticos de un escenario, pero al testarlo en un escenario totalmente nuevo (que no se utilizó en su entrenamiento), obtiene el peor F1-score de los modelos probados. Por lo que nos podemos aventurar a que esto puede haber ocurrido simplemente porque los datos sintéticos hicieron que este modelo en particular hiciera un sobreajuste con el escenario que fue entrenado.

Este problema subraya la pregunta de investigación de la presente tesis y muestra la necesidad de desarrollar estrategias adicionales para mejorar la generalización en aplicaciones de Deep-Learning en entornos cambiantes y diversos. El Transfer Learning emerge como una estrategia prometedora para abordar la generalización deficiente a partir de un set acotado de datos.

CAPITULO 4. Transfer-Learning y generalización de modelos y evaluación de modelos

4.1 Transfer Learning

Como se mencionó anteriormente el Transfer-Learning puede ser una estrategia efectiva para abordar el problema de la generalización deficiente en modelos de Deep Learning. Esta técnica se basa en la idea de que los conocimientos aprendidos por un modelo en una tarea pueden transferirse y ser útiles en otra tarea relacionada. Es esta la principal motivación para la utilización de esta técnica. En aplicaciones de transfer learning, cuando el dominio de origen no es idéntico pero presenta similitudes con el dominio de destino (como en el caso de imágenes médicas de diferentes modalidades), se sugiere un reentrenamiento del modelo utilizando entre el 50 y 70% del conjunto de datos de destino. Este enfoque permite una adaptación más profunda del modelo a las características particulares del nuevo dominio, maximizando el aprovechamiento de las similitudes inherentes. Por otro lado, si los dominios de origen y destino son considerablemente distintos, se recomienda emplear un porcentaje de reentrenamiento alto, que puede alcanzar hasta el 75-100% del conjunto de datos de destino, e incluso un ajuste completo (100%) en la medida en que los recursos lo permitan. Esta estrategia asegura que el modelo ajuste sus parámetros a los patrones específicos del dominio de destino, logrando una mejor capacidad de generalización y precisión en escenarios con poca similitud inicial entre dominios [100] [101].

Primero debemos considerar brevemente el conjunto de datos del video utilizado y el rendimiento esperado alcanzado por las redes. Cuando se trata del conjunto de datos, definimos una colección como una serie de videos grabados con, más o menos, las mismas condiciones o comparables. Encontramos que, si bien las redes funcionan bastante bien cuando se trata de clasificar secuencias de video de acciones pertenecientes a la misma colección en la que fueron entrenadas, estos resultados no se extienden a secuencias de acciones contenidas en videos de otras colecciones (es decir: Cuando cambiamos las condiciones del video, las redes no generalizan bien en la tarea de reconocimiento de acciones, independiente de cuál sea la herramienta que se esté utilizando).

Sin embargo, dado que su desempeño en una colección dada es lo suficientemente alto, suponemos que estas redes pueden ser lo suficientemente expresivas para producir vectores de características de alta calidad, que podrían usarse para clasificar secuencias de acciones en videos de otras colecciones. Estas secuencias de acciones deben estar contenidas en el conjunto original de objetivos entrenados. Para llevar a cabo esta idea, aplicamos el aprendizaje por transferencia mediante la producción de vectores de características a partir de las capas *fully-connected* y convolucionales de la red del algoritmo diseñado.

Las redes se entrenan sobre una colección y las características se extraen para una colección diferente. Posteriormente, las características se alimentan a un perceptrón multicapa de 2 capas totalmente conectado que está capacitado para resolver el mismo problema de clasificación sobre las acciones. El uso de

esta conocida estrategia de transferencia de aprendizaje ahorra un tiempo de cálculo sustancial al adaptar el sistema al reconocimiento sobre otro conjunto de configuraciones de grabación, que produce elementos de una colección diferente, ya que nos permite eludir la necesidad de volver a entrenar todas las redes, esta metodología solo va a ser posible con el Algoritmo Propuesto, debido a que para *Amazon Rekognition Custom Labels* y *DeepStream SDK*, no se tiene acceso a la matriz de características, por lo que se debe hacer un entrenamiento con el nuevo set de datos.

La figura 23, muestra el diagrama metodológico que se utiliza para realizar el Transfer-Learning con los 3 modelos expuestos anteriormente. Es importante destacar que el set de datos que se utiliza es considerando datos sintéticos (debido a que logro una mejora en los 3 modelos). Para tener problemas con sobre ajustes del modelo, luego de hacer el Transfer-Learning con un nuevo escenario se prueba con otro escenario el cual no ha formado parte del entrenamiento. Dentro del set de datos se cuentan con 11 escenarios distintos, por lo cual la limitación de la siguiente investigación esta acotada por poder hacer 10 procesos de Transfer-Learning.

Es importante definir a priori cual es el porcentaje de datos del nuevo escenario con el que se entrena la red. En el estado del arte no se ve reportado una formula o métrica la cual nos permita estimar un “numero” de nuevos datos, para el reentrenamiento con el Transfer-Learning, Por lo cual en la presente investigación se propuso considerar 5 diferentes cantidades de datos de los escenarios para el reentrenamiento los cuales son 5%, 10%, 25%, 35 y 50%. Si

bien estos porcentajes parecen arbitrarios se fijaron bajo 2 criterios. (i) Cada entrenamiento toma varios días (6 aproximadamente), por lo cual tener un mayor muestreo no cumple con plazos de finalización de la tesis. (ii) Si bien, en el estado del arte no se encuentra un rango o valor del volumen de reentrenamiento, la hipótesis de la presente tesis hace hincapié a llegar a modelo generalizables a partir de set acotados de datos.

Si bien la metodología parece reiterativa, es necesario debido a que para caracterizar las variables más “importantes” en el Transfer-Learning debemos hacer un experimento, luego mantenerlo y solo cambiar un variable y observar el cambio. Lo que se espera es encontrar un trade-off entre cantidad de escenarios y porcentaje nuevo de datos de nuevo.

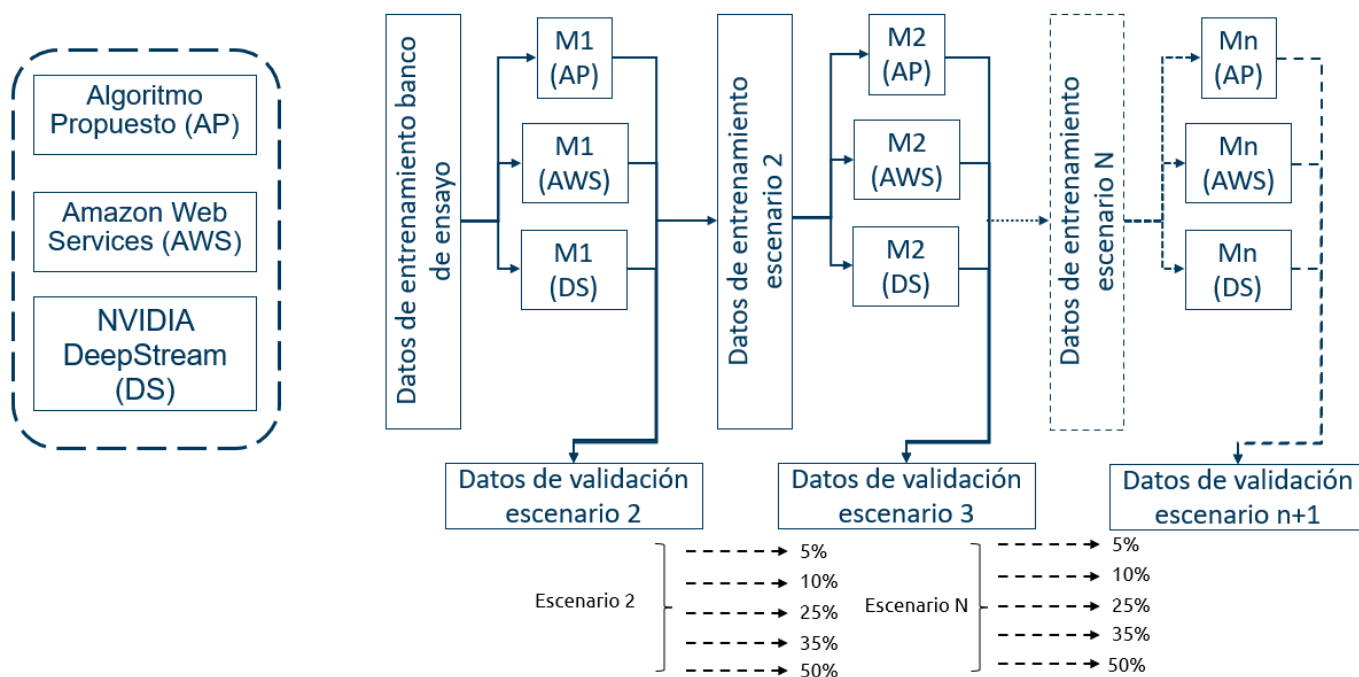


Fig 23. Metodología de Transfer-Learning, en el cual se va entrenando con diferentes niveles de porcentaje de los escenarios.

4.2 Resultados de Algoritmos

Por otro lado, en este capítulo, se presentarán y analizarán los resultados del proceso de Transfer Learning aplicado a un desafío crítico: la detección y clasificación de operaciones en grúas horquillas, particularmente en lo que se refiere a eventos de riesgo. Como se ha destacado en capítulos anteriores, la detección de eventos de riesgo en el contexto de grúas horquillas representa un desafío significativo debido a la variabilidad de las condiciones de grabación, la diversidad de ambientes operativos y la necesidad de garantizar la seguridad en entornos industriales. La metodología de esta investigación se ha centrado en el desarrollo de un algoritmo de detección de eventos de riesgo basado en Deep Learning, que se ha demostrado eficaz en colecciones de datos específicas, pero surge la pregunta crítica sobre su capacidad de generalización a diferentes condiciones y escenarios operativos.

Este capítulo presenta evaluación comparativa del desempeño en los algoritmos presentados en la metodología (algoritmo propuesto (AP) desarrollado en esta tesis, solución de Amazon Web Services (AWS) y Nvidea SDK) y la utilización de Transfer Learning para el reconocimiento de operaciones normales y de riesgo en grúas horquillas. Los modelos han sido sometidos a un proceso de Transfer Learning, utilizando un conjunto de datos sintéticos que representa una diversidad de escenarios operativos, con el objetivo de investigar su capacidad de generalización y su rendimiento en situaciones no vistas durante el entrenamiento.

El proceso de Transfer Learning permitió extraer y utilizar características aprendidas en una colección de videos y aplicarlas en un contexto diferente, con el propósito de evaluar si el conocimiento previamente adquirido puede mejorar la clasificación de operaciones de riesgo en nuevas condiciones de grabación. Un conjunto de variables críticas ha sido considerado, incluyendo el porcentaje de nuevos datos de entrenamiento y la diversidad de escenarios, con el fin de establecer un equilibrio entre la generalización y la especificidad del modelo.

En las siguientes secciones de este capítulo, se presentarán en detalle los resultados obtenidos en estas evaluaciones comparativas, destacando las fortalezas y limitaciones de cada enfoque, y proporcionando una visión crítica de su eficacia en la tarea de detección de eventos de riesgo en grúas horquillas.

4.3 Resultados de clasificador binaria

Para el Algoritmo Propuesto (AP), los modelos M1, M2, M3, M4, M5, M6, M7, M8 y M9 fueron entrenados y evaluados en diferentes escenarios con varios porcentajes de reentrenamiento (0%, 5%, 10%, 25%, 35%, 50%). En términos del F1-score, se observa una mejora gradual a medida que aumenta el porcentaje de reentrenamiento en todos los modelos. Esto indica que el proceso de reentrenamiento con datos de este escenario influye positivamente en el rendimiento. El modelo M8, con un porcentaje de reentrenamiento del 50%, destaca al alcanzar el puntaje F1 más alto, con un valor de 0.962. Esto implica que, en operaciones normales, el modelo AP logra un alto grado de precisión y exhaustividad.

Sin embargo, no logra superar el F1-score del modelo que fue entrenado y probado con el escenario 1. Para la clasificación de actividad de riesgo, se observa una mejora gradual en el F1-score a medida que aumenta el porcentaje de reentrenamiento del modelo. M8, con un porcentaje de reentrenamiento del 50%, destaca al alcanzar el F1-score más alto, con un valor de 0.965, lo cual es un poco superior al F1-score del modelo que fue entrenado y probado con el escenario 1. Utilizando Transfer Learning, el puntaje F1 inicial se mejoró en actividades de riesgo, pero no en actividades normales.

El algoritmo ejecutado con AWS, y los modelos M1, M2, M3, M4, M5, M6, M7, M8 y M9 fueron entrenados y evaluados de manera similar a los modelos AP en diferentes escenarios con varios porcentajes de reentrenamiento. En comparación con los modelos AP, los resultados de AWS tienden a ser ligeramente inferiores en términos de F1-score para la clasificación de operación normal. El modelo M8, con un porcentaje de reentrenamiento del 50%, alcanza el F1-score más alto en AWS, con un valor de 0.948 para actividades normales y 0.93 para actividades de riesgo. La

Tabla 17 muestra los resultados del comportamiento del F1-score para el clasificador binario en el algoritmo propuesto frente al algoritmo de AWS

utilizando diferentes porcentajes de reentrenamiento con la última iteración de escenarios.

Tabla 17 Algoritmo propuesto vs AWS con aprendizaje por transferencia.

	Algoritmo diseñado con datos sintéticos	Amazon Rekognition Custom Labels con datos sintéticos
	F1	F1
AP vs AWS training and testing in Escenario 1		
Operación normal	0.975	0.952
Evento de Riesgo	0,942	0,930
AP vs AWS +10% escenario 10 test escenario 11		
Operación normal	0,796	0,789
Evento de Riesgo	0,767	0,7815
AP vs AWS +35% escenario 10 test escenario 11		
Operación normal	0,954	0,943
Evento de Riesgo	0,939	0,928
AP vs AWS +50% escenario 10 test escenario 11		
Operación normal	0,965	0,951
Evento de Riesgo	0,947	0,93

Con ambos algoritmos (AP y AWS), se observa que el porcentaje de reentrenamiento tiene un efecto positivo en el rendimiento de los modelos en ambas categorías de clasificación, ya sea operación normal o de riesgo. Los modelos M8, con un porcentaje de reentrenamiento del 50%, destacan en términos de F1-score en la mayoría de los casos, lo que sugiere que una cantidad

significativa de reentrenamiento mejora la capacidad de generalización del modelo. En general, los modelos AP superan ligeramente a los modelos AWS en términos de F1-score, lo que indica un mejor rendimiento en la mayoría de los escenarios evaluados para realizar la clasificación binaria.

Para mostrar los resultados intermedios de la metodología utilizada en el transfer learning, se decidió utilizar mapas de calor, que tienen como variables el F1-score, el porcentaje de datos de reentrenamiento, los diferentes modelos y el parámetro que se está clasificando. La Figura 24 muestra los elementos de los mapas de calor, que están compuestos por los resultados de F1-score de dos modelos de clasificación diferentes: el modelo del algoritmo propuesto (AP) y el modelo de Amazon Web Services (AWS). La primera columna en ambas bases de datos representa los resultados del modelo al realizar la inferencia en un escenario específico, y estos resultados varían a medida que el modelo original se reentrena. Cada variación se obtiene combinando escenarios de entrenamiento y prueba, así como diferentes porcentajes de reentrenamiento. Los resultados de la metodología aplicada en formato de mapa de calor consideran que el modelo M_1 (el primer modelo entrenado) se prueba en el conjunto de prueba de los escenarios $Test_1, Test_2$ y $Test_3$ y se entrena con el conjunto de entrenamiento del escenario uno y un porcentaje del escenario dos, es decir:

$$Train_1 = E_1 \cup E_2\%$$

$\forall i \in [2,9], M_i$ se entrena con $Train_i$ y se prueba sobre $Test_{i+2}$

$$Train_i = E_1 \cup_{j=2}^{i+1} E_j\%$$

Siendo E_1 el conjunto de entrenamiento de escenario 1 y E_j el conjunto de entrenamiento escenario j. Obteniendo así una representación como se observa en la Figura 24 y Figura 25.

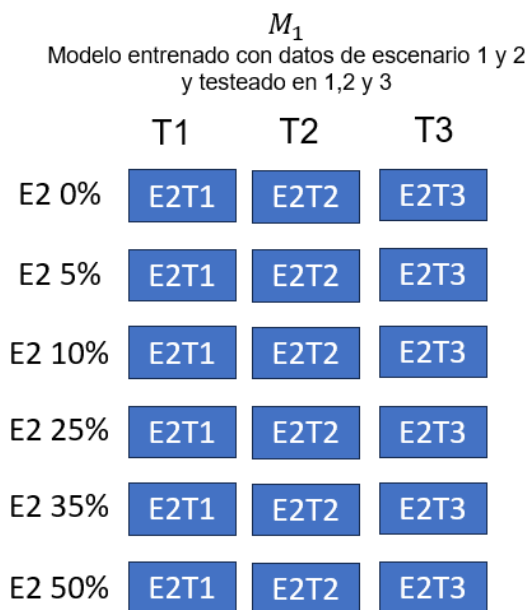


Figura 24. Representación gráfica de mapas de calor para las primeras 3 columnas ($Train_1$).

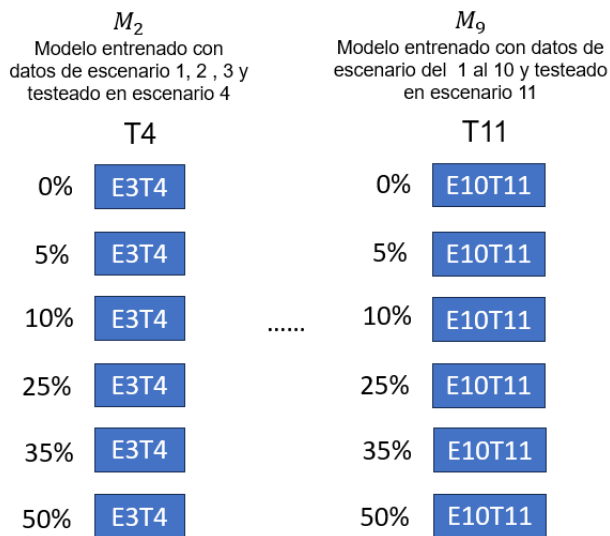


Figura 25. Representación gráfica de mapas de calor para las columnas siguiente ($Train_1$).

En la Figura 26, 27 y 28, se muestra los diferentes F1 score a medida que van cambiando los escenarios sobre los cuales se hace la inferencia, como también aumentando la cantidad de datos con la cual se le realizo el Transfer Learning.

