



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado

Facultad de Ingeniería de Sistemas e Informática

Unidad de Posgrado

**Modelo de detección inteligente de ataques cibernéticos
en sistemas de distribución de agua SCADA usando
Machine Learning**

TESIS

Para optar el Grado Académico de Magíster en Ingeniería de
Sistemas e Informática con mención en Ingeniería de Software

AUTOR

Amanda Liliana GALARZA YALLICO

ASESOR

Dra. Fany Yexenia SOBERO RODRÍGUEZ

Lima, Perú

2026



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Galarza, A. (2026). *Modelo de detección inteligente de ataques cibernéticos en sistemas de distribución de agua SCADA usando Machine Learning*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Amanda Liliana Galarza Yallico
Tipo de documento de identidad	DNI
Número de documento de identidad	09742737
URL de ORCID	https://orcid.org/0000-0002-3311-5424
Datos de asesor	
Nombres y apellidos	Fany Yexenia Sobero Rodríguez
Tipo de documento de identidad	DNI
Número de documento de identidad	20120467
URL de ORCID	https://orcid.org/0000-0002-0323-6110
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Marco Antonio Coral Ygnacio
Tipo de documento de identidad	DNI
Número de documento de identidad	09983907
Miembro del jurado 1	
Nombres y apellidos	William Martín Enriquez Maguiña
Tipo de documento de identidad	DNI
Número de documento de identidad	06179457
Miembro del jurado 2	
Nombres y apellidos	Herminio Paucar Curasma
Tipo de documento de identidad	DNI
Número de documento de identidad	32920084
Datos de investigación	
Línea de investigación	C.0.5.8. Inteligencia artificial

Grupo de investigación	No aplica.
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Latitud: -12.055926 Longitud: -77.083469
Año o rango de años en que se realizó la investigación	2023 - 2024
URL de disciplinas OCDE	Sistemas de automatización, Sistemas de control https://purl.org/pe-repo/ocde/ford#2.02.03



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, Decana de América
Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

**ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE
MAGÍSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON MENCIÓN EN
INGENIERÍA DE SOFTWARE.**

Al primer (1) día del mes de julio de 2026, siendo las 12:00 pm., se reunieron en el Auditorio Alfredo Celso Alva Bravo, el Jurado de Tesis conformado por los siguientes docentes:

Mg. Marco Antonio Coral Ygnacio (Presidente)
Mg. William Martín Enriquez Maguiña (Miembro)
Mg. Herminio Paucar Curasma (Miembro)
Dra. Fany Yexenia Sobero Rodríguez (Miembro Asesor)

Se inició la Sustentación invitando a la candidata a Magíster **AMANDA LILIANA GALARZA YALLICO**, para que realice la exposición oral de la tesis para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software, siendo la Tesis intitulada:

“MODELO DE DETECCIÓN INTELIGENTE DE ATAQUES CIBERNÉTICOS EN SISTEMAS DE DISTRIBUCIÓN DE AGUA SCADA USANDO MACHINE LEARNING”

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

..... Presisais (16) bueno

Por tanto, el presidente del Jurado, de acuerdo con el Reglamento General de Estudios de Posgrado, otorga a la Bachiller **AMANDA LILIANA GALARZA YALLICO** el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software,

Siendo las 13:00 horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.

Mg. Marco Antonio Coral Ygnacio
(Presidente)

Mg. William Martín Enriquez Maguiña
(Miembro)

Mg. Herminio Paucar Curasma
(Miembro)

Dra. Fany Yexenia Sobero Rodríguez
(Miembro Asesor)



Anexo 1

CERTIFICADO DE SIMILITUD

Yo, Fany Yexenia Sobero Rodríguez en mi condición de asesor acreditado mediante Dictamen de la UPG N° 000154-2024-UPG-VDIP-FISI/UNMSM de la tesis, cuyo título es, **“Modelo de detección inteligente de ataques cibernéticos en sistemas de distribución de agua SCADA usando Machine Learning”**, presentado por la egresada **Amanda Liliana Galarza Yallico**, para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software,

CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 10% de similitud, nivel PERMITIDO para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software.

Lima, 06 de julio de 2026

Firma del Asesor

Nombres y Apellidos del asesor: Fany Yexenia Sobero Rodríguez

DNI: 20120467

* Según corresponda: bachiller, magíster, doctor; título profesional o título de segunda especialidad

La vigencia del presente certificado corresponde a un período de noventa (90) días desde la fecha de su emisión; siempre que no haya observaciones en el texto durante el proceso de revisión por el jurado informante o surja alguna contingencia.

DEDICATORIA

Esta tesis está dedicada a los dos grandes amores de mi vida, a mi hija Urpi, que me enseñó que nunca es tarde para seguir aprendiendo, a mi hijo Atik por motivarme con sus palabras de aliento para perseverar y lograr mi objetivo propuesto, a mi esposo por su paciencia y consejo. A mis padres y hermanos, esto también va para ustedes.

AGRADECIMIENTO

A Dios gracias por cuidar de mí e iluminar mi camino en la vida.

A la profesora Dra. Fany Sobero Rodríguez, gracias por su apoyo y guía, por dedicar su tiempo en la asesoría y revisión del presente trabajo para culminarlo con satisfacción.

A todos los profesores de maestría en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software de la UNMSM, gracias por compartir sus enseñanzas valiosas, ser mentores, y guías para seguir en el camino hacia el conocimiento y el desarrollo personal.

Por último, gracias a mi familia por la paciencia, por el tiempo quitado, por su comprensión, pero sobre todo gracias por su amor incondicional y apoyo emocional.

ÍNDICE

ÍNDICE.....	IV
LISTA DE CUADROS	VII
LISTA DE FIGURAS	VIII
RESUMEN	X
ABSTRACT	XI
CAPÍTULO 1: INTRODUCCIÓN	12
1.1. Situación problemática	12
1.2. Formulación del problema	16
1.2.1. Problema general.....	16
1.2.2. Problemas específicos.....	16
1.3. Justificación teórica	17
1.4. Justificación práctica.....	18
1.5. Objetivos.....	19
1.5.1. Objetivo general.....	19
1.5.2. Objetivos específicos	19
1.6. Organización de la tesis	20
CAPÍTULO 2: MARCO TEÓRICO	22
2.1. Antecedentes del Problema.....	22
2.2. Bases Teóricas.....	31
2.2.1. Sistemas SCADA.....	31
2.2.1.1 Características de un sistema SCADA	32
2.2.1.2 Componentes de un sistema SCADA.....	33
2.2.1.3 Red de Comunicación en un sistema SCADA.....	35
2.2.1.4 Instrumentos de Campo	37
2.2.1.5 La pirámide de control de un sistema SCADA.....	37
2.2.2. Machine Learning	38

	V
2.2.2.1 Clasificación de Machine Learning	39
2.2.2.2 Algoritmos de Machine Learning	41
2.2.3. Ataques ciberfísicos.....	43
2.2.3.1 Clasificación de Ataques ciberfísicos	44
2.2.3.2 Escenarios de ataques ciberfísicos	48
2.2.4. Conjuntos de datos	49
2.2.5. Métricas para evaluación del modelo.....	51
CAPÍTULO 3: METODOLOGÍA	54
3.1 Identificación de variables	54
3.1.1 Variable independiente	54
3.1.2 Variable dependiente.....	54
3.2 Operacionalización de las variables	55
3.3 Matriz de Consistencia	56
3.4 Tipo y Diseño de investigación	57
3.5 Unidad de análisis	57
3.6 Población de Estudio	57
3.7 Tamaño de muestra.....	57
3.8 Selección de muestra	58
3.9 Técnicas de recolección de datos.....	58
3.10Análisis e interpretación de la información	59
3.11Metodología de implementación	59
CAPÍTULO 4: IMPLEMENTACIÓN	61
4.1 Propuesta Técnica para la integración	61
4.2 Metodología de Implementación del Modelo	68
4.3 Descripción del Conjunto de Datos (Dataset).....	70
4.4 Preprocesamiento de los datos	79
4.5 Filtrado y Extracción de características	82

	VI
4.6 Diseño del Modelo de detección.....	87
CAPÍTULO 5: RESULTADOS Y DISCUSIÓN.....	100
5.1. Presentación de resultados	100
5.1.1. Métricas de rendimiento del modelo	101
5.1.2. Tiempo de detección de los ataques	105
5.2. Discusión	107
5.2.1. Métrica <i>sTTD</i>	107
5.2.2. Exactitud del modelo.....	108
5.2.3. Puntuación F1.....	109
5.2.4. Precisión del modelo.....	110
5.2.5. Exhaustividad del modelo	111
CONCLUSIONES	113
RECOMENDACIONES	115
REFERENCIAS BIBLIOGRÁFICAS.....	117

LISTA DE CUADROS

<i>Cuadro 1.</i>	Datasets más usados por los investigadores	50
<i>Cuadro 2.</i>	Operacionalización de las variables	55
<i>Cuadro 3.</i>	Matriz de Consistencia	56
<i>Cuadro 4.</i>	Fases de las metodologías de artículos del estado del arte	68
<i>Cuadro 5.</i>	Variables de los conjuntos de datos C-town.....	75
<i>Cuadro 6.</i>	Ataques del conjunto de datos BATADAL_dataset04	76
<i>Cuadro 7.</i>	Ataques del conjunto de datos BATADAL_test_dataset.....	78
<i>Cuadro 8.</i>	Selección de Hiperparámetros	94
<i>Cuadro 9.</i>	Estructura del algoritmo.....	96
<i>Cuadro 10.</i>	Parámetros del modelo.....	98
<i>Cuadro 11.</i>	Comparación del modelo propuesto con el modelo AE de referencia	104
<i>Cuadro 12.</i>	Valores para el cálculo de la métrica sTTD	107
<i>Cuadro 13.</i>	Comparación de la métrica sTTD con otros estudios.....	108
<i>Cuadro 14.</i>	Comparación de la métrica Accuracy con otros estudios	109
<i>Cuadro 15.</i>	Comparación de Puntuación F1 con otros estudios	110
<i>Cuadro 16.</i>	Comparación métrica Precisión con otros estudios	111
<i>Cuadro 17.</i>	Comparación métrica exhaustividad con otros estudios	112

LISTA DE FIGURAS

<i>Figura 1.</i>	Causas y efectos del problema.	14
<i>Figura 2.</i>	Tendencia en las vulnerabilidades de ICS/OT reportadas.	15
<i>Figura 3.</i>	Componentes sistema SCADA de distribución de agua.....	33
<i>Figura 4.</i>	Componentes de comunicación sistema SCADA de agua.....	37
<i>Figura 5.</i>	Pirámide de Control-Comunicaciones.	38
<i>Figura 6.</i>	Pirámide de Control-Niveles.....	38
<i>Figura 7.</i>	Clasificación de Machine Learning.	39
<i>Figura 8.</i>	Máquina de Vectores de Soporte.	43
<i>Figura 9.</i>	Tipos de ataques.	45
<i>Figura 10.</i>	Matriz de confusión.	52
<i>Figura 11.</i>	Arquitectura de software para integración.	61
<i>Figura 12.</i>	Metodología para la detección de ataques.....	70
<i>Figura 13.</i>	Red de distribución de agua C-town.....	71
<i>Figura 14.</i>	Conjunto de datos denominado (BATADAL_dataset03).	72
<i>Figura 15.</i>	Conjunto de datos denominado (BATADAL_dataset04).	73
<i>Figura 16.</i>	Conjunto de datos denominado (BATADAL_test_dataset).....	74
<i>Figura 17.</i>	Correlación en BATADAL_dataset03.	81
<i>Figura 18.</i>	Selección de atributos del conjunto de datos de validación.	83
<i>Figura 19.</i>	Correlación de la variable Presión en nudos P_J289.	84
<i>Figura 20.</i>	Correlación de la variable de Caudal F_PU6.....	85
<i>Figura 21.</i>	Correlación de la variable Presión en nudo P_J256.....	85
<i>Figura 22.</i>	Ranking de atributos del análisis PCA.....	86
<i>Figura 23.</i>	Arquitectura de Autoencoder.	89
<i>Figura 24.</i>	Definición de Parámetros.	98

<i>Figura 25.</i>	Entrenamiento del modelo con early stopping.....	99
<i>Figura 26.</i>	Crear Autoencoder para detección.....	99
<i>Figura 27.</i>	Resultado de la creación de autoencoder.	99
<i>Figura 28.</i>	Detección en el dataset test de validación.	100
<i>Figura 29.</i>	Detección en el dataset test de prueba.	101
<i>Figura 30.</i>	Rendimiento de la métrica Accuracy por estudio.....	109

RESUMEN

Las infraestructuras críticas que son esenciales para nuestra vida en la actualidad, tanto las redes de distribución de agua como las redes eléctricas, están controladas por sistemas de control, supervisión y adquisición de datos (SCADA). Garantizar la seguridad de estos sistemas frente a los ataques cibernéticos y amenazas maliciosas es una preocupación importante en los últimos tiempos porque los efectos negativos de estos ataques no solo pueden interrumpir o desactivar por completo el funcionamiento del sistema, sino que también pueden generar graves consecuencias relacionadas con la seguridad.

En el presente trabajo de tesis se desarrolla un modelo de aprendizaje no supervisado utilizando codificadores automáticos (AE) para la detección de ataques ciberfísicos por su capacidad de aprender características de los datos y actuar como un método de reducción de dimensionalidad. Para mejorar el rendimiento de la detección de AE se aplicaron cuatro restricciones de aseguramiento, pesos vinculados, pesos ortogonales, características no correlacionadas y norma unitaria.

Para la validación de la propuesta se utilizó el conjunto de datos C-Town y para probar las hipótesis planteadas se hizo uso de las métricas más utilizadas en los estudios del estado del arte de este trabajo de estudio. Los resultados obtenidos del modelo propuesto son prometedores con una puntuación de tiempo hasta la detección de un ataque STTD=0.947, superior a la del modelo AE base y un Accuracy=96% que supera a los otros estudios de investigación encontrados en la literatura.

Palabras Clave: SCADA, aprendizaje automático, codificadores automáticos, ataques ciberfísicos, infraestructuras críticas.

ABSTRACT

The critical infrastructures that are essential to our lives today, water distribution networks and electricity grids are controlled by control, monitoring, and data acquisition (SCADA) systems. Ensuring the security of these systems against cyberattacks and malicious threats is a major concern in recent times because the negative effects of these attacks can not only disrupt or completely disable the system's operation but can also lead to serious security-related consequences.

In the present thesis work, an unsupervised learning model is developed using automatic encoders (AE) to detect cyber-physical attacks due to their ability to learn data characteristics and act as a dimensionality reduction method. To improve the performance of AE detection, four assurance constraints were applied: linked weights, orthogonal weights, uncorrelated characteristics, and unit norm.