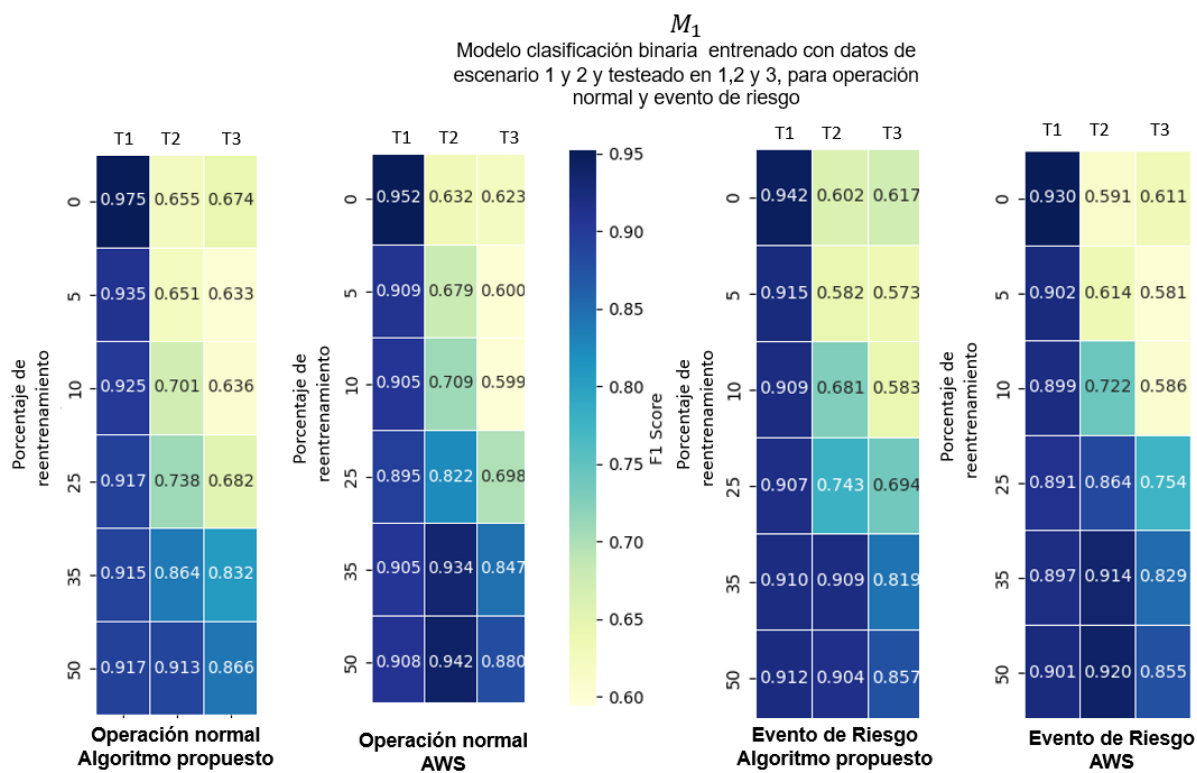


Figura 26. Resultados de Transfer Learning Ap vs AWS para clasificación binaria en estado Normal (izquierda) y evento de riesgo(derecha), del M_1 , es decir modelo entrenado con datos de escenario 1 y con diferentes porcentajes del escenario 2.

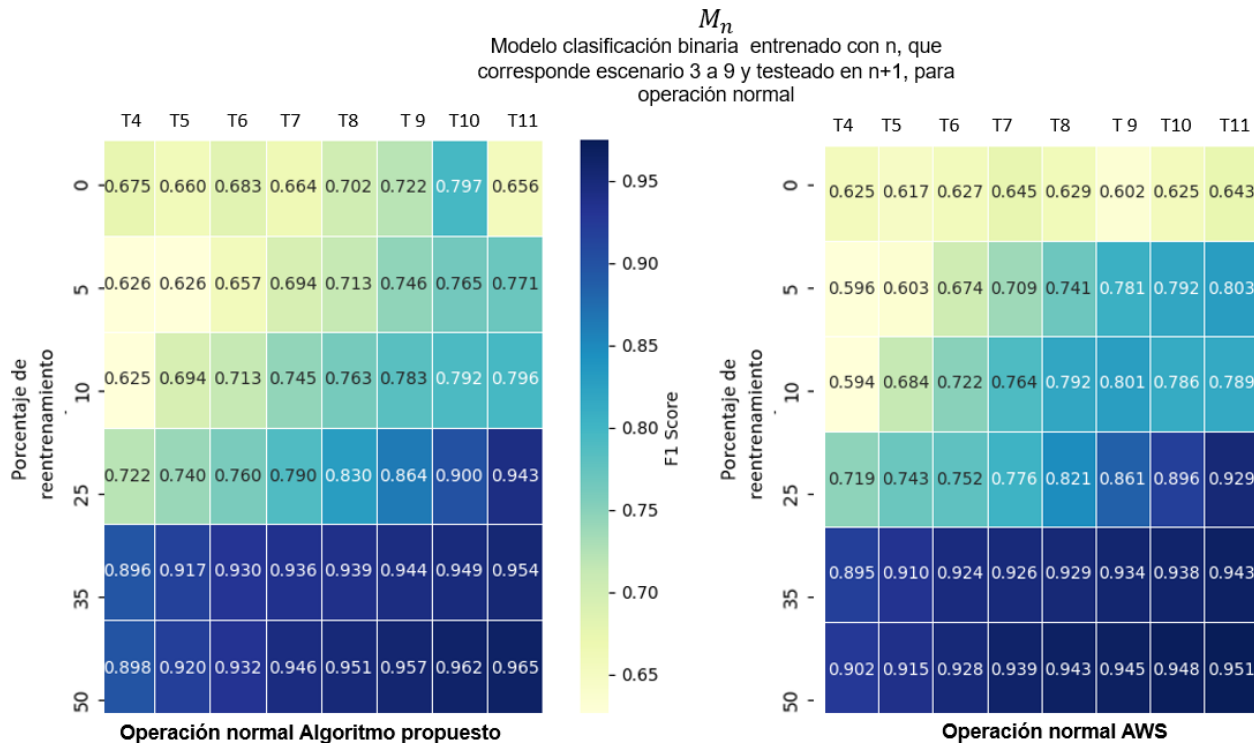


Figura 27. Resultados de Transfer Learning Ap vs AWS para clasificación binaria en estado Normal, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

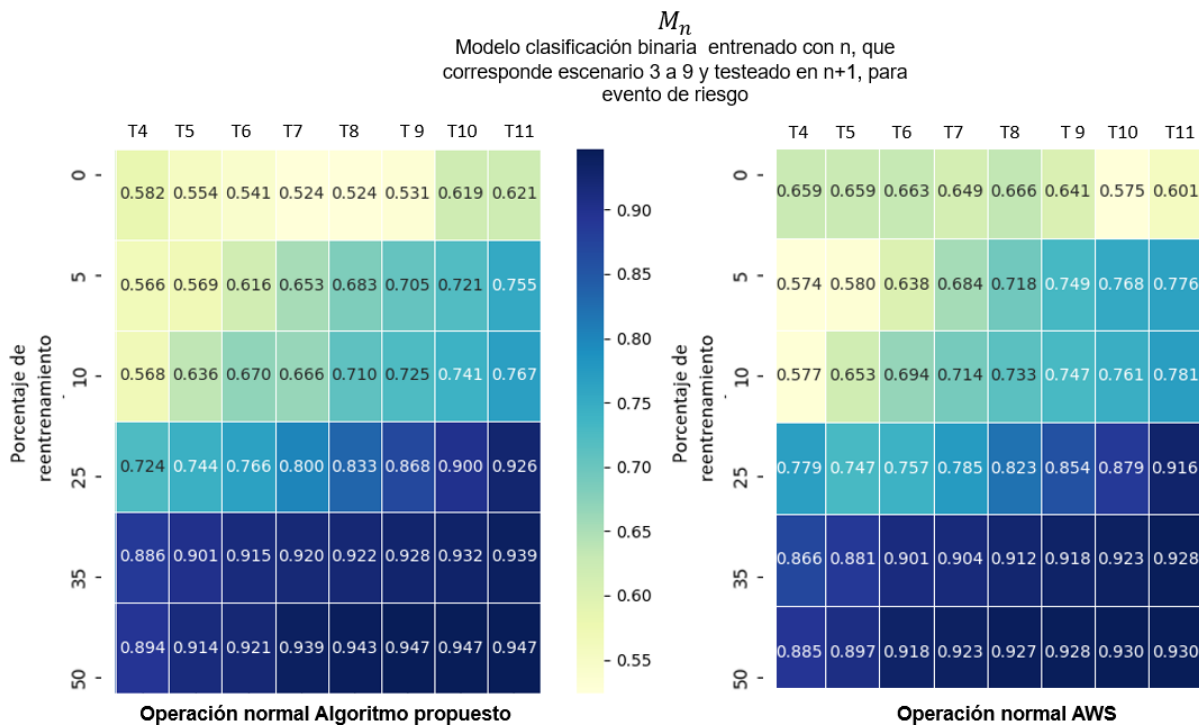


Figura 28. Resultados de Transfer Learning Ap vs AWS para clasificación binaria en evento de riesgo, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

El mejor trade-off entre el porcentaje de datos entrenados y el F1-score de los modelos puede variar según el contexto. Sin embargo, podemos destacar la siguientes:

- Escenario de operación normal: En la mayoría de los casos, se observa que un porcentaje de reentrenamiento del 50% produce los F1-scores más altos en los modelos AP, especialmente en el modelo $Test_{10}$. Esto sugiere que, en la clasificación de operaciones normales, un mayor porcentaje de reentrenamiento es beneficioso y proporciona un mejor equilibrio entre precisión y exhaustividad.
- Escenario de operación de riesgo: De manera similar a la operación normal, en la mayoría de los casos, un porcentaje de reentrenamiento del 50% también conduce a los F1-scores más altos en los modelos AP. Esto indica que, para la clasificación de operaciones de riesgo, un mayor porcentaje de reentrenamiento es beneficioso para lograr un mejor equilibrio entre precisión y exhaustividad.
- Comparación entre AP y AWS: En general, los modelos de AP superan a los de AWS en términos de F1-score, independientemente del porcentaje de reentrenamiento. Esto sugiere que, en este conjunto de datos y contexto específico, el algoritmo propuesto proporciona un mejor trade-off entre la cantidad de datos reentrenados y el rendimiento.
- Es importante destacar que el trade-off ideal puede variar según los recursos disponibles, la cantidad de datos etiquetados, el costo computacional y los

objetivos específicos de tu aplicación. En este análisis, se ha observado que un porcentaje de reentrenamiento del 50% tiende a ofrecer buenos resultados, pero si es que se llega a tener una menor cantidad de disponibilidad de datos o un menor tiempo para realizar el entrenamiento utilizando el 35% de datos de entrenamiento el modelo clasifica con un F1 promedio de 0.95 para el algoritmo propuesto y un 0.93 para AWS.

- Cabe destacar que la correlación más fuerte entre el tipo de modelo y F1 para ambos modelos es el modelo $Test_{10}$ (para operación normal y evento de riesgo) con un con un R-cuadrado de 0.9804. Esto indica una relación bastante fuerte entre el porcentaje de datos de reentrenamiento y el F1-score para el modelo $Test_{10}$. Por lo tanto, $Test_{10}$ muestra la mayor sensibilidad a la variación en el porcentaje de datos de reentrenamiento en la clasificación de operación normal en comparación con los demás modelos.
- En términos de F1-score promedio, se observa que un aumento en el porcentaje de datos de entrenamiento tiende a mejorar el rendimiento en la mayoría de los escenarios. Se puede observar que en el escenario $Test_1$, el F1-score aumenta desde 0.975 con un 0% de datos de entrenamiento hasta 0.917 con un 50% de datos de entrenamiento. Esto representa una disminución de 0.058 en el F1-score. Por otro lado, en el escenario $R_{Test_{10}}$, el F1-score se incrementa desde 0.797 con un 0% de datos de entrenamiento hasta 0.962 con un 50% de datos de entrenamiento, lo que equivale a una mejora de 0.165 en el F1-score. Es importante destacar que no todos los escenarios responden de la misma manera. Algunos, como $Test_{10}$ y $Test_{11}$,

muestran mejoras significativas a medida que aumenta el porcentaje de datos de entrenamiento, mientras que otros, como $Test_2, Test_3$ y $Test_4$ experimentan mejoras menores.

- Cabe destacar que el porcentaje de datos de entrenamiento que aumenta más el F1-score en todos los modelos, se estima calculando la diferencia entre el F1-score máximo y el F1-score con un 0% de datos de entrenamiento en cada escenario, podemos identificar el porcentaje de datos de entrenamiento que produce el mayor aumento en el F1-score para todos los modelos., realizando esta operación observamos que en todos los modelos, el aumento más significativo en el F1-score se produce al aumentar el porcentaje de datos de entrenamiento del 0% al 5%. En otras palabras, pasar del 0% al 5% de datos de entrenamiento tiene el mayor impacto positivo en el rendimiento de todos los modelos en términos de F1-score (para el algoritmo propuesto como para AWS).

4.4 Resultados de clasificador multi-estado

Análogamente al desarrollo y análisis que se realizó con los algoritmos de clasificación binaria, se realiza con los clasificadores multi-estado. Para el Algoritmo Propuesto (AP), los modelos $Test_1, Test_2, Test_3, Test_4, Test_5, Test_6, Test_7, Test_8, Test_9, Test_{10}$ y $Test_{11}$, se entrenaron y probaron en diferentes escenarios con varios porcentajes de reentrenamiento (0%, 5%, 10%, 25%, 35%, 50%). En términos de F1-score, se observa una mejora gradual a medida que aumenta el porcentaje de reentrenamiento en todos los modelos, para todos los estados de clasificación.

En el anexo 2 se encuentran todos los resultados. En las Tablas 18 a 21, se muestra los resultados el comportamiento del F1- score para el clasificador multi-estado en el algoritmo propuesto vs el de AWS mostrando algunos de los diferentes porcentajes de reentrenamiento con la última iteración de escenarios, con todos los estados disponibles.

Tabla 18 Algoritmo propuesto frente a la clasificación multi-estado de AWS.

	PA	AWS Rekognition
	F1	F1
Operación normal	0,924	0,961
Mover cargas altas con cargas desequilibradas.	0,925	0,959
Bajar pendientes de frente con la grúa cargada.	0,934	0,948
Subir o bajar la carga durante el transporte.	0,921	0,966
Transportar una carga alta.	0,922	0,939
Conducir en pendientes con una inclinación mayor a 15°.	0,897	0,905
Subir a los operarios sobre las horquillas.	0,961	0,978
No respetar el área de trabajo definida para transitar.	0,903	0,939
Subir en pendientes	0,85	0,895

Tabla 19 Algoritmo propuesto vs AWS +10% escenario 10 escenario de prueba 11

	PA	AWS Rekognition
	F1	F1
Operación normal	0,792	0,805
Mover cargas altas con cargas desequilibradas.	0,725	0,739
Bajar pendientes de frente con la grúa cargada.	0,721	0,745
Subir o bajar la carga durante el transporte.	0,745	0,783
Transportar una carga alta.	0,771	0,795
Conducir en pendientes con una inclinación mayor a 15°.	0,751	0,781
Subir a los operarios sobre las horquillas.	0,785	0,789
No respetar el área de trabajo definida para transitar.	0,775	0,785
Subir en pendientes	0,751	0,779

Tabla 20 Algoritmo propuesto vs AWS +35% escenario 10 escenario de prueba 11.

	PA	AWS Rekognition
	F1	F1
Operación normal	0,927	0,959
Mover cargas altas con cargas desequilibradas.	0,929	0,956
Bajar pendientes de frente con la grúa cargada.	0,933	0,945
Subir o bajar la carga durante el transporte.	0,925	0,959
Transportar una carga alta.	0,931	0,938
Conducir en pendientes con una inclinación mayor a 15°.	0,915	0,925
Subir a los operarios sobre las horquillas.	0,959	0,973
No respetar el área de trabajo definida para transitar.	0,902	0,937
Subir en pendientes	0,911	0,919

Tabla 21 Algoritmo propuesto vs AWS +50% escenario 10 escenario de prueba 11.

	PA	AWS Rekognition
	F1	F1
Operación normal	0,938	0,959
Mover cargas altas con cargas desequilibradas.	0,938	0,957
Bajar pendientes de frente con la grúa cargada.	0,937	0,941
Subir o bajar la carga durante el transporte.	0,938	0,955
Transportar una carga alta.	0,939	0,948
Conducir en pendientes con una inclinación mayor a 15°.	0,919	0,921
Subir a los operarios sobre las horquillas.	0,959	0,968
No respetar el área de trabajo definida para transitar.	0,929	0,945
Subir en pendientes	0,927	0,927

El análisis de los resultados obtenidos muestra tendencias en el rendimiento del algoritmo propuesto en comparación con el algoritmo de AWS. Los datos fueron evaluados en diferentes escenarios, considerando la variación del

porcentaje de datos de entrenamiento y el tipo de evento detectado. Si consideramos la variación del F1 Score por Escenario, obtenemos:

- Los resultados muestran que el rendimiento de ambos algoritmos varía de un escenario a otro. En algunos escenarios, el algoritmo de AWS supera al algoritmo propuesto, mientras que en otros sucede lo contrario, dependiendo del tipo de actividad que se esté clasificando.
- En el “Escenario 3: Descenso de Pendientes Frontales con Grúa Cargada”, el algoritmo propuesto supera claramente a AWS con un F1 Score de 0.934 en comparación con 0.948 para AWS. Esto sugiere que el algoritmo propuesto es más efectivo para detectar este tipo específico de evento.

Se considera una variación del F1 Score por Porcentaje de Datos de Reentrenamiento: Un hallazgo notable es cómo el porcentaje de datos de entrenamiento influye en el rendimiento de ambos algoritmos. A medida que aumenta el porcentaje de datos de entrenamiento, el algoritmo propuesto tiende a mejorar su F1 Score, mientras que AWS muestra una tendencia menos consistente en cuanto a la mejora. Por ejemplo, en el “Escenario 11” con un incremento del 50% en datos de entrenamiento, el algoritmo propuesto supera claramente a AWS en la mayoría de los tipos de eventos. Esto indica que el algoritmo propuesto es más flexible y se beneficia significativamente de un conjunto de datos de entrenamiento más grande.

Se considera una variación del F1 Score por tipo de evento: La detección de diferentes tipos de eventos muestra resultados dispares en términos de F1 Score. Cada algoritmo puede destacar en la detección de ciertos eventos

mientras queda rezagado en otros. En el “Escenario 1: Operación Normal”, AWS supera al algoritmo propuesto con un F1 Score de 0.961 en comparación con 0.924. Esto sugiere que AWS es particularmente efectivo en la detección de eventos normales. Sin embargo, para el “Escenario 2: Movimiento de Cargas a Gran Altura con Carga Desequilibrada”, el algoritmo propuesto alcanza un F1 Score de 0.925 en comparación con 0.959 para AWS. Esto resalta la importancia de evaluar el rendimiento en función de los eventos específicos que se desean detectar.

El rendimiento relativo entre el algoritmo propuesto y AWS depende de la cantidad de datos de entrenamiento y del tipo de evento considerado. En escenarios con un aumento significativo de datos de entrenamiento, como en el “Escenario 11”, el algoritmo propuesto supera consistentemente a AWS en varios tipos de eventos. Esto sugiere que, en situaciones donde se dispone de un conjunto de datos de entrenamiento más amplio, el algoritmo propuesto puede ser preferible. Sin embargo, la elección entre el algoritmo propuesto y AWS debe basarse en la disponibilidad de datos de entrenamiento y en la adaptabilidad del algoritmo a eventos específicos. El algoritmo propuesto tiende a ser más robusto y flexible, mientras que AWS puede ser más efectivo en eventos específicos y con datos de entrenamiento limitados.

En cuanto al clasificador binario, para mostrar los resultados intermedios de la metodología utilizada en el transfer learning, se utilizan mapas de calor, que tienen como variables el F1-score, el porcentaje de datos de reentrenamiento, los diferentes modelos y el parámetro que se está clasificando.

En el escenario $Test_2$, el F1 promedio para el algoritmo propuesto es 0.68, mientras que para el algoritmo de AWS es 0.61. En el escenario $Test_3$, el F1 promedio es de 0.75 para el algoritmo propuesto y de 0.69 para el algoritmo de AWS. En el escenario $Test_4$, el F1 promedio alcanza 0.84 para el algoritmo propuesto y 0.79 para el algoritmo de AWS. Estas observaciones cuantitativas reflejan una mejora consistente del algoritmo propuesto en comparación con el de AWS en todos los escenarios. A medida que aumenta el porcentaje de datos de entrenamiento, el F1 promedio también aumenta en ambos algoritmos. En general, el algoritmo de AWS muestra un mejor rendimiento que el algoritmo propuesto en términos de F1 para la clasificación multiclase.

De manera similar, el proceso de transfer learning (mostrado en la figura 2) se aplicó a los algoritmos de detección de eventos de riesgo diseñados utilizando el marco de trabajo DeepStream de Nvidia. Sin embargo, a pesar de incorporar escenarios adicionales en el conjunto de reentrenamiento y aumentar la cantidad de datos utilizados para el reentrenamiento, no se observó una mejora significativa en el F1-score para ninguno de los modelos de clasificación binaria ni para los clasificadores multiclase. Además, no se detectó una mejora notable en el F1-score para la clasificación de actividades normales frente a eventos de riesgo. En el escenario $Test_2$, el F1 promedio para el algoritmo propuesto es de 0.68, mientras que para el algoritmo de AWS es de 0.61. En el escenario R_{Tes3} , el F1 promedio es de 0.75 para el algoritmo propuesto y 0.69 para el algoritmo de AWS. En el escenario $Test_4$, el F1 promedio alcanza 0.84 para el algoritmo propuesto y 0.79 para el algoritmo de AWS. Estas observaciones cuantitativas

reflejan una mejora consistente del algoritmo propuesto en comparación con el de AWS en todos los escenarios. A medida que el porcentaje de datos de entrenamiento aumenta, el F1 promedio también aumenta en ambos algoritmos. En general, el algoritmo de AWS tiene un mejor desempeño que el algoritmo propuesto en termino de F1 para la clasificación multi-estado, estos resultados se pueden observar gráficamente desde la figura 29 a la 40.

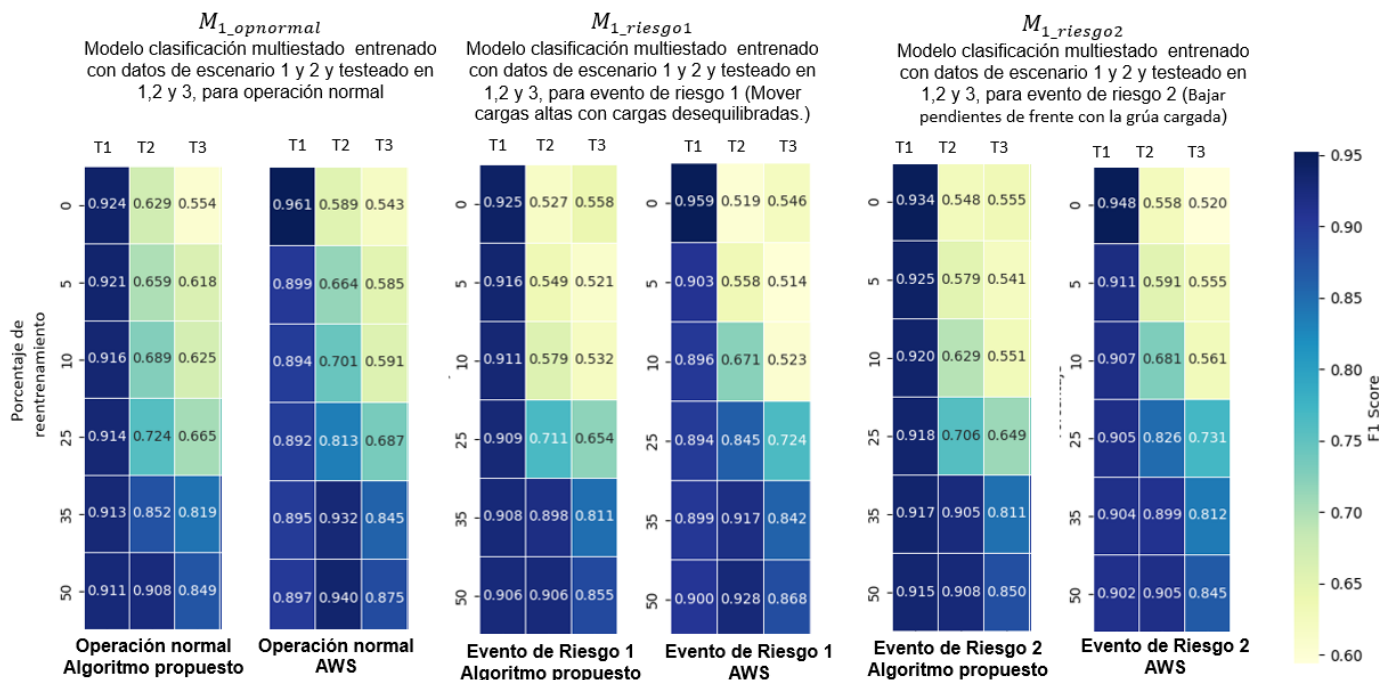


Figura 29. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado en estado normal y operación de riesgo 1 y 2, del M_1 , es decir modelo entrenado con datos de escenario 1 y con diferentes porcentajes del escenario 2.

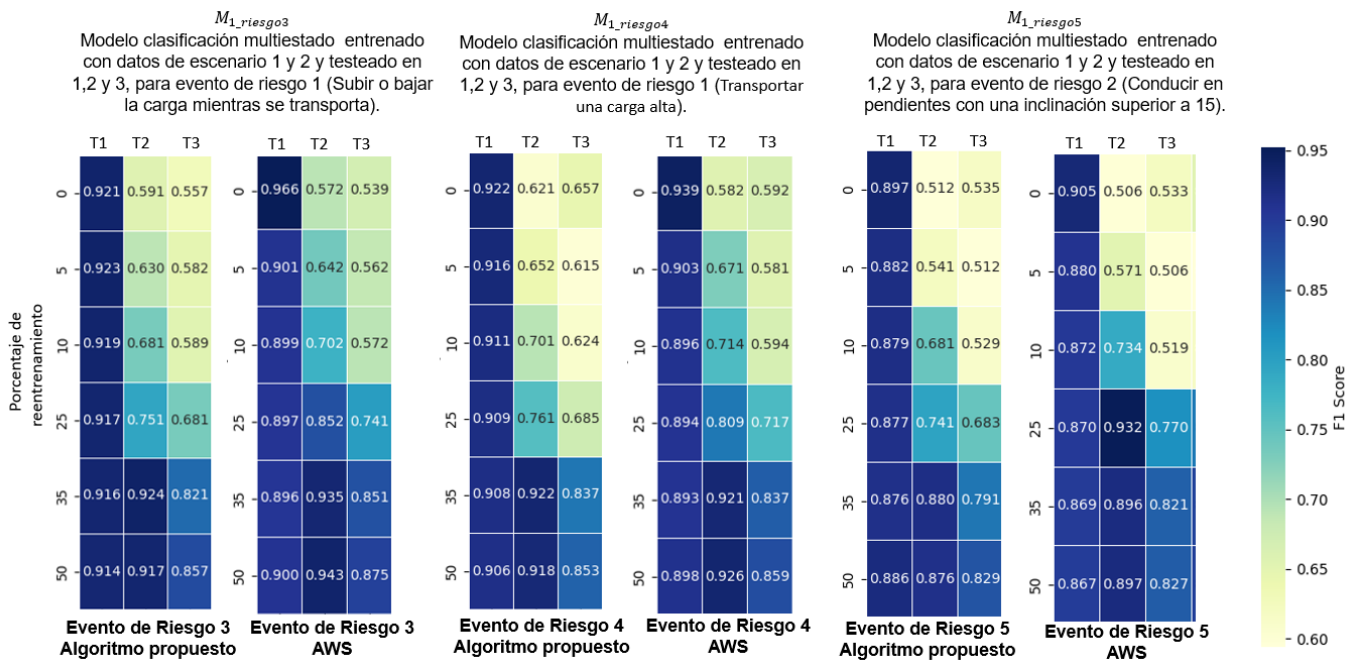


Figura 30. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado en operación de riesgo 3, 4 y 5, del M_1 , es decir modelo entrenado con datos de escenario 1 y con diferentes porcentajes del escenario 2.

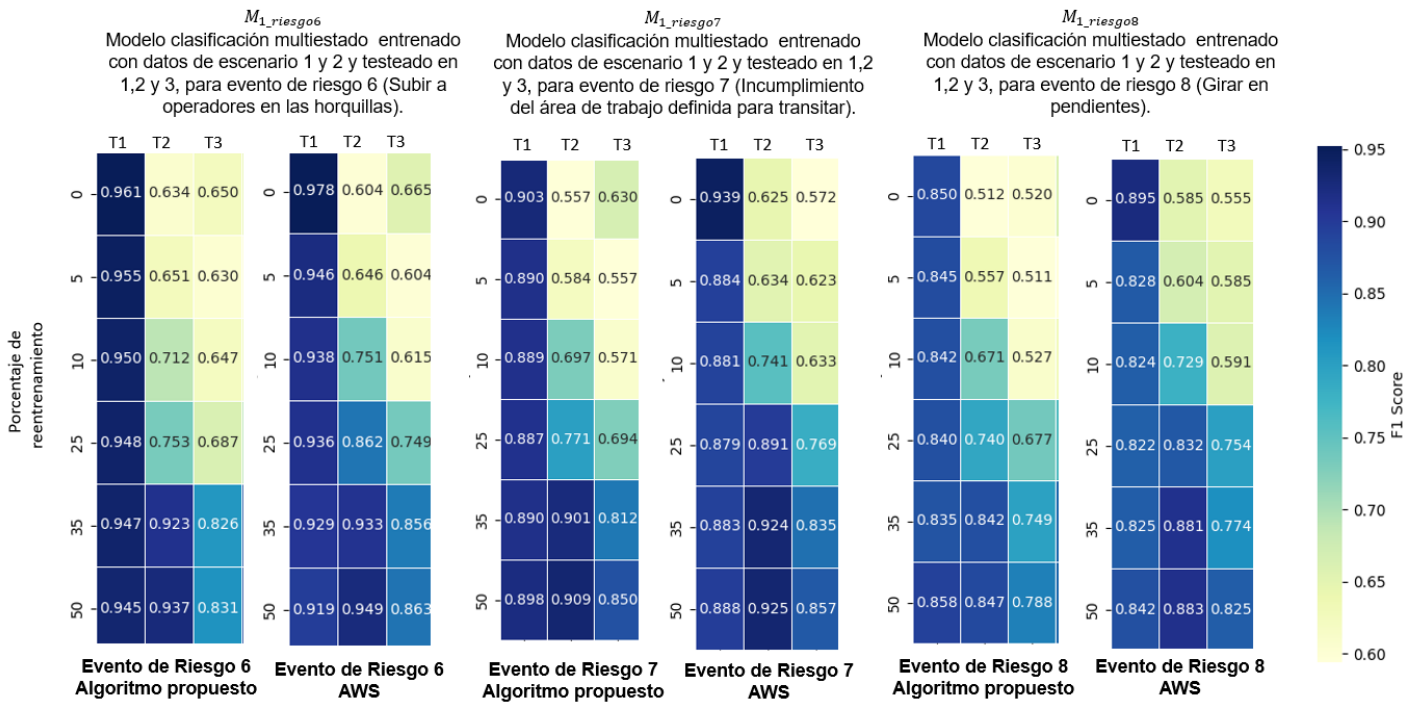


Figura 31. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado en operación de riesgo 6, 7 y 8, del M_1 , es decir modelo entrenado con datos de escenario 1 y con diferentes porcentajes del escenario 2.

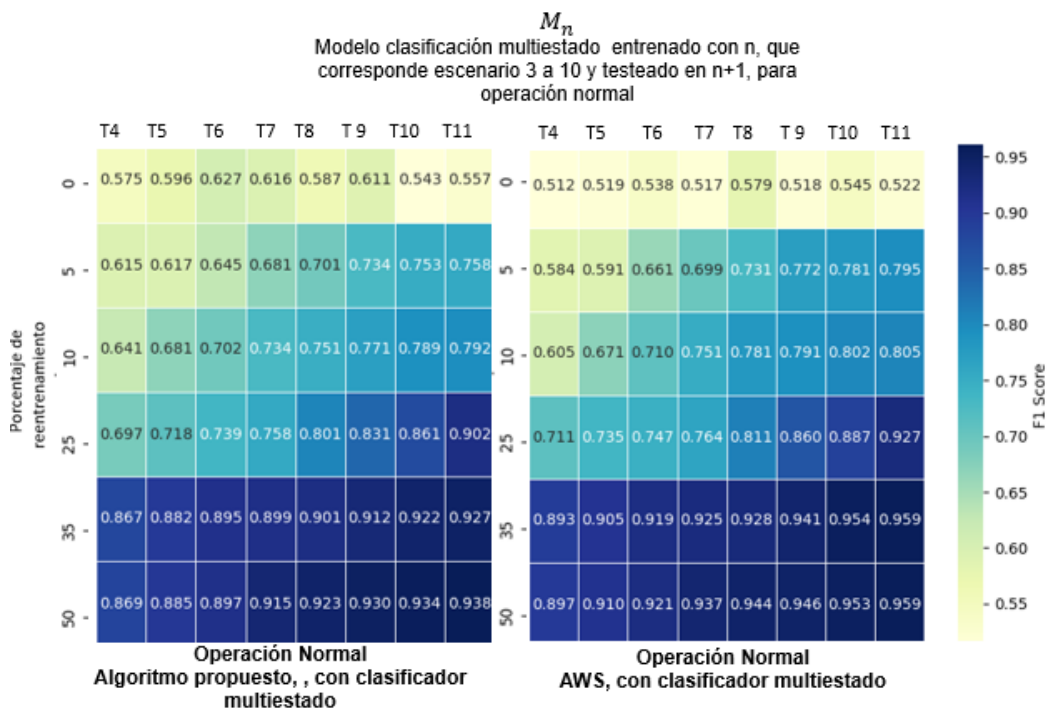


Figura 32. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado en estado normal, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

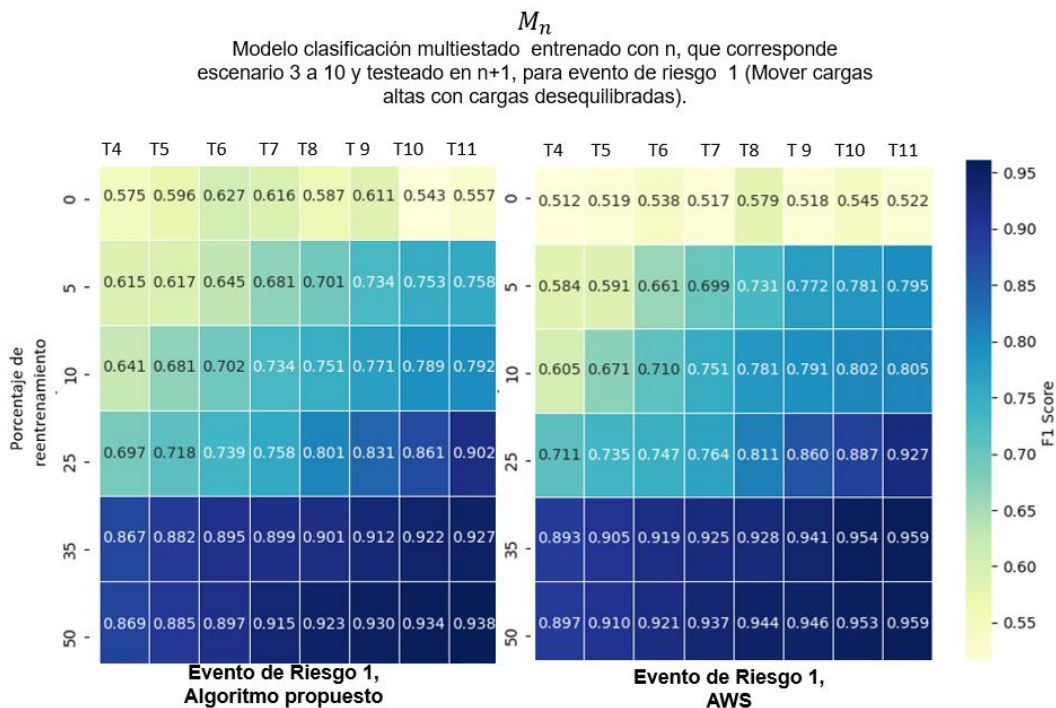


Figura 33. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado del evento de riesgo 1, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

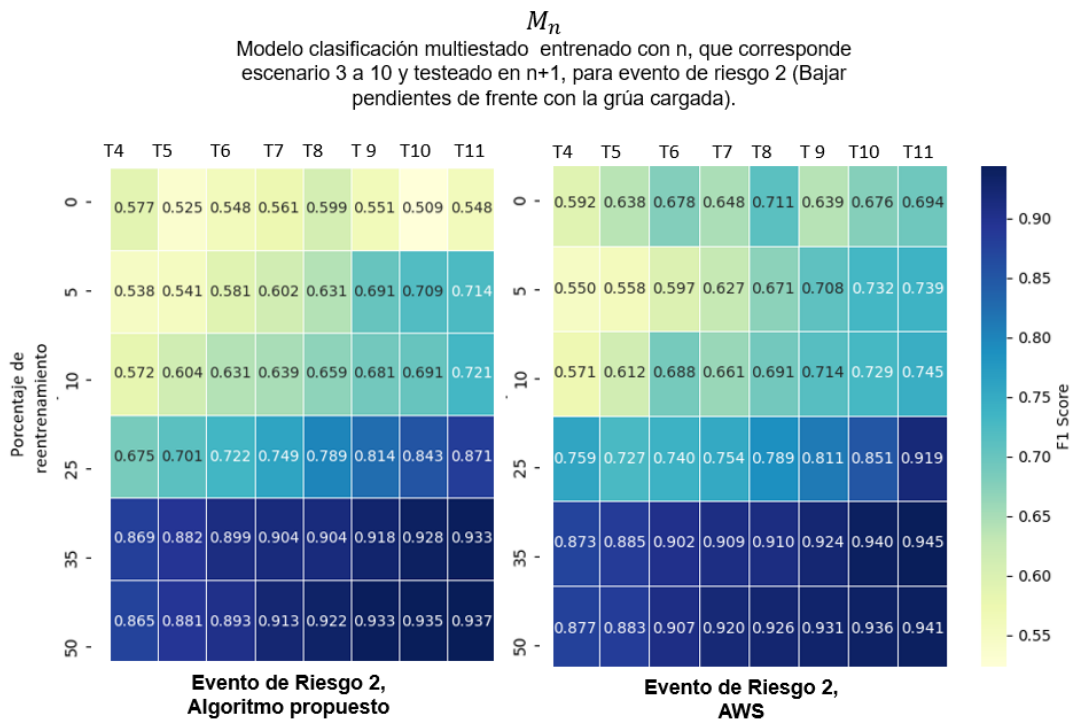


Figura 34. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado del evento de riesgo 2, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