For the validation of the proposal, the C-Town dataset was used and to test the hypotheses raised, the most used metrics in the studies of the state of the art of this study work were used.

The results obtained from the proposed model are promising with a time to detection score of an attack STTD=0.947 higher than that of the base AE model and an Accuracy=96% that exceeds the other research studies found in the literature.

Keywords: SCADA, machine learning, automatic encoders, cyber-physical attacks, critical infrastructures.

CAPÍTULO 1: INTRODUCCIÓN

1.1. Situación problemática

La Cátedra de Seguridad Hídrica Sostenible de la UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) perteneciente al Instituto de Ambiente de la Universidad Internacional de la Florida ha estado trabajando en los últimos años para llamar la atención sobre la importancia de incluir la protección del recurso en todos los planes para la seguridad hídrica actuales o que se están desarrollando. En el proceso de alcanzar la seguridad hídrica, es esencial considerar que los sistemas hídricos están sujetos a diversos obstáculos. Estos desafíos son de naturaleza institucional, técnica, política, financiera y de información. Para lograr los Objetivos de Desarrollo Sostenible (ODS) establecidos en la Agenda 2030 de las Naciones Unidas, es fundamental abordar estos desafíos, especialmente el Objetivo No. 6, que se enfoca en el agua y el saneamiento (Donoso, 2022).

En América Latina, en abril y mayo del 2022, las instituciones gubernamentales de Costa Rica y Perú fueron atacadas por un grupo de ciberdelincuentes rusos llamado Conti. Este mismo grupo atacó a siete instituciones públicas en Costa Rica, causando pérdidas en importaciones y exportaciones por un valor de 10 millones de dólares y obligando al presidente a declarar estado de emergencia nacional (Cancino, 2022).

Según el informe del Centro Nacional de Seguridad Digital (CNSD) de Perú en su informe alerta integrada de seguridad digital N° 128, página 4 de 8 (2022), menciona que Conti reveló en la Dark Web (Internet Oscura) que hackeó la Dirección General de Inteligencia del Ministerio del Interior

(DIGIMIN), la banda de ciberdelincuentes presumió desde la Deep Web (Internet Oscura) ser una amenaza para los servicios críticos de Perú, incluido el servicio de agua y de electricidad (Centro Nacional de Seguridad Digital, 2022).

SCADA, por sus siglas en inglés, se refiere a los sistemas de supervisión, control y adquisición de datos que se han convertido en el estándar industrial para el control y la gestión de infraestructuras críticas. Los informes indican que el sistema de distribución de agua se encuentra entre los cuatro primeros sectores más afectados por ciberataques en los Sistemas de Control Industrial (ICSs Industrial Control Systems) , y en los últimos años el número de ataques cibernéticos se ha incrementado, se encontró que los atacantes se infiltraron en el sistema a través de diferentes componentes del sistema de Distribución de agua (WDS) inteligente, como enrutadores, PLCs (Controladores Lógicos Programables), servidores de correo y sistemas SCADA. La manipulación de datos, el robo de datos, el robo de recursos computacionales, el robo de agua y la contaminación ambiental fueron algunas de las formas en que se produjeron los ataques (Housh et al., 2022).

Los sistemas de supervisión, control y adquisición de datos (SCADA) y los sistemas de control distribuido (DCS) son elementos esenciales de la infraestructura crítica (CI). Estos sistemas ya no se encuentran aislados, y su vinculación con las tecnologías de la cuarta revolución industrial ha expandido la zona de ataque. Los ciberataques se han vuelto más avanzados y los criminales tienen la habilidad de sobrepasar los sistemas de seguridad tradicionales; por ende, la identificación de ataques se ha transformado en un campo complicado. Las tecnologías de defensa han integrado métodos de aprendizaje automático (ML) que tienen la capacidad de enfrentar tipos de amenazas más extensos.(Pinto et al., 2023).

La información de los sistemas SCADA de agua puede resultar complicada de gestionar debido a su complejidad, variabilidad, irregularidades y susceptibilidad. Los datos comprenden diversas mediciones y pueden fluctuar a lo largo del tiempo debido a alteraciones en las condiciones del entorno y modificaciones en las operaciones. Los patrones inusuales y las mínimas variaciones pueden influir de manera considerable en el análisis y la toma de decisiones. (Mboweni et al., 2023).

La automatización tiene un rol crucial en los sistemas de producción industrial. Para agilizar los procesos industriales, la mayoría de las industrias llevan a cabo operaciones a distancia mediante sistemas de Supervisión, Control y Adquisición de Datos (SCADA). Es más crucial garantizar una red SCADA que garantizar una red de TI (Tecnologías de la Información). La mayoría de las soluciones de seguridad tecnológica se fundamentan en premisas comunes de seguridad, las cuales frecuentemente no son efectivas para los sistemas SCADA. (Upadhyay & Sampalli, 2020).

Para un mejor entendimiento de la situación problemática, se muestra la figura 1, que permitirá visualizar de manera esquematizada las posibles causas y efectos del problema identificado.

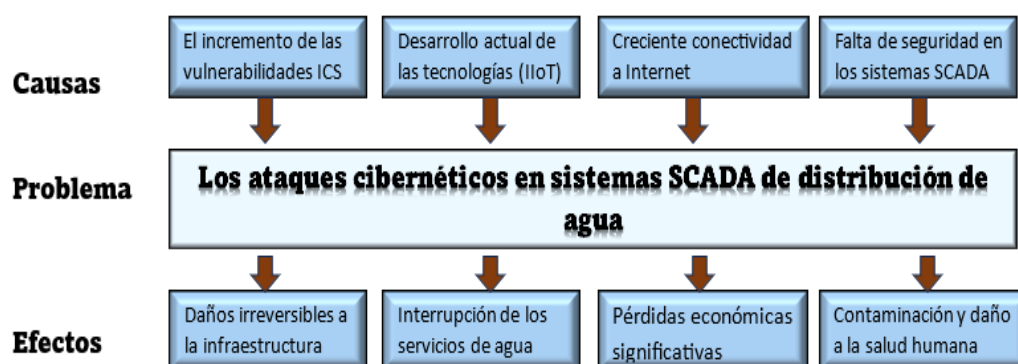


Figura 1. Causas y efectos del problema.

En la figura 2 se muestra el número de vulnerabilidades que afectaron a ICS y tecnología operativa (OT), un total de 2170 CVE (Common Vulnerabilities and Exposures), (sistema de catalogación pública que identifica y enumera las vulnerabilidades de seguridad conocidas) individuales en 2022, lo que representa un aumento del 27% en comparación con 1703 en 2021.



Figura 2. Tendencia en las vulnerabilidades de ICS/OT reportadas.

Fuente. Datos tomados de DRAGOS (2022).

Es fundamental utilizar tecnologías inteligentes como el aprendizaje automático para proteger estos sistemas, ya que los ataques producidos en la capa SCADA son los más complejos y difíciles de detectar, pero además tienen las consecuencias más devastadoras (D. T. Ramotsoela et al., 2019).

Por lo tanto, el incremento de vulnerabilidades en ICS, el desarrollo actual de las tecnologías, la creciente conectividad a internet y la falta de seguridad en los sistemas SCADA nos conducen al problema de ataques cibernéticos en el sistema SCADA de distribución de agua. Un sistema SCADA pertenece al nivel de control, el cual consiste en componentes de control distribuido. Este nivel involucra la sala de control para propósitos de

monitoreo y control. Es posible que los componentes de este nivel no se actualicen periódicamente; en consecuencia, se aplican pocos mecanismos de ciberseguridad en este nivel. A diferencia del nivel corporativo, las necesidades de seguridad son particulares y distintas. Los ataques cibernéticos en cada nivel también varían. También se emplean técnicas de aprendizaje automático para investigar los ataques cibernéticos de estos sistemas. Se tiene la necesidad de atender esa situación y por ello se plantea el presente trabajo.