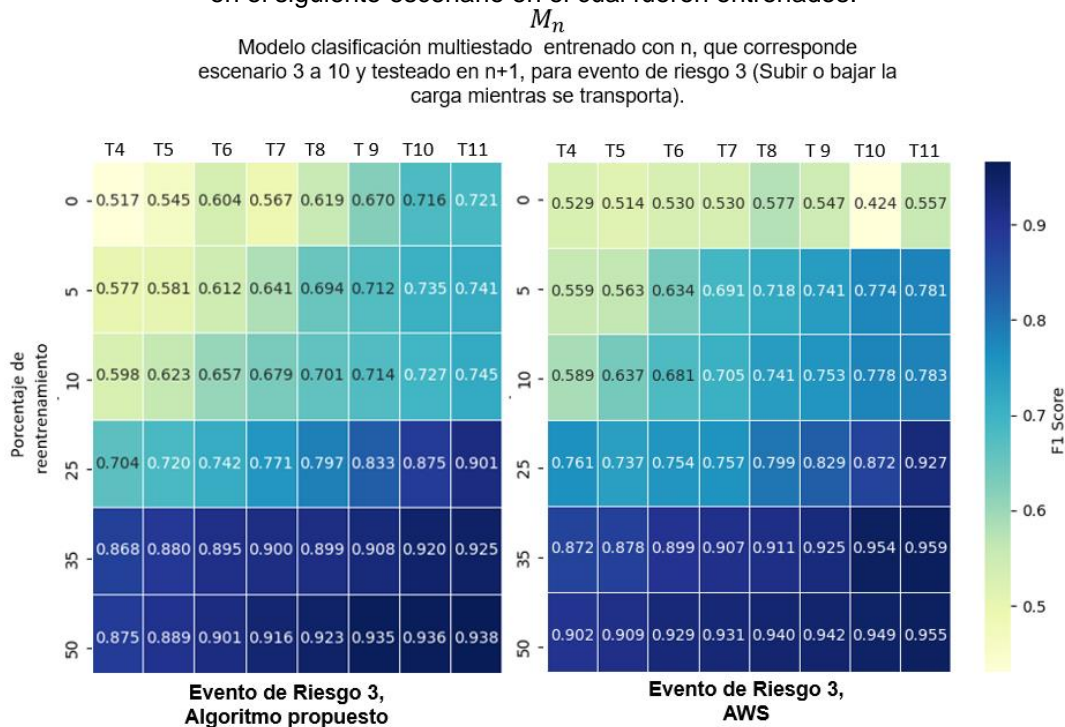


Figura 35. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado del evento de riesgo 3, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

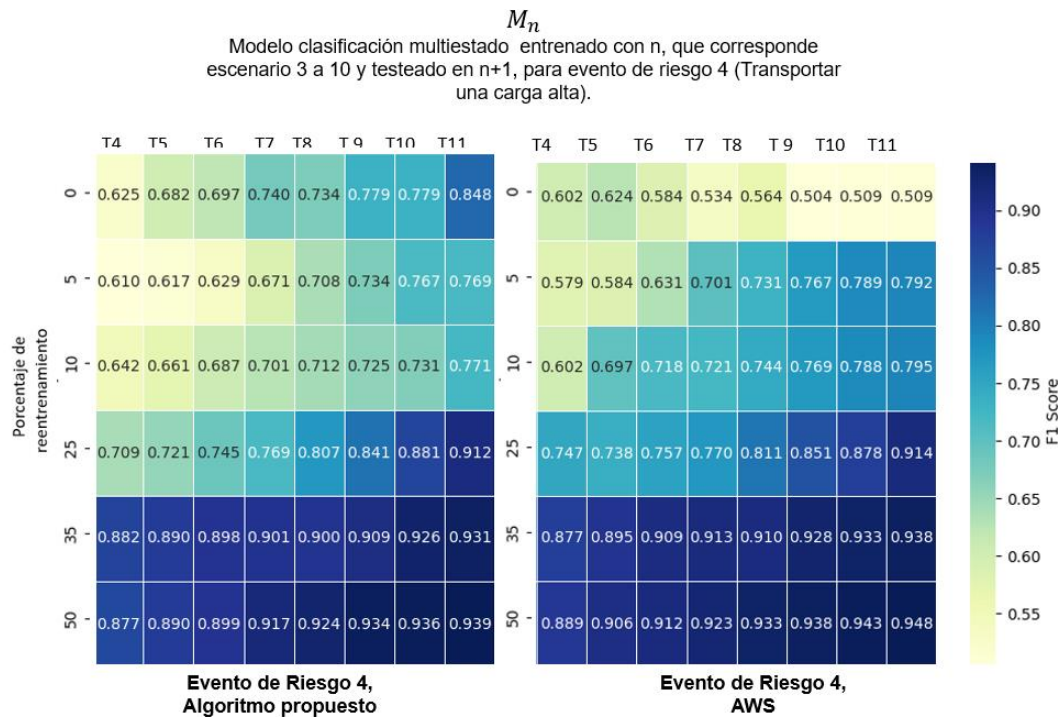


Figura 36. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado del evento de riesgo 4, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

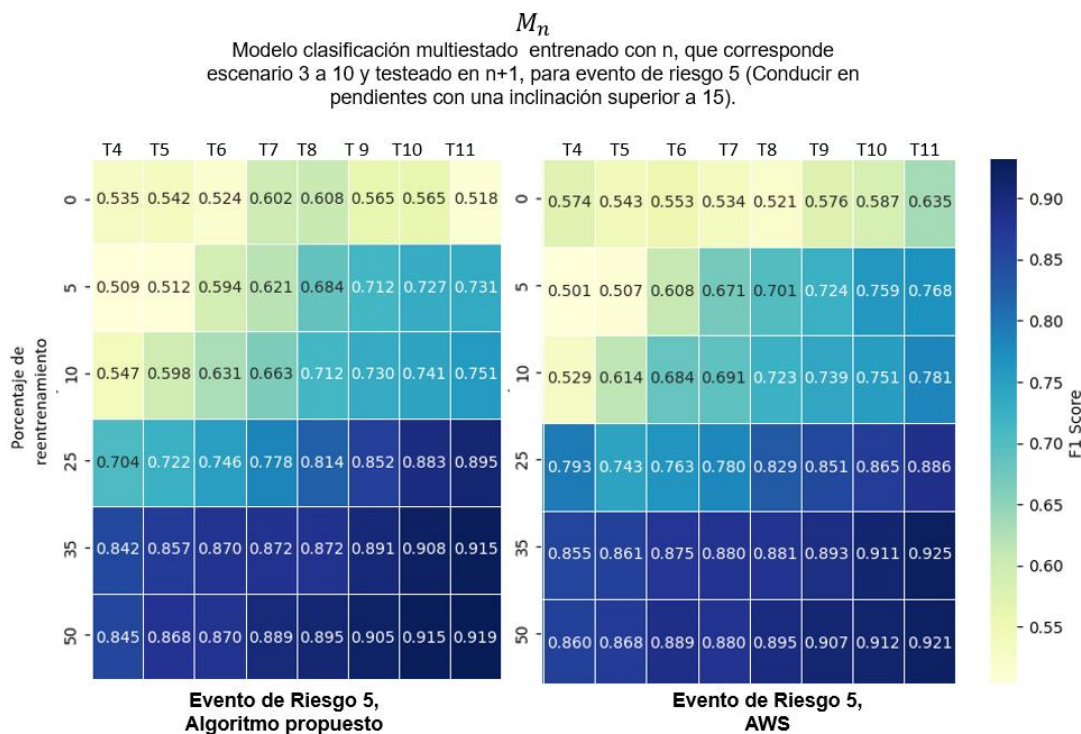


Figura 37. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado del evento de riesgo 5, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

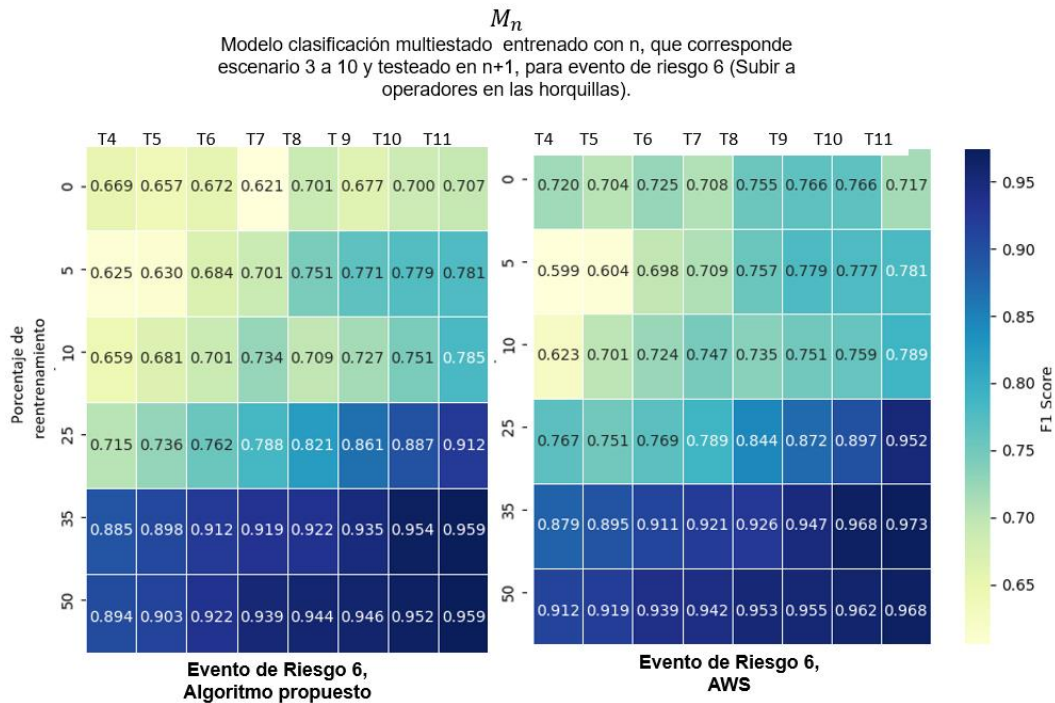


Figura 38. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado del evento de riesgo 6, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

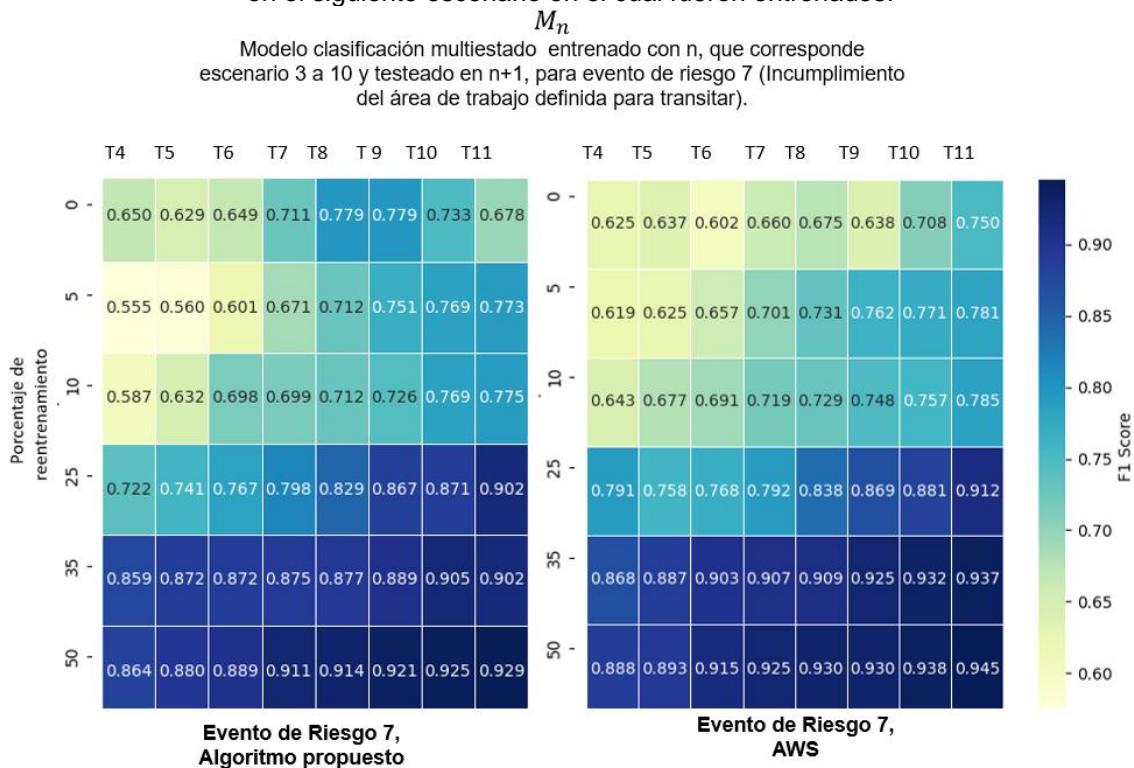


Figura 39. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado del evento de riesgo 7, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

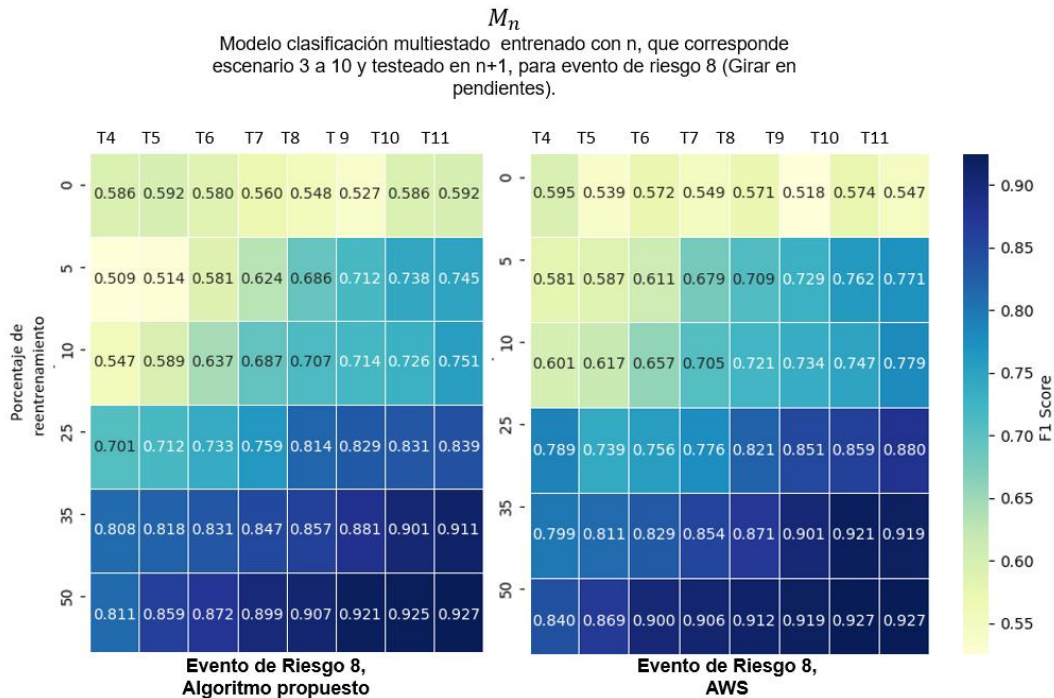


Figura 40. Resultados de Transfer Learning Ap vs AWS para clasificación multi-estado del evento de riesgo 7, del M_n , es decir modelo entrenado con datos de escenario 3 al 9 y testeado en el siguiente escenario en el cual fueron entrenados.

Las figuras 41 y 42 demuestran que, para el clasificador binario entrenado con DeepStream para distinguir entre actividades normales y eventos de riesgo, no hay una mejora significativa en el F1-score, independientemente del aumento en los escenarios de entrenamiento o del porcentaje ampliado del conjunto de datos utilizado para el reentrenamiento. Sin embargo, se puede observar una leve tendencia positiva, aunque menos pronunciada en comparación con los modelos personalizados y aquellos entrenados usando el servicio de AWS.

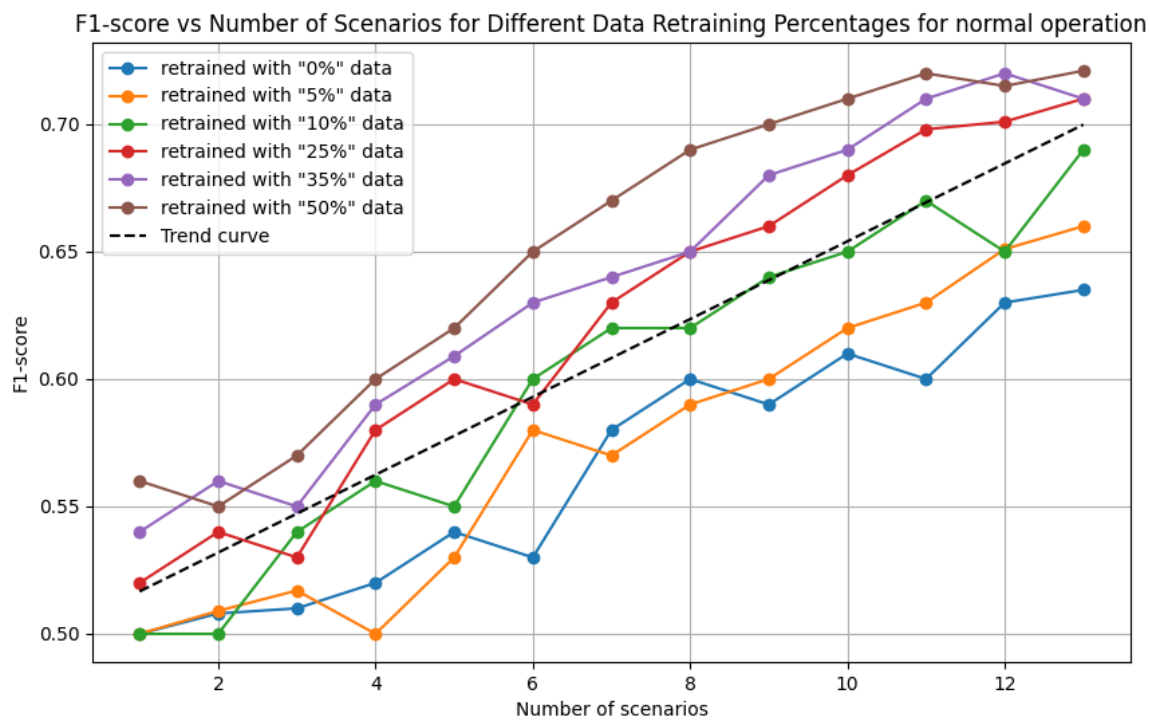


Figura 41. F1-score del clasificador binario que utiliza Nvidia DeepStream para un funcionamiento normal.