1.2. Formulación del problema

1.2.1. Problema general

¿De qué manera el diseño de un modelo de detección inteligente basado en algoritmos y técnicas de Machine Learning, permite detectar tempranamente ataques cibernéticos en sistemas SCADA de distribución de agua, con un desempeño óptimo en comparación con modelos autoencoder o no supervisados?

1.2.2. Problemas específicos

- **PE1:** ¿De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el menor tiempo de detección posible en comparación con estudios basados en modelos autoencoder o modelos no supervisados?
- **PE2:** ¿De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mayor exactitud en comparación con estudios basados en modelos autoencoder o modelos no supervisados?

- **PE3:** ¿ De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mejor puntuación F1, es decir, con mejores predicciones de detección de ataques en comparación con estudios basados en modelos autoencoder o modelos no supervisados?
- **PE4:** ¿De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor precisión en comparación con estudios basados en modelos autoencoder o modelos no supervisados?
- **PE5:** ¿ De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor exhaustividad en comparación con estudios basados en modelos autoencoder o modelos no supervisados?

1.3. Justificación teórica

Los ciberataques tienen como objetivo obtener acceso a la información de infraestructuras importantes como los sistemas SCADA de distribución de agua potable mediante el ataque a dispositivos o redes. En los últimos diez años, han aumentado en número e importancia debido a sus efectos devastadores, que pueden resultar en pérdidas multimillonarias para las partes afectadas.

Según Nikolopoulos & Makropoulos (2022), el problema es muy importante para los autores debido a las graves implicaciones de tales amenazas, que incluyen impactos sociopolíticos y de salud. Por ejemplo, un ataque en la planta de tratamiento de Milwaukee Water Works en Estados Unidos

provocó un brote de *Cryptosporidium* en 1993 que afectó a 403 mil personas, resultó en aproximadamente 50 a 70 muertes con un daño económico estimado de \$96.2 millones. Otro ejemplo es el derrame químico de 2014 en West Virginia, también en Estados Unidos, en donde aproximadamente 40 mil litros del compuesto tóxico metilciclohexanometanol (MCHM) y éter se filtraron de un complejo industrial al río Elk. Los atacantes ingresaron a la planta de tratamiento de agua y contaminaron la principal fuente de agua de West Virginia, American Water, en Charleston, dejando a 300 mil personas sin agua potable.

La presente tesis se realiza con el propósito de aportar al conocimiento existente sobre algoritmos de aprendizaje automático para la detección de ataques cibernéticos en sistemas de distribución de agua SCADA, reducir los impactos negativos de los ataques, realizar la caracterización de los ataques, seleccionar los modelos y técnicas adecuadas, seleccionar la data histórica y un software de simulación SCADA, diseñar un algoritmo para detección de ciberataques para aumentar la eficacia en comparación con otros estudios similares.

1.4. Justificación práctica

La industria relacionada con los recursos del agua no es ajena a la tendencia global que se ha visto en los últimos años del incremento significativo de los ciberataques de diversa índole. Es importante tener en cuenta que los ataques cibernéticos en esta industria pueden impactar, además de las bases de datos o bancos de información, sus infraestructuras hidráulicas y sus sistemas de alerta temprana. Por lo tanto, las instituciones del sector del agua pueden ser extremadamente vulnerables a ataques cibernéticos si no tienen las políticas o sistemas de seguridad cibernética adecuados (Donoso, 2022).

Por lo descrito anteriormente, es importante el diseño de un modelo de aprendizaje automático que pueda no solo minimizar los costos significativos que acarrearán estos ciberataques, sino que pueda contrarrestar un efecto desastroso en la seguridad, la infraestructura e incluso la salud de la población.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar un modelo de detección inteligente, basado en algoritmos y técnicas de Machine Learning, que permita detectar tempranamente ataques cibernéticos en sistemas SCADA de distribución de agua, con un desempeño óptimo en comparación con modelos autoencoder o no supervisados.

1.5.2. Objetivos específicos

Para lograr el objetivo general, es necesario alcanzar los siguientes objetivos específicos:

- **OE1:** Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el menor tiempo de detección posible en comparación con estudios basados en modelos autoencoder o modelos no supervisados.
- **OE2:** Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mayor exactitud en comparación con estudios basados en modelos autoencoder o modelos no supervisados.
- **OE3:** Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mejor puntuación F1, es decir, con mejores predicciones de detección de

ataques en comparación con estudios basados en modelos autoencoder o modelos no supervisados.

- **OE4:** Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor precisión en comparación con estudios basados en modelos autoencoder o modelos no supervisados.
- **OE5:** Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor exhaustividad en comparación con estudios basados en modelos autoencoder o modelos no supervisados.

1.6. Organización de la tesis

Este trabajo está organizado de la siguiente manera:

El capítulo 1 pertenece a la introducción; en ella se explica la situación problemática del tema de investigación, se formula el problema general, la justificación teórica y práctica; al final, se formulan los objetivos de la investigación.

El capítulo 2 expone el marco teórico. Se inicia con los antecedentes de la investigación, que nos ayudan a contextualizar el problema de investigación, y también con las bases teóricas que han permitido fundamentar nuestro estudio. En él se exponen teorías y conceptos relevantes que están relacionados con la detección de ataques cibernéticos.

En el capítulo 3 se exhibe la metodología que ha sido utilizada en la investigación, donde se identifican las variables, se presenta la matriz de consistencia, se expone el tipo y diseño de la investigación; en la última sección se presenta la metodología de implementación del modelo a

desarrollar.

En el capítulo 4 se muestra la implementación del modelo propuesto. Se presenta una metodología denominada MADICS, útil para la detección de anomalías de aprendizaje automático; además, está respaldada por las fases que han sido utilizadas en los diferentes estudios expuestos en el capítulo 2.

El capítulo 5, presenta los resultados y la discusión para validar nuestras hipótesis de investigación que se encuentran en el capítulo tres dentro de la matriz de consistencia.

Finalmente, se presentan las conclusiones y recomendaciones que se han realizado en el estudio.

CAPÍTULO 2: MARCO TEÓRICO

2.1. Antecedentes del Problema

El artículo de los autores Taormina et al. (2018) tuvo como objetivo mostrar los resultados de la competencia Battle of the Attack Detection Algorithms (BATADAL) el cual es un evento importante en el campo de la detección de ataques ciberfísicos en sistemas de distribución de agua (WDS). Para ello se compara el rendimiento de los algoritmos en la detección de ataques ciberfísicos, cuya frecuencia ha aumentado en los últimos años junto con la adopción de tecnologías inteligentes de agua. Los participantes fueron desafiados a desarrollar un algoritmo que pueda detectar ataques después de recibir conjuntos de datos con observaciones SCADA (simuladas). Los tiempos de detección y la precisión de clasificación se utilizaron para evaluar la eficacia de cada algoritmo presentado. La batalla se llevó a cabo con siete equipos y presentaron una diversidad de estrategias exitosas que hacen uso de análisis de datos, verificación de reglas y mecanismos de detección basados en modelos. Como resultado, el ganador de la competencia fue el grupo liderado por Housh y Ohar con una puntuación $S=0.970$ y un mejor tiempo de detección de $STTD=0.965$; además, dichos resultados se presentaron en el Simposio de Análisis de Sistemas de Distribución de Agua (Congreso Mundial de Medio Ambiente y Recursos Hídricos) que tuvo lugar en Sacramento, California, en mayo de 2017.

Housh et al. (2022) propusieron un artículo de investigación que tuvo como objetivo un modelo semisupervisado que utilizó una reducción de la dimensionalidad mediante el análisis de correlación canónica máxima (MCCA) seguido de la descripción de vectores de soporte (SVDD). A

diferencia de estudios anteriores, los autores mostraron una configuración del proceso de clasificación, que produce un alto rendimiento en diferentes redes y conjuntos de datos. Además, este estudio desarrollo un método de detección para considerar el problema de localización de los ataques. El algoritmo fue desarrollado y probado en dos casos de estudio y varios conjuntos de datos, demostrando un alto rendimiento constante en la detección y localización de ataques cibernéticos. Como resultado se obtuvo en la métrica recall (sensibilidad) = 0.946 y un tiempo de detección de STTD=0.956 con el conjunto de datos C-Town.

Kadosh et al. (2020) este estudio de investigación tuvo como objetivo el desarrollo de un método de descripción de datos de vector de soporte (SVDD) junto con una metodología de selección de características personalizada, que se basa en la comprensión física de la topología WDS. En el método propuesto alternativo al sistema de detección basado en modelos, utiliza un algoritmo basado en datos. El algoritmo desarrollado clasifica automáticamente el comportamiento normal y anormal del sistema basándose únicamente en las lecturas de los sensores. Por lo tanto, combina las ventajas de incluir las reglas de gobierno físico del WDS y supera la necesidad de un modelo físico bien calibrado. Como resultado se obtuvo que el algoritmo desarrollado alcanzó una puntuación $S=0.954$. Esta puntuación es comparable con los estudios que utilizaron el conjunto de datos de entrenamiento completamente etiquetado de otros estudios en detección de ciberataques.

Mahmoud et al. (2022) tuvo como objetivo plantear una arquitectura de detección de dos etapas que utiliza algoritmos auto supervisados y no supervisados para proporcionar sistemas de alarma y confirmación de ataques ciber físicos, donde cada etapa presenta un método de detección único con un beneficio distintivo, como detección rápida en la etapa 1 y exactitud y precisión en la etapa 2. La etapa 1 tiene como objetivo detectar

ataques casi en tiempo real y proporcionar ataques confiables a través de un sistema de alarma de ataques heurístico y auto supervisado recopilando datos de los dispositivos WDS como la presión del tanque, flujo de la bomba, flujo de la tubería para comprender el patrón de las lecturas promedio de la red mientras que la etapa 2 tiene como objetivo confirmar los ataques de CP utilizando un algoritmo no supervisado de detección precisa que es iForest, en esta etapa se utiliza la misma entrada que en la etapa 1, pero no se basa en los hechos aprendidos previamente, descubre la estructura subyacente entre las mediciones del dispositivo WDS y los patrones de lectura a lo largo del tiempo para clasificar los datos proporcionados. Como resultado, la arquitectura propuesta demostró ser práctica debido a su dependencia únicamente de los datos hidráulicos normales en comparación con otros estudios.

Rodríguez et al. (2023) el artículo tuvo como objetivo proponer una metodología novedosa para la detección y localización de ciberataques en WDS combinando herramientas de inteligencia computacional y análisis estructural, lo que representa la principal contribución de los investigadores. En el contexto de las WDS, se ha prestado poca atención al problema de la detección de ciberataques y a encontrar el lugar o área donde se está produciendo el ataque. La metodología propuesta es aplicable a sistemas del mundo real ya que requiere de: (1) Datos de medición nominales o normales para entrenar a los autocodificadores y ajustar los parámetros de la prueba estadística de promedio móvil ponderado exponencial adaptativo (EWMA) y (2) Una comprensión de la configuración física de las áreas de medición del distrito (DMA) con los sensores y actuadores para definir las ecuaciones que se utilizarán para el análisis estructural, que forman un modelo de comportamiento de la red (no que requieren un modelo físico parametrizado).

El estudio de Almajed et al. (2022) tuvo como objetivo desarrollar un nuevo

marco para la detección de ciberataques que hace uso de inteligencia artificial (IA) y aprendizaje automático (ML). Para la identificación de ciberataques utilizan el algoritmo Heuristic Multi-Swarm Optimization (HMS) junto con el Self-tuned Fuzzy Logic basado en Hidden Markov Model HMM (SFL-HMM). Además, hacen uso de una técnica conocida como Análisis Discriminante Lineal (LDA) para la obtención de características y de MATLAB un simulador que es utilizado para evaluar la nueva estrategia, y las métricas obtenidas de esa simulación se comparan con las obtenidas de los métodos de otros estudios. Como resultado, se obtuvo un rendimiento en la métrica Accuracy $\geq 75\%$.

Choi et al. (2020) propusieron un artículo de investigación que tuvo como objetivo el desarrollo de un modelo de detección novedoso para la detección de ataques cibernéticos, utiliza datos de detección remota en sistemas de distribución de agua (es decir, sensor de flujo de tubería, sensor de presión nodal, sensor de nivel de agua del tanque y controladores lógicos programables) mediante modelos de aprendizaje automático como la máquina de aprendizaje extremo (ELM), el vecino más cercano (KNN), máquina de vectores de soporte (SVM) y redes neuronales artificiales (ANN) para encontrar el mejor algoritmo de machine learning. Este estudio no solo busca el algoritmo de mejor rendimiento a través de una comparación simple de algoritmos, sino que también considera la versión mejorada del algoritmo básico para mostrar el mejor rendimiento. Como resultado en máquina de aprendizaje extremo mejorado (ELM) se obtuvo score de $S=0.91$ en comparación del modelo básico ELM un $S=0.89$.

Brentan et al. (2021) este artículo de investigación que tuvo como objetivo el desarrollo de una metodología automática de dos pasos para la detección de ataques cibernéticos en WDS, se basa en la teoría de procesamiento de señales y aplica un algoritmo de separación de fuente ciega (BSS) y análisis de componentes independiente (fastICA) a series temporales

hidráulicas. La metodología para la detección de ciberataques propuesta en este documento se basa en dos técnicas separadas. Uno de ellos proviene del campo de procesamiento de señales y aplica un algoritmo de separación de fuente ciega (BASS). Estos componentes se pueden detectar utilizando un método de control estadístico, que se corresponde con la otra técnica de este trabajo en el cual se aplica un algoritmo de detección de puntos de cambio abrupto a las señales separadas para detectar con precisión los tiempos de inicio y finalización de los ataques, lo que ayuda a caracterizar los ataques. La metodología aplicada al conjunto de datos BATADAL dio como resultado la detección de los siete ataques con alta precisión y pocos falsos positivos.

Kim et al. (2022) este artículo tuvo como objetivo el desarrollo de un modelo de red neuronal convolucional de conjunto (CNN) que emplea varias herramientas de detección de ráfagas con diferentes mecanismos de detección. Utiliza seis métodos diferentes de control estadístico de procesos (SPC). Se consideraron tres diseños de configuración de mapa de calor diferentes para identificar el mejor diseño que proporciona una mayor precisión de clasificación de CNN. El enfoque de detección de rotura de tubería CNN de conjunto propuesto se aplicó a una red en Austin, TX y mejoró la probabilidad de detección aproximadamente en un 2 % más que la del mejor método SPC conocido.