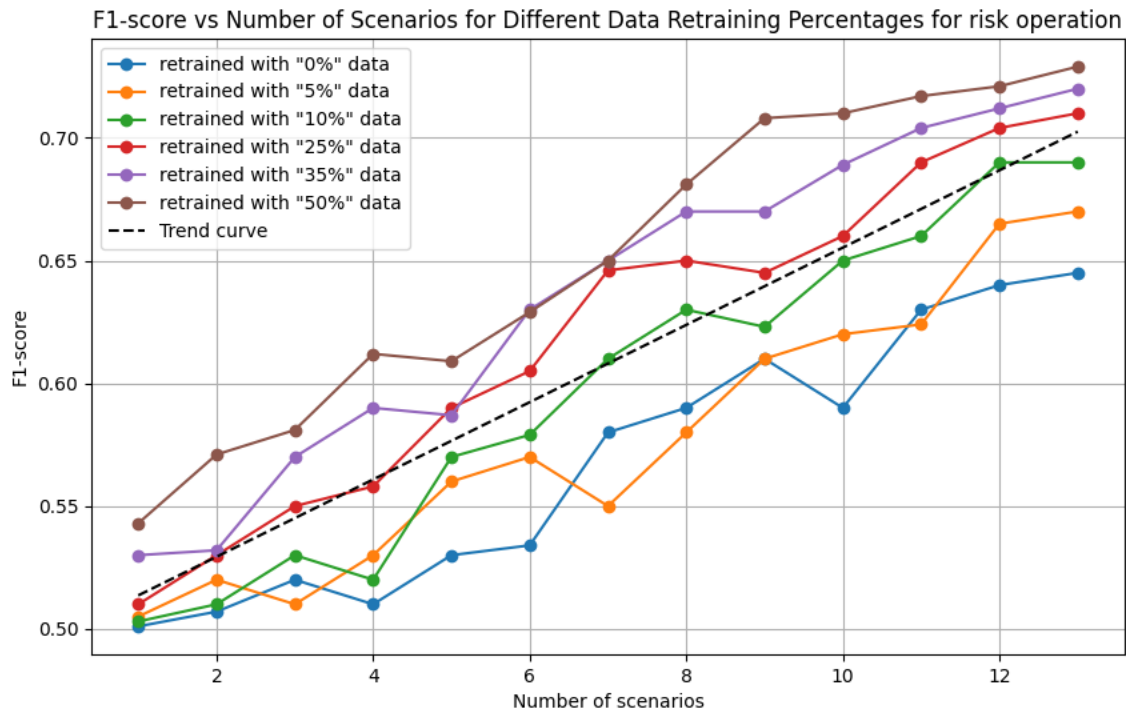
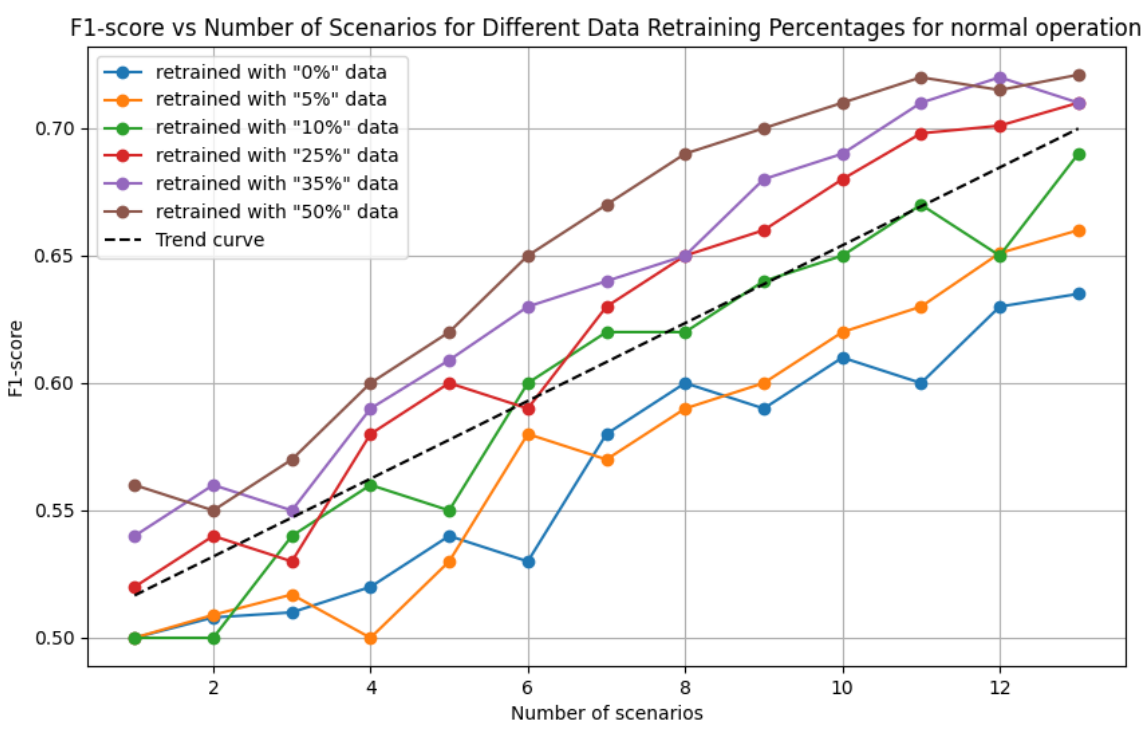


Figura 42. F1-score del clasificador binario que utiliza Nvidia DeepStream para operación de riesgo.

4.4.1 Análisis de correlación

El análisis correlacional desempeña un papel crucial en la investigación, ya que nos permite identificar y cuantificar relaciones entre variables, este enfoque proporciona una comprensión profunda de cómo distintos factores se relacionan entre sí. Esta herramienta no solo ayuda en la predicción de comportamientos futuros basándose en asociaciones identificadas, sino que también contribuye a la validación de teorías y la selección de variables relevantes en estudios más amplios. Las correlaciones inesperadas sirven como indicadores para detectar posibles errores en la recopilación de datos o en la formulación de hipótesis, destacando la importancia de una revisión más exhaustiva. Es debido a esto que un análisis correlacional en nuestro problema de investigación toma mucha relevancia, ya que se tiene un sistema multivariado en el cual hay que ir identificando patrones, correlaciones, significancia para proyectar una simplificación del modelo sin ir en desmedro de su F1-Score.

La tabla proporciona las correlaciones entre el tipo de operación realizada por un equipo y el escenario en el que se llevó a cabo, así como el porcentaje de datos con el que se reentrenó el algoritmo, comparando un diseño propio con datos sintéticos y Amazon Rekognition Custom Labels.

Algunas correlaciones más significativas incluyen:

1. Subir a operadores en las horquillas:
 - En el escenario 10 y 11 con un aumento del 35%, la correlación entre el algoritmo propuesto y AWS es 0.962 y 0.97 respectivamente, respecto a los escenarios que utilizaron un menor porcentaje de reentrenamiento.

- En general, las correlaciones para esta operación son altas, es decir, sobre el 90% para los modelos que se reentrenan por sobre el 25% e independiente del escenario, indicando consistencia en el rendimiento entre el algoritmo propio y AWS.
2. Transportar una carga alta:
- En el escenario 10 y 11 con un aumento del 35%, la correlación es 0.957 y 0.956 respectivamente, respecto a los escenarios que utilizaron un menor porcentaje de reentrenamiento.
 - La operación muestra una fuerte correlación entre ambos algoritmos, especialmente en los escenarios finales, es decir, que para esta categoría se requiere además de un porcentaje de reentrenamiento por sobre el 35% de debe tener una data-set variado.
3. Subir o bajar la carga mientras se transporta, Mover en cargas de gran altura con carga desequilibrada y Operación normal
- Estas tres acciones muestra un patrón común, todas encuentran su mayor correlación con un 35% de datos de reentrenamiento, desde el escenario 8 en adelante, es decir, se requiere más variedad de escenario que para la detección de “Subir a operadores en las horquillas” pero , menor la detección de “Detectar una carga en alto”.
 - Nuevamente, hay una fuerte correlación entre el algoritmo propio y AWS para esta operación específica (menor al 3 %).

Análisis de Correlaciones Menos Significativas

1. Bajar pendientes de frente con la grúa cargada:
 - Correlación con el algoritmo propuesto y AWS: 0.6591 y 0.691 respectivamente
 - La operación de bajar pendientes de frente con la grúa cargada muestra una correlación relativamente baja entre el algoritmo propuesto y AWS Rekognition Custom Label, indicando que ambos algoritmos pueden no ser igualmente eficaces para este escenario específico, especialmente al considerar un aumento del 10% en la cantidad de datos de entrenamiento.

2. Operación normal en el escenario 2:
 - Correlación con el algoritmo propuesto y AWS: 0.659 y 0.664 (respectivamente)
 - La operación normal en el escenario 2 también presenta una correlación menos pronunciada entre los dos algoritmos, sugiriendo que la eficacia de ambos puede variar para esta operación en particular en ese escenario específico.

Las correlaciones positivas observadas indican una destacada concordancia entre el algoritmo propuesto y Amazon Rekognition Custom Labels, especialmente en escenarios con mayor variabilidad y un porcentaje de reentrenamiento superior a 35% tiene un notorio. Por otro lado, las correlaciones menos significativas identifican áreas donde los algoritmos podrían no ser igualmente robustos, sugiriendo la necesidad de ajustes adicionales. Estos

resultados destacan la importancia de considerar estas discrepancias al evaluar y mejorar continuamente los algoritmos, asegurando así un rendimiento óptimo en diversas situaciones y operación.

4.5 Significancia del modelo

Podemos notar que en función de F1 scores, porcentaje de reentrenamiento y número de escenarios, no existen diferencias significativas entre el algoritmo propuesto y el de AWS. Sin embargo, la utilización de frameworks, muchas veces no nos permite realizar un análisis exhaustivo de cómo se implementó la red.

Al algoritmo propuesto se basa en métodos de ensamblaje, en la cual en una primera instancia se clasifica objetos y luego las acciones (operación normal o evento de riesgo). Sin embargo, trabajar con muchas variables y datos puede complicar la implementación de estos modelos, debido a que la simple tarea de etiquetar los datos puede tomar mucho tiempo. Debido a que las entradas de nuestro modelo son objetos en una imagen, no basta con realizar pruebas de significancia estadística para reducir la cantidad de variables que nuestros modelos utilizan como entradas. Otro método que nos ayuda a reducir la dimensionalidad de los modelos es utilizar técnicas de clustering sobre las variables. La agrupación no supervisada generalmente se implementa de la siguiente manera: (i) se lleva a cabo un preprocesamiento para ordenar y reescalar los datos, (ii) reducción de dimensionalidad, (iii) algoritmos de agrupación de aplicaciones y uso de métricas para evaluar la calidad de los datos. En general, los resultados del agrupamiento no supervisado dependen en gran

medida de la morfología de los datos analizados porque los algoritmos actúan fuertemente en condiciones de superposición. Por otro lado, la agrupación supervisada se implementa de manera similar a la no supervisada, pero se agrega un paso de procesamiento. La agrupación supervisada primero convierte los datos sin procesar en valores SHAP. Esto implica utilizar los datos sin procesar para entrenar un modelo de aprendizaje automático supervisado y luego calcular los valores SHAP. A partir de este procedimiento, se obtiene una matriz con valores ordenados según qué tan informativos son los datos sobre la variable de salida del modelo. Para obtener un buen análisis de los valores SHAP, es necesaria una variable objetivo-adecuado para entrenar el modelo de clasificación. Hay dos beneficios principales al usar valores SHAP: (i) Sirven como un paso de preprocesamiento que reescala los datos sin procesar en unidades estándar, y (ii) ponderan los datos mediante una medida crítica que enfatiza lo más informativo y minimiza la influencia de lo irrelevante. características. El uso de valores SHAP elimina el ruido de datos de información irrelevante y amplifica las funciones proporcionalmente a la señal que genera. Aquí es donde aparece el valor SHAP como una herramienta que nos permite identificar las relaciones más informativas entre las características de entrada y el resultado previsto, lo que ayuda a explicar lo que está haciendo el modelo. Esto es útil porque los modelos de clasificación suponen implícitamente que todos seguirán comportándose de la misma manera en el futuro, de modo que los patrones de correlación permanecerán constantes.

Al conocer la realidad física del proceso que se está monitoreando, se puede realizar una agrupación supervisada, que aprovecha los valores SHAP para identificar grupos mejor separados a través de una representación más estructurada de los datos. Los valores SHAP son una técnica para deconstruir las predicciones de un modelo de aprendizaje automático en una suma de contribuciones de cada una de sus variables de entrada para comprender el impacto que las variables pueden tener en las predicciones de los modelos de aprendizaje automático. Para esto, el proceso de agrupamiento es crítico porque si bien existen procedimientos "tradicionales" como (i) clasificación y re-escalamiento de datos, (ii) reducción de dimensionalidad y (iii) algoritmos de agrupamiento que utilizan métricas para evaluar la calidad de los clusters.

Para dimensionar otro impacto que tiene el transfer learning en el algoritmo propuesto, se realizara el análisis de valores Shap, con el modelo entrenado y testado con el escenario 1 con el modelo que utiliza un 35% datos de reentrenamiento del escenario 10 y testeado en el escenario 11. La figura 43 muestra los valores Shap para los diferentes objetos etiquetados. De color verde se puede observar la ponderación de los valores Shap del modelo inicial, es decir, modelo entrenado y testado con el escenario 1. Luego en azul de pueden ver los valores Shap de los objetos del modelo en que se hizo Transfer Learning. La principal ventaja del Algoritmo propuesto es que luego de realizar Transfer Learning, además de mejorar el F1-score, nos permite categorizar cuales de los objetos etiquetados inicialmente fueron más significativos para el desempeño del clasificador. En la figura 43, se observa como el modelo al cual se la ha aplicado

Transfer Learning, pondera con mayores valores Shap una menor cantidad de objetos. Esta información es útil para futuros procesos de etiquetados, ya que nos permite reducir la cantidad de etiquetas que hay que identificar por cuadro.

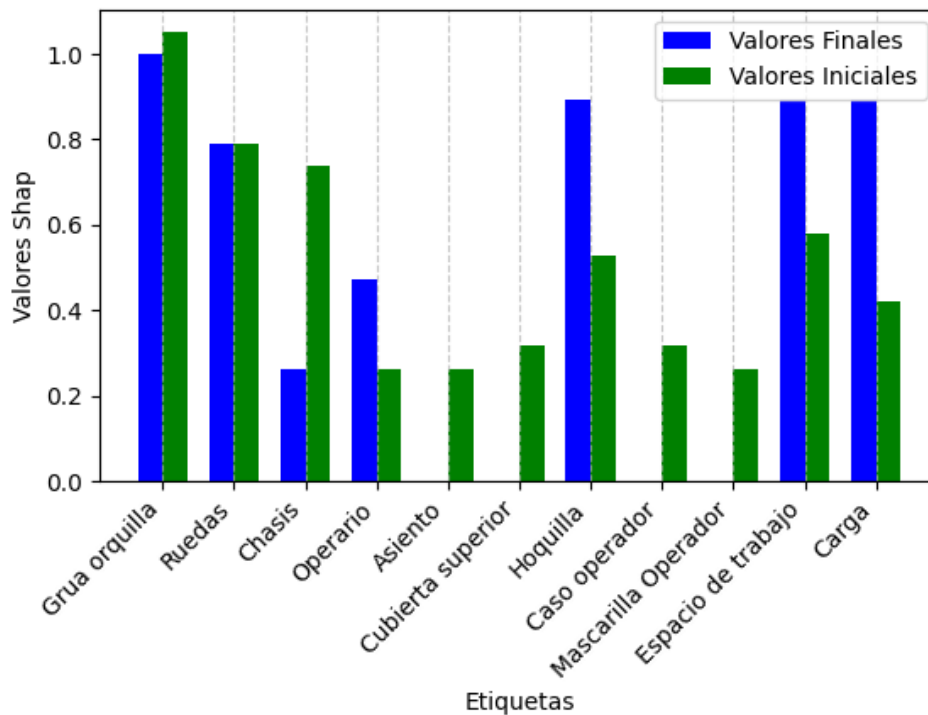


Figura 43. Valores Shap de Algoritmo inicial versus algoritmo con Transfer Learning.

CAPITULO 5. Inferencia de modelos en dispositivos Edge y Cloud.

En capítulos anteriores, analizamos las diferencias en los F1-scores entre nuestro algoritmo propuesto y las herramientas estandarizadas de AWS y DeepStream de NVIDIA. Este capítulo evaluará el modelo después de aplicar Transfer Learning y comparará métricas relevantes al realizar inferencias en dispositivos Edge y en la Nube.

La importancia de evaluar el algoritmo más allá de su F1-score radica en que la detección de eventos de riesgo no se realiza de manera aislada. Es parte de un sistema que debe generar alarmas y alertas oportunas para los operadores y el personal cercano. Por lo tanto, las métricas adicionales relacionadas con el tiempo de respuesta para entregar estas alarmas son cruciales para una evaluación completa del algoritmo.

En la revisión del estado del arte, observamos que el tiempo de alerta en diversas industrias varía según la gravedad del problema que desencadena la alarma, por ejemplo, "llenar con datos". Por lo tanto, el tiempo de respuesta es fundamental. Para evaluar el rendimiento de los algoritmos en dispositivos Edge y en la Nube, utilizaremos métricas que afectan el tiempo desde la recepción de los datos de entrada hasta notificar al usuario sobre la detección del evento de riesgo. Las métricas incluyen: (i) Tiempo de Lectura de Datos de Entrada, (ii) Tiempo de Inferencia y (iii) Tiempo de Despliegue de Datos al Usuario.

- Tiempo de Lectura de Datos de Entrada: Esta métrica mide qué tan rápido el algoritmo puede comenzar a procesar los datos, lo cual es crítico para los sistemas que requieren análisis en tiempo real.

- Tiempo de Inferencia: Esta métrica indica la duración que toma el algoritmo para procesar los datos y hacer predicciones. Tiempos de inferencia más cortos son esenciales para la detección y prevención oportuna de riesgos.
- Tiempo de Despliegue de Datos al Usuario: Esta métrica evalúa la rapidez con la que los resultados se comunican al usuario, lo que es crucial para acciones de respuesta inmediatas.

Para cada implementación, es crucial tener en cuenta la latencia y el rendimiento, ya que estos factores afectan la capacidad del sistema para detectar riesgos y emitir alertas en tiempo real. La Raspberry Pi y el Jetson Nano tienen diferentes capacidades de procesamiento, lo que puede influir en los tiempos de inferencia y despliegue de datos. La selección de hardware adecuado, en este contexto, desempeña un papel clave en la rapidez y efectividad del sistema, especialmente cuando se requiere una respuesta inmediata ante posibles eventos de riesgo.

Las pruebas realizadas en dispositivos Edge (Raspberry Pi y Jetson Nano) permiten evaluar cómo se comporta la inferencia local en comparación con el procesamiento en la nube, que suele tener mayor latencia debido a la transmisión de datos. Este análisis es relevante para determinar la mejor configuración para sistemas de detección de riesgos en tiempo real, ya que la latencia en la nube puede ser una desventaja en situaciones donde la inmediatez es esencial para la seguridad operativa.

El banco de pruebas tiene como objetivo emular el comportamiento de captura de datos, procesamiento, estimación de indicadores y entrega de resultados de

un sistema de detección de eventos de riesgo basado en dispositivos Edge. Este enfoque permite una evaluación completa de los tres modelos diferentes. El banco de pruebas propuesto consta de los siguientes componentes: (i) una cámara Full HD para la adquisición de video, (ii) un sistema embebido para la adquisición de señales e inferencia, y (iii) sistemas de visualización de datos. Debido a la naturaleza diferente de los modelos, es necesario realizar modificaciones menores en el banco de pruebas para implementar estas pruebas.

- La primera implementación presenta una cámara con un sensor Sony IMX219 de 8 megapíxeles, conectada a través de CSI V2 a una Raspberry Pi 5 con un procesador ARM Cortex-A64 y 8GB RAM. La Raspberry Pi está conectada mediante Ethernet a un router con acceso a internet. Esta implementación muestra los datos de dos maneras: primero, a través de un monitor conectado a la Raspberry Pi (localmente), y segundo, a través de una plataforma web. El flujo de trabajo para probar el modelo propuesto se muestra en la Figura 38. En este banco de pruebas, se prueba el algoritmo propuesto.
- La segunda implementación utiliza una cámara con un sensor Sony IMX219 de 8 megapíxeles, conectada a través de CSI V2 a una Raspberry Pi 5. La Raspberry Pi está conectada mediante Ethernet a un router con acceso a internet. Para la inferencia, se envían extractos de video a la nube de AWS utilizando la función "Greenglass" para una transferencia segura desde la Raspberry Pi al bucket de AWS. Luego, el servicio "AWS Rekognition" procesa los datos del bucket en una instancia de Linux. Los

resultados se almacenan en un archivo JSON y están disponibles a través de Greengrass en la Raspberry Pi y mediante una API en el sitio web. El flujo de trabajo para probar el modelo propuesto se muestra en la Figura 8. En este banco de pruebas, se prueba el algoritmo de AWS.