Ahmed et al. (2022), en este artículo, tuvieron como objetivo el desarrollo de una técnica práctica de marca de agua para detectar los ataques. Los experimentos se realizaron en un sistema de distribución de agua llamado WADI para explorar cómo abordar los ataques de repetición. Utilizaron la señal de marca de agua. Inicialmente, los experimentos se realizaron en condiciones normales de funcionamiento, es decir, sin lanzar ningún ataque, para obtener un comportamiento de referencia del sistema. Para obtener un estado estacionario del sistema, la planta se deja correr por

algún tiempo (5-15 min). Para un funcionamiento normal de la planta, los tanques de los consumidores fueron vaciados y la demanda de los consumidores fue ajustada a una constante de $0,15 \text{ m}^3/\text{h}$ para los seis consumidores. Como resultado, se obtuvo que para un tamaño de fragmento de datos de 250 se logró una tasa de positivos verdaderos (TPR) = 100%.

En Ramotsoela et al. (2020), en este artículo de investigación, se tuvo como objetivo el desarrollo de un modelo para la detección de valores atípicos (anomalías) cuando el modelo intenta aprender el comportamiento normal del sistema y predecir elementos particulares. Cuando el error entre los valores previstos y reales está más allá de un umbral predeterminado, una instancia de datos se considera anómala. Primero utilizaron una serie de redes neuronales populares para aprender el comportamiento normal del sistema y predecir los niveles de los tanques dentro del conjunto de datos BATADAL. La distancia de Mahalanobis se utiliza entonces como la técnica estadística de la segunda etapa para distinguir entre datos normales y anómalos. Además, propusieron una técnica de conjunto basada en la votación para crear un algoritmo más robusto. En esencia, el objetivo del algoritmo fue mejorar las métricas de rendimiento que evalúan la precisión de la detección y mejorar el rendimiento general aprovechando las fortalezas de cada algoritmo individual. Como resultado, se obtuvo un rendimiento en la métrica STTD=0.946.

Mboweni et al. (2023), en este artículo, tuvieron como objetivo formular técnicas de preprocesamiento ideales para mejorar el rendimiento de los modelos de aprendizaje automático para detectar ataques en infraestructuras críticas de sistemas de agua; además, proporcionaron técnicas de preprocesamiento específicas para el procesamiento de datos en el control industrial para producir un conjunto de datos más informativo. Los datos para el procesamiento de datos fueron de componentes

hidráulicos, como sensores que miden parámetros como la presión, el caudal, la temperatura y la posición, y actuadores como bombas, válvulas y cilindros. Como resultado, se obtuvo una mejora en la métrica de puntuación F1-Score = 97.8%.

Sheng et al. (2021) este artículo tuvo como objetivo el desarrollo de un nuevo modelo ciberfísico para el sistema SCADA. El modelo asocia los ataques de red con el proceso industrial del sistema SCADA a través del modelado de los estados de los dispositivos ICS y el mapeo de ataques a las operaciones en ellos. De esta manera, el modelo no solo puede detectar intrusiones contra la red SCADA, sino también evaluar sus niveles de riesgo frente al proceso industrial. Como resultado, se probaron un total de nueve algoritmos supervisados de clasificación para detectar varios ataques en los datasets, el modelo propuesto obtuvo un índice de puntuación F1-Score = 0.987.

Nikolopoulos & Makropoulos (2022) este artículo tuvo como objetivo el desarrollo de una plataforma llamada RISKNOUGHT para evaluaciones de estrés ciberfísico el cual puede acomodar combinaciones de ataques físicos (sabotaje, destrucción de componentes, etc.) que durante la simulación afectan el estado físico de un componente o un atributo de las redes de abastecimiento o distribución de agua Network Distribution Water (WDN) con ataques cibernéticos de denegación de servicio (DoS), comunicación secuestro, etc.) que durante la simulación afectan el comportamiento del esquema de control implementado. Las responsabilidades que implican el modelado de sucesos de contaminación y esquemas complejos de lógica de control que funcionan como estrategias de mitigación contra la contaminación, como el aislamiento y la purga. Como resultado la evaluación de la plataforma RISKNOUGHT en una red de referencia evidencia sus habilidades y brinda datos a las compañías de agua acerca de ataques ciberfísicos que pueden incluir sucesos de contaminación.

Tsiami y Makropoulos (2021) en su artículo que tuvo como objetivo el desarrollo de un algoritmo de detección de análisis de componentes principales (CPA) basado en predicción que aprovecha las interdependencias inherentes de los datos SCADA a través de redes neuronales convolucionales de gráficos temporales (TGCN) y hace uso de la distancia de Mahalanobis. Este estudio examina la capacidad inherente de un TGCN para incorporar las relaciones entre los nodos de un gráfico, lo que lo convierte en una herramienta valiosa en la detección y localización de CPA. Se utilizaron los conjuntos de datos disponibles públicamente presentados en la competencia Battle of the Attack Detection Algorithms (BATADAL). Como resultado, el algoritmo propuesto mostró un fuerte rendimiento de detección, equivalente a los modelos de mejor rendimiento desarrollados para esa competencia con un score $S=0.931$.

Selim et al. (2021), en este artículo, tuvieron como objetivo el estudio analítico de detección de anomalías, actividades maliciosas y ataques cibernéticos en una infraestructura crítica de agua con infraestructura de Internet de las cosas industrial (IIoT). El estudio utiliza varios algoritmos de aprendizaje automático para clasificar los eventos de anomalía, incluidos varios ataques y fallas de hardware de IIoT. Este sistema propuesto fue evaluado y probado con diferentes experimentos. La evaluación del rendimiento se calculó después de probar el sistema propuesto mediante el uso de un conjunto de datos del mundo real que cubre 15 escenarios de anomalías. Asimismo, se utilizan varios algoritmos de aprendizaje automático, como Regresión logística (LR), Análisis discriminante lineal (LDA), K-Vecino más cercano (KNN), Árbol de regresión y clasificación (CART), Gaussian Naive Bayes (NB) y Support Vector Machine. (SVM). Como resultado, el algoritmo CART brindó los mejores resultados para la clasificación en función de los parámetros de evaluación y métricas como exactitud, precisión, recuperación y puntaje F1. Las exactitudes (accuracy)

más altas alcanzadas fueron: 94%, 91% y 88% para CART, KNN y SVM-LDA, respectivamente.

Kravchik & Shabtai (2022), en este estudio, tuvieron como objetivo el desarrollo de un método de detección de ataques basado en redes neuronales simples y livianas, redes neuronales convolucionales 1D y codificadores automáticos. Aplican estas redes neuronales a los dominios de tiempo y frecuencia de los datos y discuten los pros y los contras de cada enfoque de representación. La metodología de implementación es la siguiente: primero, se normalizan los datos; luego, en la fase de entrenamiento, los datos de entrenamiento normalizados pasan al paso de selección del sensor (característica). Las características seleccionadas serán modeladas por el algoritmo de aprendizaje. El método sugerido se evalúa en tres conjuntos de datos públicos WADI, BATADAL y SWAT, y se logran tasas de detección que coinciden o superan los resultados de detección publicados anteriormente por otros investigadores, al tiempo que demuestra tiempos cortos de entrenamiento y detección. Como resultado se tuvo en el conjunto de datos de WADI, ya que pudieron mejorar los resultados informados anteriormente en otros estudios en un 38%.

Moazeni & Khazaei (2021a), en este artículo, tuvieron como objetivo el desarrollo de un modelo de optimización de dos niveles para ataques cibernéticos que pueden resultar en consecuencias físicas de fallas de múltiples nodos en redes de agua. El modelo inyecta datos falsos específicos en las mediciones de las bombas, lo que provoca una acumulación de presión en los nodos objetivo. La formulación de ciberataques de inyección de datos falsos en el caudal de las bombas resulta en fallas en cascada en sistemas de agua inteligentes. Como resultado, se obtuvo que al inyectar datos falsos en el rango del $\pm 10\%$ de

los valores nominales en las mediciones objetivo, esto mostró que el problema de optimización no lineal de dos niveles propuesto modela con exactitud un escenario realista de ciberataque.

Moazeni & Khazaei (2021b) tienen como objetivo desarrollar un novedoso modelo de ataque de optimización utilizando el algoritmo de estimación de estado por hora del sistema de agua. Se asume que los atacantes sólo tienen acceso a las lecturas de la(s) bomba(s) que alimentan el depósito de agua y al contador de agua que registra la demanda total de la red, siendo el nivel de agua del depósito el objetivo del ataque. Como resultado, la presencia del algoritmo de estimación de estado por hora no permite a los atacantes cambiar fácilmente las mediciones a través de la red de agua sin ser detectados. Dichas mediciones se eliminarán inmediatamente de la estimación de estado y se marcarán como datos erróneos.

2.2. Bases Teóricas

Se definen los siguientes conceptos que complementan la investigación actual: sistemas SCADA, Machine Learning (Aprendizaje Automático), ataques ciberfísicos, conjuntos de datos o datasets y métricas para evaluación del modelo.

2.2.1. Sistemas SCADA

Los sistemas SCADA se conocen en español como Control Supervisión y Adquisición de Datos. Según Rodríguez (2011), la interfaz gráfica que conecta al usuario con el sistema SCADA permite la gestión y control de cualquier sistema local o remoto.

Un sistema SCADA es un conjunto de aplicaciones de software específicamente desarrolladas para funcionar en computadoras de control

de la producción. Permite la accesibilidad a la planta a través de la comunicación digital con actuadores e instrumentos, así como una interfaz gráfica de alto nivel para usuarios con componentes como ratones o cursores, pantallas táctiles, lápices ópticos y otros.

A pesar de que en un principio solo era un programa que permitía la supervisión y adquisición de datos en procesos de control, hay una variedad de productos de hardware y buses recientemente desarrollados para este tipo de sistemas. Además, estos sistemas SCADA se pueden conectar a través de una interfaz de computador personal (PC) a la planta central, que se conecta a la computadora principal de supervisión. El sistema puede comunicarse con los dispositivos de campo, como controladores autónomos, autómatas programables, sistemas de dosificación, etc., para controlar automáticamente los procesos desde la pantalla del computador, que el usuario puede configurar y cambiar fácilmente. Además, proporciona a varios usuarios toda la información generada durante el proceso productivo (Pérez López, Esteban, 2015).

2.2.1.1 Características de un sistema SCADA

Según Gómez et al. (2008), las características principales de un SCADA son las siguientes:

- Recoger, procesar y almacenar datos de manera constante y confiable.
- Representación tanto visual como animada de las variables y seguimiento del proceso mediante alarmas.
- Realizar acciones de control para cambiar el progreso del proceso. Esto puede ocurrir a través de reguladores autónomos fundamentales (como menús, consignas, alarmas, etc.) o de manera directa a través de las salidas que están conectadas al proceso.
- Arquitectura adaptable, abierta y flexible.

- Conectarse a otras aplicaciones y bases de datos, ya sean locales o dispersas en redes de comunicación.
- Observación de la evolución de las variables de control desde un terminal.
- Transferir datos entre PC y dispositivos de campo.
- Gestionar bases de datos con menos tiempo de acceso.
- Mostrar y representar los datos en un gráfico, interfaz de usuario o Interfaz Hombre-Máquina.
- Utiliza los datos obtenidos para el control estadístico, la gestión de la calidad, la producción, la gestión financiera y la gestión administrativa.
- Informar al operador sobre cualquier cambio en la planta, así como sobre eventos como alarmas. Estos cambios están registrados en el sistema para que puedan analizarse más tarde.

2.2.1.2 Componentes de un sistema SCADA

Según Valdivia Miranda et al.(2019), para que una aplicación específica de software industrial como lo es un sistema SCADA pueda tratar y gestionar los datos recopilados, necesita componentes de hardware específicos integrado en su propio sistema, que se describen a continuación.

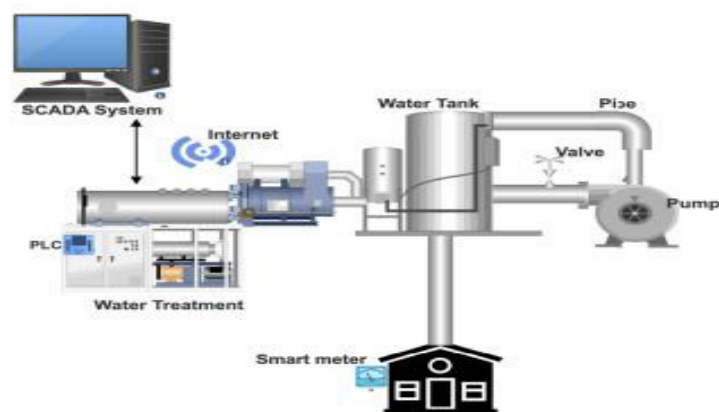


Figura 3. Componentes sistema SCADA de distribución de agua.

Fuente. Datos tomados de Addeen et al. (2024).

La figura 3 muestra los componentes de un sistema SCADA de distribución de agua. Se puede diferenciar este tipo de sistema SCADA de otro tipo de sistema SCADA por la función que cumplen los sensores y actuadores en este tipo de sistemas. Los sensores recopilan datos del proceso, como temperatura, presión, nivel, % de cloro, si una bomba está encendida o apagada y flujo o caudal de agua. En tanto, los actuadores son los que controlan el proceso mediante la activación o desactivación de dispositivos de las diferentes estaciones remotas, como, por ejemplo, una estación remota de bombeo de agua.

A continuación de los componentes principales de un sistema SCADA de distribución de agua.

MTU (Máster Terminal Unit) u Ordenador Central:

La computadora principal del sistema es responsable de supervisar y recopilar datos de las demás subestaciones. En sistemas complejos, las computadoras pueden estar conectadas a los instrumentos de campo o estar directamente sobre estos instrumentos. La mayoría de las veces, esta computadora es una PC que soporta HMI.

Debido a esto, el sistema SCADA más básico consta de una sola computadora, el MTU, que tiene la responsabilidad de supervisar la estación en su totalidad.

Las siguientes son las funciones principales del MTU:

- Preguntar a las RTU regularmente y proporcionar instrucciones, que suelen seguir un modelo de maestro-esclavo.
- Funciona a modo de una interfaz de usuario y permite administrar alarmas, presentar datos de variables actualizados en tiempo real y recopilar y mostrar información "historizada".
- Consigue ejecutar software cualificado que lleva a cabo tareas

específicas afines al proceso supervisado por SCADA.

RTU (Remote Terminal Unit) u Ordenadores Remotos:

Las subestaciones son administradas y controladas por estas computadoras que se encuentran en los nodos estratégicos del sistema. Además, utilizan el software de la aplicación SCADA para controlar los elementos finales de control y recibir señales de los sensores de campo.