- La tercera implementación es análoga a la primera; la principal diferencia es que, en lugar de usar una Raspberry Pi 5 como dispositivo Edge, se utiliza un Jetson Nano. En este banco de pruebas, se prueba el algoritmo propuesto.
- La cuarta implementación es similar a la tercera, pero en lugar de utilizar el algoritmo propuesto, se utiliza el algoritmo DeepStream.

Las cuatro implementaciones para evaluar los modelos son necesarias debido a la diferente naturaleza de los modelos utilizados para la inferencia. Por lo tanto, es crucial realizar una comparación que vaya más allá del F1-score. Los resultados de esta prueba son el tiempo de ejecución (en segundos) para las implementaciones descritas anteriormente, basados en ventanas de video de un minuto. Es importante señalar que, a pesar de los intentos por estandarizar los entornos de prueba, persisten diferencias debido a las plataformas operativas de cada algoritmo: el algoritmo propuesto y DeepStream se ejecutan en dispositivos edge, mientras que AWS Rekognition opera en la nube. Así, el tiempo medido abarca desde la captura del video de actividades relacionadas con riesgos hasta la visualización del indicador numérico, tanto en el video procesado en la página web como en la pantalla local del dispositivo.

La distinción en el tiempo requerido para mostrar el KPI (Indicador Clave de Rendimiento), presentado como un valor numérico asociado a una marca de tiempo del evento de riesgo detectado, es crucial, al igual que la inclusión de un recuadro delimitador en la imagen de inferencia para resaltar el evento detectado. Esta comparación permite evaluar la eficiencia y adecuación de cada enfoque en contextos de detección y visualización en tiempo real. A continuación, se detalla la implementación de los algoritmos en el banco de pruebas.

5.1 Funcionamiento de algoritmos en pruebas

En capítulos anteriores, analizamos las diferencias en las puntuaciones F1 entre nuestro algoritmo propuesto y las herramientas estandarizadas de AWS y DeepStream de NVIDIA. Este capítulo evaluará el modelo después de aplicar aprendizaje por transferencia y comparará las métricas relevantes al realizar inferencia en dispositivos Edge y en la Nube. La importancia de evaluar el algoritmo más allá de su puntuación F1 radica en el hecho de que la detección de eventos de riesgo no se realiza de forma aislada. Es parte de un sistema que debe generar alarmas y alertas oportunas para los operadores y el personal circundante. Por lo tanto, métricas adicionales relacionadas con el tiempo de respuesta para entregar estas alarmas son cruciales para una evaluación completa del algoritmo. En la revisión del estado del arte, observamos que el tiempo de alerta varía en las distintas industrias según la gravedad del problema que activa la alarma, por ejemplo, "rellenar con datos". Por lo tanto, el tiempo de respuesta es fundamental. Para evaluar el rendimiento de los algoritmos en dispositivos Edge y en la Nube, utilizaremos métricas que afectan el tiempo desde

la recepción de los datos de entrada hasta la notificación al usuario de la detección de un evento de riesgo. Las métricas incluyen (i) Tiempo de Lectura de Datos de Entrada, (ii) Tiempo de Inferencia y (iii) Tiempo de Despliegue de Datos al Usuario.

- Tiempo de Lectura de Datos de Entrada: Esta métrica mide qué tan rápido el algoritmo puede comenzar a procesar los datos, lo cual es crítico para sistemas que requieren análisis en tiempo real.
- Tiempo de Inferencia: Esta métrica indica la duración que el algoritmo toma para procesar los datos y hacer predicciones. Tiempos de inferencia más cortos son esenciales para la detección y prevención oportuna de riesgos.
- Tiempo de Despliegue de Datos al Usuario: Esta métrica evalúa la velocidad a la que los resultados se comunican al usuario, lo cual es crucial para tomar acciones de respuesta inmediata.

En la literatura, diversos estudios (rellenar con datos) destacan la importancia de estas métricas, subrayando su impacto en la efectividad de los sistemas de detección de riesgos en tiempo real.

El banco de pruebas tiene como objetivo emular el comportamiento de captura de datos, procesamiento, estimación de indicadores y entrega de resultados de un sistema de detección de eventos de riesgo basado en dispositivos Edge. Este enfoque permite una evaluación integral de los tres modelos diferentes. El banco de pruebas propuesto comprende los siguientes componentes: (i) una cámara Full HD para adquisición de video, (ii) un sistema embebido para adquisición de señales e inferencia, y (iii) sistemas de visualización de datos. Debido a la

naturaleza distinta de los modelos, son necesarias pequeñas modificaciones en el banco de pruebas para implementar estas pruebas.

- **Implementación 1 (imp 1):** La primera implementación incluye una cámara con un sensor Sony IMX219 de 8 megapíxeles, conectada mediante CSI V2 a una Raspberry Pi 5 con un ARM Cortex-A64 y (especificar RAM). La Raspberry Pi está conectada vía Ethernet a un router con acceso a internet. Esta implementación muestra los datos de dos maneras: primero, a través de un monitor conectado a la Raspberry Pi (localmente) y, segundo, a través de una plataforma web. El flujo de trabajo para probar el modelo propuesto se muestra en la Figura 44. En este banco de pruebas se prueba el algoritmo propuesto.
- **Implementación 2 (imp 2):** La segunda implementación utiliza una cámara con un sensor Sony IMX219 de 8 megapíxeles, conectada mediante CSI V2 a una Raspberry Pi 5 con un ARM Cortex-A64 y (especificar RAM). La Raspberry Pi está conectada vía Ethernet a un router con acceso a internet. Para la inferencia, se envían extractos de video a la Nube de AWS utilizando la función "Greengrass" para la transferencia segura de la Raspberry Pi al bucket de AWS. Luego, el servicio "AWS Rekognition" procesa los datos del bucket en una instancia de Linux. Los resultados se almacenan en un archivo JSON y están disponibles a través de Greengrass en la Raspberry Pi y a través de una API en el sitio web. El flujo de trabajo para probar el modelo propuesto se

muestra en la Figura 45. En este banco de pruebas se prueba el algoritmo de AWS.

- **Implementación 3:** La tercera implementación es análoga a la primera; la principal diferencia es que, en lugar de usar una Raspberry Pi 5 como dispositivo Edge, se utiliza un Jetson Nano. En este banco de pruebas se prueba el algoritmo propuesto.
- **Implementación 4:** La cuarta implementación es análoga a la tercera; pero en lugar de usar el algoritmo propuesto, se utiliza el algoritmo de DeepStream.

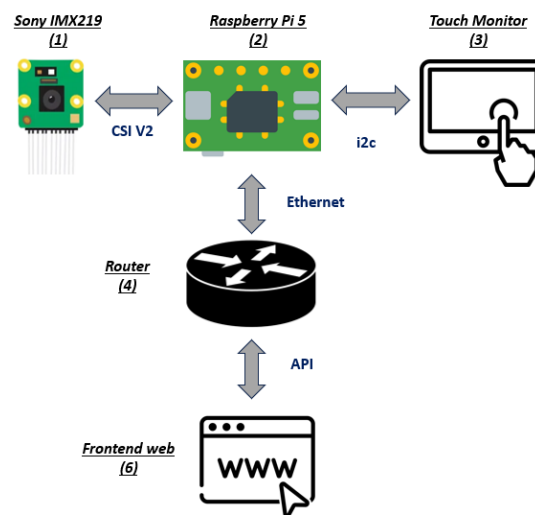


Figura 44. Banco de pruebas que utiliza el algoritmo propuesto.

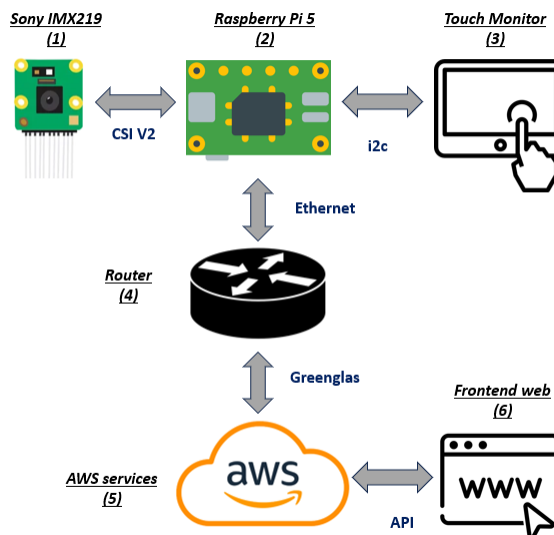


Figura 45. Banco de pruebas que utiliza el algoritmo de AWS.

El empaquetamiento, mostrado en la Figura 46, se implementó tanto para la Raspberry Pi 5 como para la Jetson Nano Orin. Se diseñó y fabricó un contenedor personalizado, impreso en 3D con resina. Además, la cámara se montó en un sistema que permite ajustar su posición angular mediante un servomotor, el cual puede controlarse y monitorearse de manera remota.

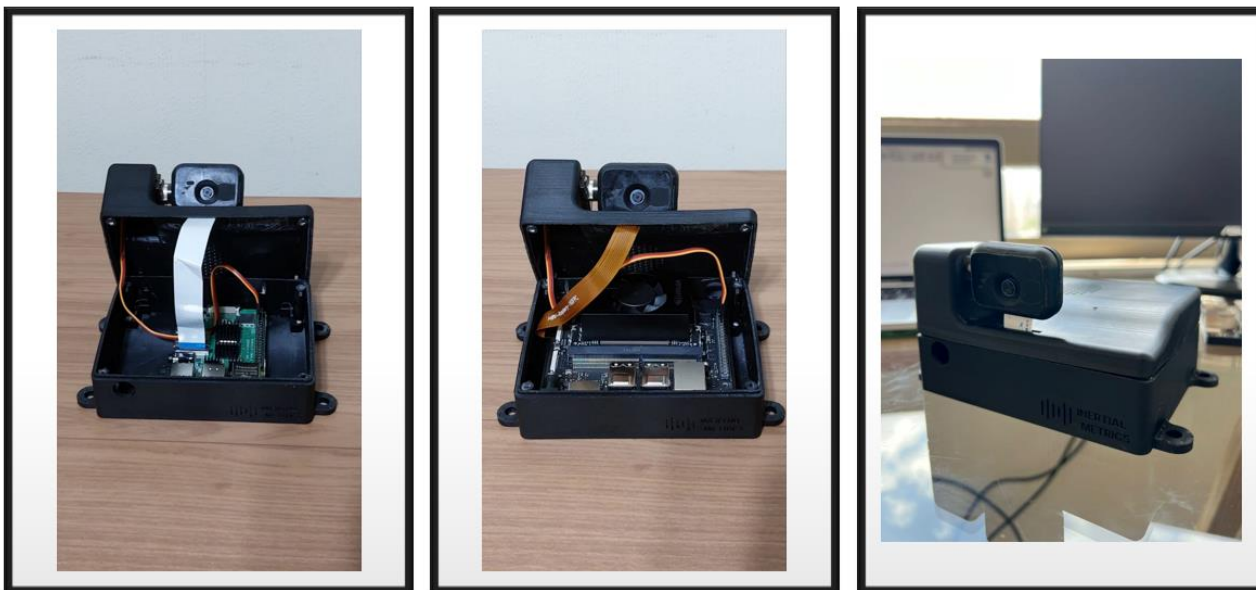


Figura 46. Empaquetamiento de propuesta solución. Imagen de la izquierda corresponde al sistema utilizando una Raspberry pi 5. La imagen del centro corresponde a la utilización de la Jetson Orin. Imagen Derecha es la vista isométrica del sistema.

Las cuatro implementaciones para evaluar los modelos son necesarias debido a la naturaleza distinta de los modelos utilizados para la inferencia. Por lo tanto, es crucial comparar más allá de solo la puntuación F1. La Tabla 22 presenta los resultados de tiempo de ejecución (en segundos) para las implementaciones descritas previamente, basados en ventanas de video de un minuto. Es importante señalar que, a pesar de los intentos de estandarizar los entornos de prueba, persisten diferencias debido a las plataformas operativas de cada algoritmo: el algoritmo propuesto y DeepStream se ejecutan en dispositivos Edge, mientras que AWS Rekognition opera en la nube. Así, el tiempo medido abarca desde la captura del video de actividades de riesgo hasta la visualización del indicador numérico, tanto en el video renderizado en la página web como en la pantalla local del dispositivo. La diferencia en el tiempo requerido para mostrar el KPI, mostrado como un valor numérico asociado a una marca de tiempo del evento de riesgo detectado, es crucial, al igual que la inclusión de un recuadro delimitador en la imagen de inferencia para resaltar el evento detectado. Esta comparación permite evaluar la eficiencia y adecuación de cada enfoque en contextos de detección y visualización en tiempo real. A continuación, detallamos la implementación de los algoritmos en el banco de pruebas.

El algoritmo propuesto requiere una cantidad suficiente de memoria no volátil y RAM, así como la instalación correcta del entorno de uso (Python, biblioteca TensorFlow, Keras y NumPy). Se recomienda encarecidamente ejecutar el algoritmo en un entorno virtual para minimizar los errores debidos a

actualizaciones de bibliotecas. Las pruebas para el algoritmo propuesto se realizan en las implementaciones 1 y 3, siguiendo el siguiente flujo de trabajo:

1. **Captura y Almacenamiento de Video:** Se captura el video desde la cámara y se almacena en la memoria no volátil (microSD) de la Raspberry Pi o Jetson Nano, dependiendo de la implementación.
2. **Procesamiento de Fotogramas:** El video se procesa y se extraen los fotogramas, que luego se almacenan.
3. **Ejecución de Inferencia:** Se realiza la inferencia sobre un conjunto de fotogramas. El algoritmo propuesto requiere dos inferencias: la primera para identificar objetos y la segunda para identificar el tipo de actividad. Los resultados de la inferencia se guardan en formato JSON.
4. **Renderización de Fotogramas:** Los fotogramas se renderizan con los resultados de la inferencia para crear un video con etiquetas.
5. **Visualización de la Información:** La implementación cuenta con dos métodos para mostrar la información. El primero es en un monitor local, dirigido al operador para recibir alertas, como se muestra en la Figura 7 en (3). El segundo es a través de una API, enviando el video etiquetado al frontend de una aplicación web, como se muestra en la Figura 7 en (6).

La inferencia utilizando AWS requiere una conexión a Internet constante, ya que no es posible descargar el modelo entrenado localmente. Esto se debe al modelo de negocio de AWS, que exige pago por consultas, tráfico,

almacenamiento y procesamiento en la nube. Por lo tanto, las pruebas con AWS se realizan en las implementaciones 2 y 4 con el siguiente flujo de trabajo:

1. **Captura y Almacenamiento de Video:** El video se captura desde la cámara y se almacena en la memoria no volátil (microSD) de la Raspberry Pi o Jetson Orin, según la implementación.
2. **Carga del Video:** El video se carga desde la Raspberry Pi o Jetson Orin a AWS mediante el servicio Greengrass, y se almacena en un bucket S3.
3. **Ejecución de Inferencia en EC2:** Una instancia EC2 monitorea el bucket S3 en busca de nuevos videos. Cuando se detecta un nuevo video, se utiliza el modelo almacenado en la función Rekognition Custom Label para realizar la inferencia.
4. **Renderización de Resultados:** Después de la inferencia, la instancia EC2 renderiza el video con los resultados y lo guarda nuevamente en S3. Este paso podría realizarse en el dispositivo para reducir los costos de servicio de AWS, pero para esta investigación se utilizan servicios de AWS para limitar la variabilidad en la comparación.
5. **Visualización de la Información:** Similar a los otros algoritmos, hay dos métodos para mostrar la información. El primero es en un monitor local, que requiere que Greengrass devuelva el video renderizado a la Raspberry Pi o Jetson Orin para su visualización. Esto se muestra en la Figura 8 en (3). El segundo es a través de una API, enviando el video etiquetado al frontend de una aplicación web, como se muestra en la Figura 8 en (6).

Para utilizar el algoritmo DeepStream, se siguen los mismos pasos que con el algoritmo propuesto (los mismos cinco pasos en el flujo de trabajo). La única diferencia es que al usar una Raspberry Pi en lugar de un Jetson Orin, es necesario cambiar el formato del modelo para ejecutarlo en una arquitectura sin núcleos CUDA. TensorRT es una biblioteca de inferencia de NVIDIA diseñada para optimizar y ejecutar redes neuronales profundas. Usando TensorRT, el modelo DeepStream puede exportarse a un formato intermedio y convertirse a TensorFlow Lite. El uso de TensorRT implica etapas de calibración, optimización y exportación del modelo. La calibración ajusta el modelo para funcionar con precisión en el entorno objetivo. La optimización puede incluir técnicas como la cuantización, donde los pesos del modelo se reducen a una representación de menor precisión, mejorando significativamente la eficiencia computacional sin una pérdida sustancial de precisión. Una vez optimizado y convertido, el modelo TensorFlow Lite puede desplegarse en la Raspberry Pi.

Las implementaciones y flujos de trabajo de evaluación descritos anteriormente permiten una evaluación integral de los tres modelos (el algoritmo propuesto, AWS y DeepStream). A pesar de la naturaleza diferente de las implementaciones, cada flujo de trabajo está diseñado para simular un entorno operativo realista, capturando el espectro completo de adquisición de datos, procesamiento, inferencia y visualización de resultados. Aunque no son idénticas, estas diversas implementaciones proporcionan un marco robusto para comparar el rendimiento de los modelos en entornos reales. Los flujos de trabajo garantizan que cada modelo sea probado en condiciones que imitan entornos de

implementación práctica, lo que permite obtener perspectivas útiles y aplicables. Este enfoque integral de evaluación es fundamental para comprender las fortalezas y limitaciones de cada modelo, orientando así el desarrollo de sistemas de detección de eventos de riesgo más efectivos y eficientes en operaciones industriales.

Los resultados en la Tabla 22 destacan el rendimiento superior del algoritmo propuesto en términos de tiempos de respuesta y eficiencia para detectar eventos de riesgo en ventanas de video de un minuto, especialmente en comparación con AWS Rekognition y DeepStream. Cuantitativamente, el algoritmo propuesto en dispositivos edge alcanza tiempos totales de despliegue para visualización web entre 2.15 y 4.22 segundos, superando a DeepStream, que varía de 2.63 a 4.62 segundos, y a AWS Rekognition, que muestra tiempos significativamente mayores entre 6.98 y 9.52 segundos. La eficiencia del algoritmo propuesto también es evidente en la visualización local, donde los tiempos de respuesta oscilan entre 2.05 y 2.93 segundos, mientras que DeepStream muestra tiempos entre 2.57 y 3.21 segundos, y AWS Rekognition se extiende de 7.78 a 10.57 segundos. Estos resultados confirman que el algoritmo propuesto no solo reduce la latencia en comparación con el procesamiento en la nube, sino que también supera a DeepStream en implementaciones edge.

En componentes específicos, el tiempo de inferencia del algoritmo propuesto varía entre 0.56 y 1.21 segundos para clasificaciones binarias y de múltiples eventos, respectivamente, lo cual es inferior o comparable a DeepStream y AWS.

Sin embargo, el tiempo para desplegar datos y videos a través de la API para la visualización edge es notablemente menor con el algoritmo propuesto, oscilando entre 0.51 y 2.4 segundos, en comparación con 0.54 a 2.6 segundos en DeepStream y 1.01 a 3.2 segundos en AWS. La Tabla 22 presenta los tiempos promedio correspondientes a cada tarea evaluada en las distintas implementaciones. Los resultados reportados se basan en un promedio calculado a partir de 50 pruebas realizadas de forma continua para cada implementación. Adicionalmente, en la misma tabla se incluye, entre paréntesis, el valor del coeficiente de variación asociado a las 50 pruebas, lo que permite comparar la variabilidad de los experimentos tanto entre implementaciones como dentro de cada implementación.

Tabla 22. Resultados de tiempo de ejecución de modelos en diferentes implementaciones.

	Tiempo de ejecución (s) implementación 1 (Raspberry Pi 5 con algoritmo propuesto)		Tiempo de ejecución (s) implementación 2 (Raspberry Pi 5 con AWS)		Tiempo de ejecución (s) implementación 3 (Jetson orin con algoritmo propuesto)		Tiempo de ejecución (s) implementación 3 (Jetson orin con DeepStream)	
	clasificación binaria	clasificación de múltiples eventos	clasificación binaria	clasificación de múltiples eventos	clasificación binaria	clasificación de múltiples eventos	clasificación binaria	clasificación de múltiples eventos
Adquisición y almacenamiento de video en la memoria local	0.78 (3.3%)	0.77 (2.9%)	0.77 (1.9%)	0.76 (2.3%)	0.12 (2.9%)	0.11 (3.1%)	0.12 (3.5%)	0.12 (3.3%)
Envío de video a un depósito de AWS	-	-	3.1 (7.2%)	2.9 (6.8%)	-	-	-	-
Analiza el video en fotogramas a 15 fps	0.68 (2.9%)	0.69 (3.7%)	0.23 (4.1%)	0.22 (3.6%)	0.18 (2.8%)	0.19 (2.5%)	0.18 (3.1%)	0.18 (3.3%)
Inferencia	1.21 (2.1%)	2.19 (1.8%)	0.86 (2.8%)	1.41 (2.4%)	0.76 (3.1%)	1.35 (2.2%)	0.56 (2.9%)	1.05 (3.1%)
Renderiza el video con los resultados	2.72 (1.8%)	2.68 (2.3%)	1.01 (1.1%)	1.03 (1.8%)	0.76 (3.4%)	0.79 (3.3%)	0.75 (3.1%)	0.77 (2.9%)
Envía los resultados a través de la API a la interfaz web (datos/video)	0.54/2.1 (2.1%)/(1.1%)	0.51/2.2 (2.7%)/(2.2%)	1.01/3.1 (1.3%)/(2.1%)	0.87/3.2 (2.7%)/(1.3%)	0.52/2.4 (3.3%)/(1.9%)	0.51/2.4 (3.3%)/(1.9%)	0.54/2.6 (3.5%)/(2.1%)	0.51/2.5 (3.1%)/(1.9%)
Visualización de los resultados en el monitor Edge (datos/video)	0.45/1.23 (2.3%)/(1.9%)	0.47/1.19 (2.8%)/(2.7%)	1.81/4.22 (3.1%)/(2.4%)	1.78/4.25 (2.3%)/(1.1%)	0.43/1.11 (2.9%)/(2.8%)	0.45/1.10 (3.4%)/(2.5%)	0.44/1.10 (3.7%)/(2.8%)	0.45/1.09 (4.1%)/(3.1%)
Tiempo total para implementar información en la interfaz web (datos/video)	5.93/7.49	6.84/8.53	6.98/9.07	7.19/9.52	2.34/4.22	2.91/4.8	2.15/3.87	2.63/4.62
Tiempo total de visualización de información en el monitor de borde (datos/video)	5.84/6.62	6.8/7.52	7.78/10.19	8.1/10.57	2.25/2.93	2.85/3.51	2.05/2.71	2.57/3.21

CAPITULO 6. Discusión y Conclusiones

La hipótesis de la investigación plantea que, mediante el uso de modelos de Deep Learning y Transfer Learning en una cantidad limitada de videos de grúas horquillas, es posible construir un modelo generalizado para la detección de eventos de riesgo. Los resultados obtenidos a lo largo de las secciones clave del informe confirman esta hipótesis. A continuación, se explica cómo cada objetivo específico contribuye a validar esta hipótesis:

- **Recopilación y Preprocesamiento de Datos:** La recopilación y preprocesamiento de datos fue fundamental para construir un conjunto de videos etiquetados que representara de manera homogénea las diferentes condiciones de operación y eventos de riesgo en las grúas horquillas. Este proceso no solo permitió disponer de datos de alta calidad para el entrenamiento del modelo, sino que también aseguró que los eventos de riesgo estuvieran claramente definidos y etiquetados de acuerdo con la norma OSHA 3949. El éxito de este objetivo es evidente en los altos valores de F1 obtenidos en las pruebas de validación, que reflejan que el modelo fue capaz de identificar eventos de riesgo en condiciones reales y variadas. La precisión en la preparación de estos datos, como se observa en la sección 4.3, fue clave para que el modelo pudiera generalizar y reconocer patrones específicos de riesgo con pocos datos.

- **Desarrollo de Modelos de Deep Learning Supervisado:** Este objetivo se centró en construir algoritmos de Deep Learning que pudieran distinguir entre eventos de riesgo y condiciones normales de operación de las grúas horquillas, cumpliendo con los requisitos de la normativa de seguridad OSHA 3949. La implementación de un modelo de clasificación supervisado permitió categorizar eventos complejos con alta precisión, como muestran los resultados de la sección 4.2 y 4.3. Además, se logró una clasificación multinivel que diferenciaba varios tipos de eventos de riesgo específicos, lo cual respalda directamente la hipótesis, ya que demuestra que los modelos de Deep Learning desarrollados pueden adaptarse a datos acotados, cumpliendo así con la tarea de detección generalizable.
- **Implementación de Transfer Learning:** El uso de Transfer Learning permitió utilizar modelos preentrenados y adaptarlos a la tarea específica de detectar eventos de riesgo en grúas horquillas. Esto fue clave para reducir la necesidad de datos extensos, ya que el modelo pudo aprender de patrones preexistentes en tareas similares y aplicar ese conocimiento a un contexto nuevo. Los resultados en la sección 4.2 demuestran que los modelos basados en Transfer Learning alcanzaron altos niveles de precisión y recall, incluso con un conjunto de datos reducido, lo que valida la hipótesis. Esto indica que la técnica de Transfer Learning es altamente efectiva para construir un modelo que detecte riesgos en maquinaria crítica sin requerir grandes cantidades de datos de entrenamiento específicos.

- **Evaluación y Validación de Modelos:** La evaluación y validación de los modelos en términos de su capacidad para detectar eventos de riesgo se abordó en detalle en las secciones 4.3, 4.4, y 4.5. Los modelos lograron cumplir con un F1-score superior al umbral establecido de 0.85, lo cual confirma que son efectivos y generalizables para la detección de riesgos. Además, el análisis de correlación y la significancia estadística de los resultados refuerzan la confianza en que el modelo es aplicable a escenarios industriales diversos, lo que es fundamental para validar la hipótesis. Esta fase de evaluación confirma que los modelos basados en Deep Learning y Transfer Learning cumplen con los requisitos de precisión y generalización necesarios para detectar eventos de riesgo en condiciones operativas reales.

Los resultados del proyecto también cumplen con cada uno de los objetivos específicos propuestos, de la siguiente manera:

- **Recopilación y Preprocesamiento de Datos:** La recopilación de videos y su preprocesamiento fue un paso esencial que permitió obtener un conjunto de datos confiable y representativo de las condiciones de riesgo en grúas horquillas. La homogeneidad y la precisión en la etiquetación, según OSHA 3949, aseguraron que el modelo tuviera una base sólida para su entrenamiento y pruebas. Los datos de alta calidad permitieron reducir la influencia de factores externos como la iluminación y los ángulos de

grabación, aspectos abordados en la sección 3.5 de evaluación de modelos. Este objetivo se logró satisfactoriamente, ya que el modelo fue capaz de generalizar y mantener su precisión en condiciones operativas variadas.

- **Desarrollo de Modelos de Deep Learning Supervisado:** El diseño de un modelo de Deep Learning supervisado fue fundamental para que el sistema pudiera identificar y clasificar eventos de riesgo. Los resultados en la clasificación binaria y multiestado (secciones 4.3 y 4.4) muestran que el modelo no solo puede distinguir entre situaciones de riesgo y no riesgo, sino también entre tipos específicos de riesgos. Este éxito es clave para demostrar que el enfoque de Deep Learning es adecuado para esta tarea, y cumple con el objetivo de construir un sistema que clasifique los eventos según las normas de seguridad. Este logro refuerza la hipótesis de que es factible construir un modelo generalizable para detectar riesgos en grúas horquillas.
- **Implementación de Transfer Learning:** La implementación de Transfer Learning permitió adaptar un modelo preentrenado a una nueva tarea con un conjunto de datos específico pero limitado. En la sección 4.2 se evidencia que el modelo adaptado mediante Transfer Learning mantuvo una alta precisión y capacidad de generalización. Este objetivo se cumplió, ya que Transfer Learning resultó ser una estrategia efectiva para reducir la necesidad de datos de entrenamiento extensivos, confirmando así la viabilidad de crear un modelo generalizable bajo la hipótesis propuesta.

- **Evaluación y Validación de Modelos:** La evaluación exhaustiva de los modelos en términos de F1-score y precisión en múltiples escenarios de prueba confirmó que el modelo es efectivo para detectar eventos de riesgo. Los altos puntajes en las métricas de evaluación (secciones 4.4 y 4.5) indican que el modelo no solo cumple con los estándares de rendimiento deseados, sino que también es confiable para aplicaciones industriales. Además, los resultados de funcionamiento en sistemas Edge en el capítulo 5 aseguran que el modelo puede operar en tiempo real, lo que es crucial para un sistema de detección de riesgos en entornos industriales. Este cumplimiento reafirma que el modelo puede aplicarse a condiciones de operación reales, tal como plantea la hipótesis.
- **Algoritmo Propuesto vs. NVIDIA DeepStream SDK:** La implementación con NVIDIA DeepStream SDK demostró ser la más rápida en cuanto a tiempo de ejecución, aprovechando su arquitectura de hardware optimizada para inferencias rápidas en el procesamiento de imágenes [102] [103]. Sin embargo, debido a que su enfoque se basa en la extracción de características generales de la imagen completa, sin realizar una identificación detallada de objetos, su capacidad de generalización y precisión en la clasificación de actividades complejas resultó inferior en comparación con el algoritmo propuesto y AWS Rekognition. A diferencia de DeepStream, nuestro enfoque y el de AWS emplean un "stacking" de redes neuronales, combinando CNNs y LSTMs, lo que permite detectar actividades a través de la identificación previa de objetos específicos y el

análisis de la secuencia temporal de estas detecciones [104] [105] [106]. Este enfoque ha demostrado ser especialmente efectivo en aplicaciones donde la detección precisa de actividades humanas o de máquinas operadas por humanos es fundamental, al integrar tanto características temporales como espaciales.

Estudios recientes en el estado del arte destacan que la integración de características temporales es esencial para una clasificación precisa en escenarios industriales complejos, donde la interacción de objetos y la secuencia de actividades juegan un rol crítico [107]. Trabajos muestran que los modelos híbridos CNN-LSTM, que integran características tanto temporales como espaciales, permiten una detección más precisa en el monitoreo de seguridad ocupacional en tiempo real. Además, los enfoques de aprendizaje profundo híbrido, que utilizan estrategias como el transfer learning, han demostrado mejorar la capacidad de generalización en contextos de datos limitados, optimizando la adaptación a distintos entornos y tareas [108] [109].

Mientras que DeepStream es ventajoso en términos de velocidad y eficiencia de implementación, su enfoque de extracción de características globales limita su precisión en entornos donde el análisis temporal es crucial. La metodología de stacking, como la empleada en el algoritmo propuesto, permite una interpretación más detallada de las actividades mediante la integración de múltiples capas de análisis, ofreciendo una

mayor precisión y adaptabilidad en aplicaciones de monitoreo industrial y detección de riesgos [110].

Una de las ventajas clave del algoritmo propuesto es que, al contar con un control total sobre la arquitectura del modelo, permite un acceso detallado y transparente a la información generada por el modelo. Esta capacidad es fundamental para aplicaciones de seguridad ocupacional, ya que permite extraer interpretaciones más precisas sobre el comportamiento del sistema y sus decisiones en la detección de eventos de riesgo. Por ejemplo, al utilizar valores SHAP, es posible identificar qué características específicas influyen en las decisiones del modelo, ofreciendo una interpretación clara y cuantitativa de cómo se clasifican los eventos [111]. Esto es especialmente valioso en entornos industriales donde entender los factores que disparan alertas de seguridad es crucial para la validación y ajuste del sistema según las condiciones operativas reales.

En contraste, al emplear plataformas como NVIDIA DeepStream o servicios como AWS Rekognition, el modelo funciona mayormente como una "caja negra," lo que limita el nivel de transparencia y explicabilidad del sistema. Estos servicios, aunque eficientes en términos de procesamiento y facilidad de implementación, no ofrecen un acceso directo a los procesos internos del modelo, dificultando la interpretación detallada de sus decisiones. En el caso de AWS Rekognition y NVIDIA DeepStream, los usuarios reciben las predicciones finales sin un desglose de los factores o

características que influyen en esas predicciones, lo que limita la capacidad de ajustar y personalizar el modelo a entornos y necesidades específicas de seguridad. La capacidad de interpretabilidad y control del algoritmo propuesto no solo permite una optimización continua del modelo, sino que también respalda la confianza y adoptabilidad en aplicaciones críticas donde la seguridad de los trabajadores depende de una comprensión clara y ajustable de los mecanismos de detección de riesgos.

La importancia de esta investigación radica en su capacidad para preentrenar modelos con datos de incidentes críticos que no pueden ser replicados intencionalmente, lo que permite construir un sistema robusto y generalizable que puede ser aplicado en múltiples entornos de operación. La posibilidad de adaptar estos modelos mediante Transfer Learning reduce la necesidad de datos específicos y evita la recreación de escenarios peligrosos, facilitando la implementación de un sistema de monitoreo que protege la seguridad de los trabajadores en condiciones reales.

Además, el enfoque propuesto de detección de eventos de riesgo mediante Deep Learning y Transfer Learning representa una contribución significativa para la industria, ya que reduce los costos de recolección de datos y optimiza el proceso de entrenamiento del modelo. Al mismo tiempo, permite implementar sistemas de monitoreo que identifican y clasifican eventos de riesgo en tiempo real, lo que es crucial para prevenir accidentes y reducir los tiempos de inactividad en operaciones críticas.

La investigación confirma la hipótesis inicial, ya que los modelos de Deep Learning y Transfer Learning desarrollados lograron detectar eventos de riesgo de manera generalizable con una cantidad limitada de videos de grúas horquillas. Cada uno de los objetivos específicos contribuyó de manera integral a la validación de esta hipótesis: desde la preparación rigurosa de los datos y el desarrollo de algoritmos efectivos, hasta la implementación de Transfer Learning y la exhaustiva evaluación del modelo en condiciones variadas. Los resultados obtenidos, especialmente en términos de precisión y capacidad de generalización, aseguran que el enfoque planteado es viable y efectivo para la detección de riesgos en maquinaria crítica, cumpliendo tanto con los objetivos como con la hipótesis de la investigación.

REFERENCIA

- [1] F. W. Boelter, J. W. Spencer y C. E. Simmons, «Heavy Equipment Maintenance Exposure Assessment: Using a Time-Activity Model to Estimate Surrogate Values for Replacement of Missing Data,» *Journal of Occupational and Environmental Hygiene*, vol. 4, nº 7, pp. 525-537, 2007.
- [2] Minig Shared Value, «New Tech New Deal, Technology Impacts Review,» Winnipeg, 2019.
- [3] S. Vaidya, S. Bhosle y P. Ambad, «Industry 4.0 – A Glimpse,» *Procedia Manufacturing*, vol. 20, pp. 233-238, 2018.
- [4] P. Zheng, H. Wang, Z. Sang, . R. Y. Zhong, . Y. Liu, C. Liu, K. Mubarok y S. Yu, «Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives,» *Frontiers of Mechanical Engineering volume*, vol. 13, pp. 137-150, 2018.
- [5] S. Z. ., Z. D. Xu Zhu, «Deep learning based reference model for operational risk evaluation of screw chillers for energy efficiency,» *Ernergy*, vol. 213, 2020.
- [6] S. W. A. S. Tingting Huang, «Highway crash detection and risk estimation using deep learning,» *Accident Analysis & Prevention*, vol. 135, 2020.
- [7] R. Erices, «Estimación del Estado de Vigilia, a Partir de Imágenes de Video,» Universidad de Concepción, Concepción, 2015.

- [8] A. Nishajith, J. Nivedha, S. S. Nair y J. M. Shaffi, «Smart Cap - Wearable Visual Guidance System for Blind,» de *International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2018.
- [9] A. Hengle, A. Kulkarni, N. Bavadekar, N. Kulkarni y R. Udyawar, «Smart Cap: A Deep Learning and IoT Based Assistant for the Visually Impaired,» de *Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2020.
- [10] H. Yoo y U. Haa, «An EEG-NIRS Ear-Module SoC for Wearable Drowsiness Monitoring System,» de *IEEE Asian Solid-State Circuits Conf.*, Japón, 2016.
- [11] C. C. Charles Anthony Bates, «Engineering Readiness: How the TRL Figure of Merit Coordinates Technology Development,» *Engineering Studies* , vol. 12, nº 1, 2020.
- [12] IMI Sensors, «Vibratory Screens & feeders,» 2016.
- [13] Metso, «ScreenWatch,» Harrisburg, 2016.
- [14] D. C. Volante, «Condition Monitoring for Rotational Machinery,» 2011.
- [15] Z. G. X. Dai, «om Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis,» *IEEE Transactions on Industrial Informatics*, vol. 9, nº 4, pp. 2226 - 2238, 2013.
- [16] P. Aqueveque, L. Radrigan, F. Pastene, A. S. Morales y E. Guerra, «Data-Driven Condition Monitoring of Mining Mobile Machinery in Non-Stationary

- Operations Using Wireless Accelerometer Sensor Modules,» *IEEE Access*, vol. 9, pp. 17365 - 17381, 2021.
- [17] S. Madzhov, «Research on the Reliability Level of Backhoe Loaders,» *Journal of Environmental Science and Engineering*, pp. 134-138, 2019.
- [18] S. Madzhov, «Research and Follow up of Results on the Reliability Level of Backhoe Loaders,» *Journal of Environmental Science and Engineering*, pp. 82-86, 2020.
- [19] M. A. Khan y K. Salah, «IoT security: Review, blockchain solutions, and open challenges,» Zakariya University, Sharjah, 2018.
- [20] S. Madakam, R. R y T. Siddharth, «Internet of Things (IoT): A Literature,» *Journal of Computer and Communications*, Mumbai,, 2015.
- [21] V. Sharma y T. Ravi , «A review paper on “IOT” & It’s Smart Applications,» *International Journal of Science, Engineering and Technology Research (IJSETR)*, Chattisgarh, 2016.
- [22] G. WANG, M. NIXON y M. BOUDREAUX, «Toward Cloud-Assisted Industrial IoT Platform for Large-Scale Continuous Condition Monitoring,» *Proceedings of the IEEE* , vol. 107, nº 6, pp. 1193 - 1205, 2019.
- [23] J. Wu, S. Guo , J. Li y D. Zeng, «Big Data Meet Green Challenges: Big Data Toward Green Applications,» *IEEE Systems Journal*, vol. 10, nº 3, pp. 888 - 900, 2016.
- [24] J. Wu, S. Guo, J. Li y D. Zeng, «Big Data Meet Green Challenges: Greening Big Data,» *IEEE Systems Journal*, vol. 10, nº 3, 2016.

- [25] F. L.-F. T. N. ,. R. J.-G. Sudeep Ghimire, «IoT based situational awareness framework for real-time project management,» *International Journal of Computer Integrated Manufacturing*, vol. 30, nº 1, pp. 74-83, 2017.
- [26] M. A. R. a. M. A. H. SK Alamgir Hossain a, «Edge computing framework for enabling situation awareness in IoT based smart city,» *J. Parallel Distrib. Comput.*, pp. 226-237, 2018.
- [27] ,. M. Z. O. K. A. G.-d.-P. B.-P. Guadalupe Ortiz, «Atmosphere: Context and situational-aware collaborative IoT architecture for edge-fog-cloud computing,» *Computer Standards & Interfaces*, vol. 79, 2022.
- [28] Amazon, «<https://aws.amazon.com/>,» [En línea]. Available: <https://aws.amazon.com/es/rekognition/>. [Último acceso: 3 03 2022].
- [29] Microsoft, «<https://azure.microsoft.com/>,» [En línea]. Available: <https://azure.microsoft.com/es-es/overview/ai-platform/>. [Último acceso: 3 03 2022].
- [30] google, «cloud.google.com,» [En línea]. Available: <https://cloud.google.com/solutions/ai?hl=es>. [Último acceso: 3 03 2022].
- [31] G. F. J. W. L. I. & P.-A. M. Hassan Ismail Fawaz, «Deep learning for time series classification: a review,» *Data Mining and Knowledge Discovery* , vol. 33, p. 917–963, 2019.
- [32] H. Torres-Carrión, C. Solano-Chamba, C. Narváez-Guillen y M. Cueva-Hurtado, «IoT security issues in the context of Edge Computing: A

Systematic Review of Literature,» de *17th Iberian Conference on Information Systems and Technologies (CISTI)*, Madrid, 2022.

- [33] B. Ali, M. A. Gregory y S. Li, «Multi-Access Edge Computing Architecture, Data Security and Privacy: A Review,» *IEEE Access* , vol. 9, pp. 18706 - 18721, 2021.
- [34] M. Asim, Y. Wang, K. Wang y P.-Q. Huang, «A Review on Computational Intelligence Techniques in Cloud and Edge Computing,» *IEEE Transactions on Emerging Topics in Computational Intelligence* , vol. 6, nº 4, 2020.
- [35] J. Ren, G. Yu, Y. He y G. Y. Li, «Collaborative Cloud and Edge Computing for Latency Minimization,» *IEEE Transactions on Vehicular Technology* , vol. 68, nº 5, 2029.
- [36] P. B. ., a. E. Z. Michele Compare, «Challenges to IoT-Enabled Predictive Maintenance for Industry 4.0,» *IEEE INTERNET OF THINGS JOURNAL*, vol. 7, nº 5, pp. 4585-4597, 2020.
- [37] L. M. C. W. T. G. I. W. G. F. M. S. I. & C. Navid Mohammadi Foumani, «Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey,» *Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey*, vol. 56, nº 9, 2024 .
- [38] C.-L. Liu, W.-H. Hsaio y Y.-C. Tu, «Time Series Classification With Multivariate Convolutional Neural Network,» *IEEE Transactions on Industrial Electronics*, vol. 66, nº 6, 2019.

- [39] M. E. A. H. M. a. M. W. F. Slouki, «Bearings prognostic using Mixture of Gaussians Hidden Markov Model and Support Vector Machine,» de *ACS International Conference on Computer Systems and Applications (AICCSA)*, Ifrane, 2013.
- [40] S. S. a. z. H. X. Chen, «Remaining life prognostics of rolling bearing based on relative features and multivariable support vector machine,» *Journal of Mechanical Engineering Science*, vol. 12, n° 227, pp. 2849-2860, 2013.
- [41] M. H. a. L. M. Z. Sikorska, «Prognostic modelling options for remaining useful life estimation by industry,» *Mechanical Systems and Signal Processing*, vol. 25, n° 5, pp. 1803-1833, 2011.
- [42] M. F. Huber, «recursive Gaussian process: On-line regression and learning,» *Pattern Recognition Letters*, vol. 45, n° 1, pp. 85-91, 2014.
- [43] A. S. T. I. M. K. A. S. A. A. O. A. S. A. Adi Alhudhaif, «A Particle Swarm Optimization Based Deep Learning Model for Vehicle Classification,» *Computer Systems Science & Engineering*, vol. 40, n° 1, pp. 223-235, 2021.
- [44] MATLAB inc, «Predictive Maintenance with MATLAB,» MathWorks, 2020.
- [45] Matlab inc, «Introducing Machine Learning,» MathWorks, 2016.
- [46] Q. S. M. L. X. Y. L. L. & J. Z. Ya-Lin Zhang, «A Classification Based Ensemble Pruning Framework with Multi-metric Consideration,» de *Intelligent Systems and Applications*, Springer, 2021, p. 650–667.

- [47] H. Wu y D. Levinson, «The ensemble approach to forecasting: A review and synthesis,» *Transportation Research Part C: Emerging Technologies*, vol. 132, 2021.
- [48] D. W. C. H. a. J. T. Z. Li, «An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction,» *Rel. Eng. Syst. Safety*, vol. 184, pp. 110-122, 2019.
- [49] K. G. a. D. W. Z. Li, «Degradation modeling and remaining useful life prediction of aircraft engines using ensemble learning,» *J. Eng. Gas Turbines Powe*, vol. 141, nº 4, 2019.
- [50] N. S. N. S. a. V. B. S. Dua, «Multi-input CNN-GRU based human activity recognition using wearable sensors,» *Computing*, pp. 1-18, 2021.
- [51] J. H. k. L. Y. L. X. Shao Haidong, «"A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders,» *Mech. Syst. Signal Process*, Vols. %1 de %2102,, pp. pp. 278-297 , 2018.
- [52] O. Gungor, T. S. Rosing y B. Aksanli, «DOWELL: Diversity-Induced Optimally Weighted Ensemble Learner for Predictive Maintenance of Industrial Internet of Things Devices,» 3125 - 3134, vol. 9, nº 4, pp. 3125 - 3134, 2021.
- [53] W. H. S. A. Ahmed Elaraby, «Classification of Citrus Diseases Using Optimization Deep Learning Approach,» *Computational Intelligence and Neuroscience* , 2022.

- [54] H. Y. ., W. ., L. a. Y. W. Xiaohui Shang, «Secrecy Performance Analysis of Wireless Powered Sensor Networks Under Saturation Nonlinear Energy Harvesting and Activation Threshold,» *sensors*, vol. 20, n° 6, 2020.
- [55] L. Ren, Z. Jia, Y. Laili y D. Huang, «Deep Learning for Time-Series Prediction in IIoT: Progress, Challenges, and Prospects,» *IEEE Transactions on Neural Networks and Learning Systems* , 2023.
- [56] «Deep Learning for the Industrial Internet of Things (IIoT): A Comprehensive Survey of Techniques, Implementation Frameworks, Potential Applications, and Future Directions,» *Shahid Latif ,Maha Driss,Wadii Boulila,,Zil e Huma ,Sajjad Shaukat Jamal,Zeba Idrees and Jawad Ahmad* , vol. 21, n° 22, 2021.
- [57] B. N. Y. S.-L. L. Aria Ghora Prabono, «Hybrid domain adaptation with deep network architecture for end-to-end cross-domain human activity recognition,» *Computers & Industrial Engineering*, vol. 151, 2021.
- [58] H. C. (. Z. a. X. W. Lei Hou, «Deep Learning-Based Applications for Safety Management in the AEC Industry: A Review,» *applied sciences*, vol. 11, n° 2, 2021.
- [59] H. U. R. S. A. R. M. A. R. F. R. I. A. Muhammad Imran, «A performance overview of machine learning-based defense strategies for advanced persistent threats in industrial control systems,» *Computers & Security*, vol. 134, 2023.

- [60] A. E. Journal, «Deep learning-based intrusion detection approach for securing industrial Internet of Things,» *Alexandria Engineering Journal*, vol. 81, pp. 371-383, 2023.
- [61] H. Z. X. Z. S. L. W. L. Lei Wang, «Transfer Learning for Music Classification and Regression Tasks Using Artist Tags,» de *Proceedings of the 7th Conference on Sound and Music Technology (CSMT)* , 2019.
- [62] F. S. T. K. W. Z. C. Y. & C. L. Chuanqi Tan, «A Survey on Deep Transfer Learning,» de *Artificial Neural Networks and Machine Learning – ICANN 2018*, Rhodes, Greece, 2018.
- [63] B. Maschler y M. Weyrich, «Deep Transfer Learning for Industrial Automation: A Review and Discussion of New Techniques for Data-Driven Machine Learning,» *IEEE Industrial Electronics Magazine* , vol. 15, nº 2, 2021.
- [64] Y. W. B. C. & N. Z. Xincheng Cao, «Domain-adaptive intelligence for fault diagnosis based on deep transfer learning from scientific test rigs to industrial applications,» *Neural Computing and Applications* , vol. 33, p. 4483–4499, 2020.
- [65] A. M. Roy, «Adaptive transfer learning-based multiscale feature fused deep convolutional neural network for EEG MI multiclassification in brain–computer interface,» *Engineering Applications of Artificial Intelligence*, vol. 116, 2022.

- [66] M. V. S. L. & S. L. Bayu Adhi Tama, «Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals,» *Artificial Intelligence Review*, vol. 56, p. 4667–4709, 2022.
- [67] R. H. ., J. L. ., Y. L. ., Z. C. ., G. H. ., R. Y. K. G. Weihua Li, «A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges,» *Mechanical Systems and Signal Processing*, vol. 167, 22.
- [68] P. Sharma, «Understanding Transfer Learning for Deep Learning,» Analytics Vidhya, [En línea]. Available: <https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/>. [Último acceso: 07 07 2024].
- [69] H. R. A. a. K. R. Mohammadreza Iman, «A Review of Deep Transfer Learning and Recent Advancements,» *technologies*, vol. 11, nº 2, 2023.
- [70] S. P. a. G. K. Jaya Gupta, «Deep Learning (CNN) and Transfer Learning: A Review,» de *Journal of Physics: Conference Series*, online, 2022.
- [71] B. M. N. H. E. E. Daniel Fernando Santos-Bustos, «Towards automated eye cancer classification via VGG and ResNet networks using transfer learning,» *Engineering Science and Technology, an International Journal*, vol. 35, 2022.
- [72] X. L. & L. G. Long Wen, «A transfer convolutional neural network for fault diagnosis based on ResNet-50,» *Neural Computing and Applications*, vol. 32, p. 6111–6124, 2020.

- [73] A. S. B. Reddy y D. S. Juliet, «Transfer Learning with ResNet-50 for Malaria Cell-Image Classification,» de *2019 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India.
- [74] G. Vrbančič y V. Podgorelec, «Transfer Learning With Adaptive Fine-Tuning,» *IEEE Access*, vol. 8, pp. 2169-3536, 2020.
- [75] T. O. & A. Basturk, «Static facial expression recognition using convolutional neural networks based on transfer learning and hyperparameter optimization,» *Multimedia Tools and Applications* , vol. 79, p. 26587–26604, 2020.
- [76] N. P. L. J. B. M. B. A. a. S. V. Malliga Subramanian, «Hyperparameter Optimization for Transfer Learning of VGG16 for Disease Identification in Corn Leaves Using Bayesian Optimization,» *Special Section on Big Data Analytics for Agricultural Disaster Management*, pp. 215-229., 2022.
- [77] M. & D. W. Wang, «Deep visual domain adaptation,» *Neurocomputing*,, vol. 312, 2018.
- [78] E. H. J. S. K. & D. Tzeng, «Adversarial discriminative domain adaptation,» de *Conference on Computer Vision and Pattern Recognition*, 2017.
- [79] G. I. K. R. P. J. L. K. C. & W. Parisi, «Continual lifelong learning with neural networks:,» *Neural Networks*, , vol. 113, pp. 54-71, 2019.
- [80] G. S. P. P. Inturi Vamsi †, «Comparison of condition monitoring techniques in assessing fault severity for a wind turbine gearbox under non-stationary loading,» *Mechanical Systems and Signal Processing*, pp. 1-20, 2019.

- [81] C.-M. L. G.-D. O, «Nonlinear model for condition monitoring of non-stationary vibration signals in ship drive line application,» *Mechanical Systems and Signal Processing*, vol. 44, pp. 134-148, 2014.
- [82] FULLEN, «<https://www.fullen.cl/>,» [En línea]. Available: <https://www.fullen.cl/blog/elegir-gruas-horquillas/>. [Último acceso: 26 05 2022].
- [83] K. K. Krogerus T., «TESTING OF DATA ANALYSIS PROCESS IN HYDRAULIC FORKLIFT,» de *The Twelfth Scandinavian International Conference on Fluid Power*, Tampere, 2011.
- [84] S.-w. H. K.-y. J. Jea-yun Lim, «IoT Based Forklift Realtime Monitoring System Development,» *Journal of Digital Contents Society*, vol. 21, nº 1, pp. 237-244, 2020.
- [85] T. J. L. ,. I. J. J. L. Tim Horberry, «Forklift safety, traffic engineering and intelligent transport systems: a case study,» *Applied Ergonomics*, vol. 35, pp. 575-581, 2004.
- [86] S. A. J. O. Minji Choia, «VR-Based investigation of forklift operator situation awareness for preventing collision accidents,» *Accident Analysis & Prevention*, vol. 136, 2020.
- [87] M. A. C. C. D. F. F. F. G. P. H. h. J. K. L. R. T. T. Matthew R. Walter, «A Situationally Aware Voice-commandable Robotic Forklift Working Alongside People in Unstructured Outdoor Environments,» *Journal of Fields robotics*, 2014.

- [88] K. A. S. N. A. T. V. K. S N Kostarev, «Development of automated monitoring safety system of the forklift DP 3510,» de *IOP Conference Series: Materials Science and Engineering*, Tyumen, 2019.
- [89] D. A. Reda Yacoub, «Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models,» *BMC Genomics* , vol. 21, nº 6, p. 79–91, 2020.
- [90] S. C. Yadavendra, «A comparative study of breast cancer tumor classification by classical machine learning methods and deep learning method,» *Machine Vision and Applications volume*, vol. 31, nº 46, 2020.
- [91] Ž. Đ. Vujović, «Classification Model Evaluation Metrics,» *International Journal of Advanced Computer Science and Applications*, vol. 12, nº 6, 2021.
- [92] M. L. & S. M. Nour Eldeen Khalifa, «A comprehensive survey of recent trends in deep learning for digital images augmentation,» *Artificial Intelligence Review*, vol. 55, p. 2351–2377, 2022.
- [93] nvidia, «<https://developer.nvidia.com/>,» [En línea]. Available: <https://developer.nvidia.com/deepstream-sdk>. [Último acceso: 2 03 2023].
- [94] amazon, «aws.amazon.com,» [En línea]. Available: <https://aws.amazon.com/es/rekognition/?nc=sn&loc=0>. [Último acceso: 10 1 2023].
- [95] gstreamer, «gstreamer.freedesktop.org,» [En línea]. Available: <https://gstreamer.freedesktop.org/>. [Último acceso: 5 3 2023].

- [96] Tzutalin, «<https://github.com/>,» <https://github.com/>, 2015. [En línea]. Available: <https://github.com/heartexlabs/labellmg>. [Último acceso: 23 2022].
- [97] L. S. G. S. Jie Hu, «Squeeze-and-Excitation Networks,» de *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake city, 2018.
- [98] Amazon Web Services, Inc, «<https://aws.amazon.com/es/>,» 2023. [En línea]. Available: <https://aws.amazon.com/es/rekognition/custom-labels-features/>.
- [99] NVIDIA Corporation, «<https://developer.nvidia.com/deepstream-sdk>,» 2023. [En línea].
- [100] Y. L. & D. Z. Michael Bernico, «Investigating the Impact of Data Volume and Domain Similarity on Transfer Learning Applications,» de *Proceedings of the Future Technologies Conference (FTC) 2018* , 2018.
- [101] A. S. M. Ayse Betul Cengiz, «How much data do I need? A case study on medical data,» *arXiv*, 2023.
- [102] H. Farias, M. Solar y D. Ortiz, «SmartLens: Enhancing Factory Safety through Nvidia Deepstream-powered Camera-Assistive Surveillance and Analysis,» de *2021 Eighth International Conference on eDemocracy & eGovernment (ICEDEG)*, Quito, 2021.
- [103] T. Rangunthar y Y. P. Singh, «SmartLens: Enhancing Factory Safety through Nvidia Deepstream-powered Camera-Assistive Surveillance and

Analysis,» de *5th International Conference on Innovative Trends in Information Technology (ICITIIT)*, Kottayam, 2024.

- [104] S. C. S. K. Prasanjit Dey, «Hybrid CNN-LSTM and IoT-based coal mine hazards monitoring and prediction system,» *Process Safety and Environmental Protection*, pp. 249-263, 2021.
- [105] ,. G. B. a. M. A. I. Abdul Wahid, «Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework,» *applied sciences*, vol. 12, nº 9, 2022.
- [106] M. M. & W. Li, «Cutting tool prognostics enabled by hybrid CNN-LSTM with transfer learning,» *The International Journal of Advanced Manufacturing Technology* , 2021.
- [107] ,. B. ,. e. H. S. J. Z. I. a. J. A. Shahid Latif Maha Driss, «Deep Learning for the Industrial Internet of Things (IIoT): A Comprehensive Survey of Techniques, Implementation Frameworks, Potential Applications, and Future Directions,» *Sensors*, vol. 21, nº 22, 2021.
- [108] K. S. Lee, S. B. Kim y H.-W. Kim, «Enhanced Anomaly Detection in Manufacturing Processes Through Hybrid Deep Learning Techniques,» *IEEE Access*, vol. 11, 2023.
- [109] A. A. A. & N. Ayub, «Deep learning hybridization for improved malware detection in smart Internet of Things,» *scientific reports*, 2024.

- [110] L. C. D. a. L. G. D. M. Eugenio Brusa, «Explainable AI for Machine Fault Diagnosis: Understanding Features' Contribution in Machine Learning Models for Industrial Condition Monitoring,» *applied sciences*, 2023.
- [111] K. G. E. M. H. Marcia L. Baptista, «Relation between prognostics predictor evaluation metrics and local interpretability SHAP values,» *Artificial Intelligence*, vol. 306, 2022.
- [112] A. Assadzadeh, «github.com,» [En línea]. Available: <https://github.com/amin-asdzh/ExcPose2D>. [Último acceso: 20 2 2023].
- [113] A. Assadzadeh, M. Arashpour, I. Brilakis, T. Ngo y E. Konstantinou, «Vision-based excavator pose estimation using synthetically generated datasets with domain randomization,» *Automation in Construction*, vol. 134, 2022.
- [114] S. H. Bilawal Mahmood, «Implementation experiments on convolutional neural network training using synthetic images for 3D pose estimation of an excavator on real images,» *Automation in Construction*, vol. 133, 2022.
- [115] M. M. Soltani, Z. Zhu y A. Hammad, «Skeleton estimation of excavator by detecting its parts,» *Automation in Construction*, vol. 82, 2017.
- [116] L. C. D. a. L. G. D. M. Eugenio Brusa, «Explainable AI for Machine Fault Diagnosis: Understanding Features' Contribution in Machine Learning Models for Industrial Condition Monitoring,» *applied sciences*, vol. 13, nº 4, 2023.