Estos equipos pertenecen al nivel de automatización intermedio, con el MTU a un nivel más alto, el superior, y los diversos instrumentos de campo a un nivel inferior que son responsables de la automatización física del sistema, el control y la adquisición de datos.

Estos computadores suelen ser computadoras industriales tipo armarios de control porque no necesitan soportar un HMI de gran tamaño a este nivel, aunque puede haber subestaciones intermedias en el formato HMI en sistemas muy complejos.

Actualmente, debido a un nivel más alto de integración y una unidad central de procesamiento con mayor potencia de cálculo, es común que los controladores lógicos programables (PLCs) funcionen como RTU. La presente solución reemplaza la computadora industrial mucho más costosa y reduce los costos en sistemas con subestaciones simples.

2.2.1.3 Red de Comunicación en un sistema SCADA

Es en este nivel en donde se supervisan los datos que los instrumentos de campo transmiten a la red de computadoras desde el sistema. Dado que algunos software o instrumentos de campo como un PLC consiguen funcionar con todo tipo de BUS, el tipo de BUS que se usa para la

comunicación puede variar significativamente de acuerdo con los requisitos del sistema y el software seleccionado para la implementación del sistema SCADA.

Actualmente, la comunicación con los equipos de campo se ha estandarizado, lo que permite la instalación de un sistema SCADA en casi cualquier tipo de BUS. Las comunicaciones SCADA están disponibles en formatos convencionales, tales como RS-232, RS-422 y RS-485, y se podrá conectar el sistema a un bus DMS (Distancia Mínima de Seguridad) existente utilizando el protocolo TCP/IP. Esto incluye toda clase de buses del campo industrial, así como métodos de comunicación más avanzados como satélite, Bluetooth (radio), cable, microondas y otros. Además, los módems, que soportan los protocolos de comunicación SCADA y facilitan la implementación de aplicaciones, son ejemplos de interfaces de comunicación específicas que se utilizan para comunicarse con un sistema SCADA.

La implementación general en sistemas WAN (Wide Área Network o Red de Área Amplia) es una característica adicional de SCADA en lo que respecta a las comunicaciones, lo que permite la deslocalización geográfica de los distintos terminales RTU.

La figura 4 muestra un ejemplo de los componentes de comunicación de una empresa prestadora de servicios de agua potable que administra un sistema SCADA de distribución de agua.

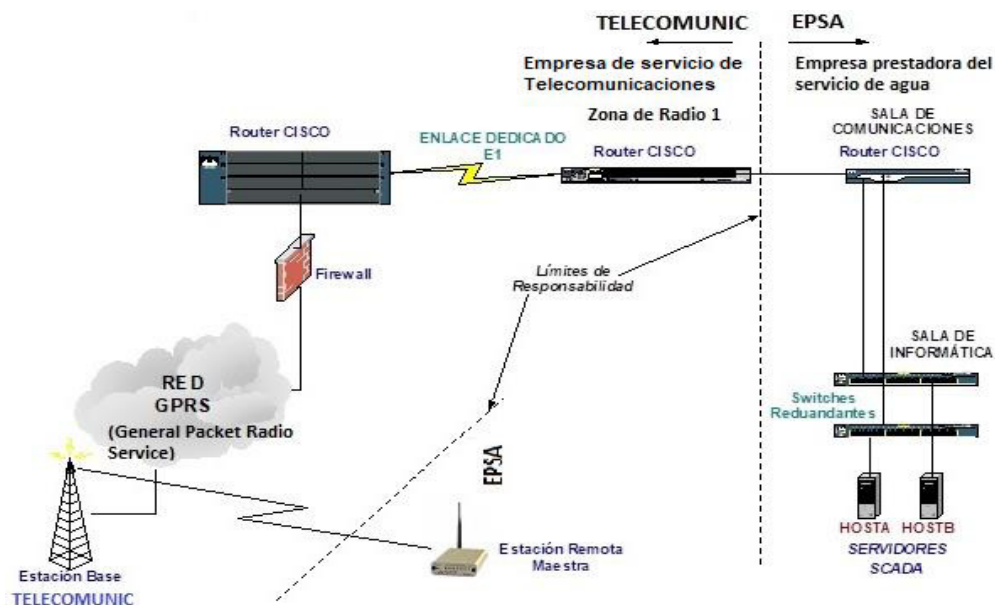


Figura 4. Componentes de comunicación sistema SCADA de agua.

2.2.1.4 Instrumentos de Campo

Son todos aquellos que pueden automatizar o controlar el sistema (PLC, controladores de procesos industriales y actuadores en general) y recopilar datos del sistema (alarmas y sensores).

Los componentes del sistema SCADA son creados por diferentes proveedores sin coordinación entre sí, lo que es una característica. De esta manera, hay una variedad de proveedores para RTU, incluidos radios, módems, minicomputadoras, interfaz de usuario, control y detección de pérdidas, etc.

2.2.1.5 La pirámide de control de un sistema SCADA.

La estructura jerárquica de los sistemas de gestión remota o SCADA se define por la pirámide de control, según el Centro de Ciberseguridad Industrial (CCI). Las figuras 5 y 6 muestran las diversas distribuciones que

la componen, junto con los diversos elementos de control asociados.

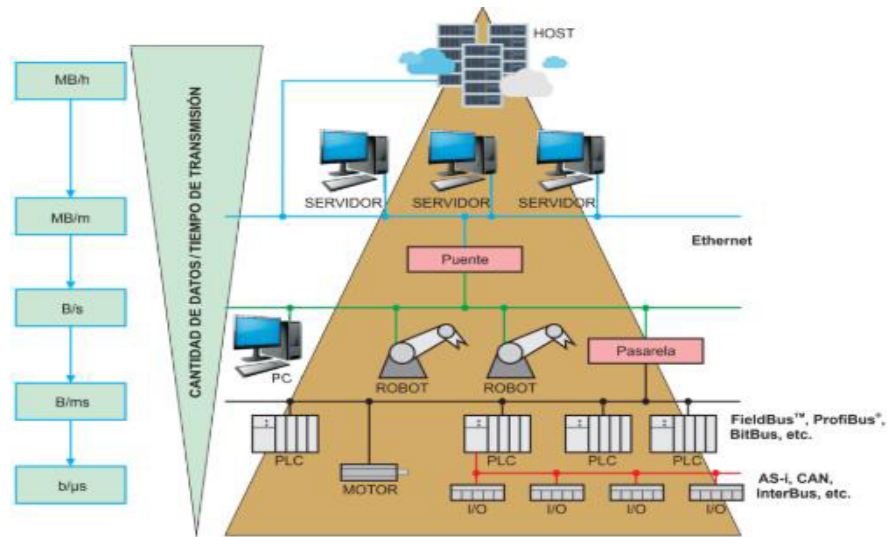


Figura 5. Pirámide de Control-Comunicaciones.

Fuente. Datos tomados de Valdivia Miranda et al. (2019).



Figura 6. Pirámide de Control-Niveles

Fuente. Datos tomados de Sevillano et al.(2021).

2.2.2. Machine Learning

También llamado aprendizaje automático, rama de la IA (Inteligencia Artificial) que analiza cómo usar algoritmos para crear un modelo a partir de un conjunto de datos denominado datos de entrenamiento, que contiene una serie de características o atributos específicos. El modelo tiene la capacidad de hacer predicciones y tomar decisiones una vez que se ha creado. En otras palabras, el aprendizaje automático permite a una computadora aprender cómo realizar tareas anteriores sin haber sido programada para hacerlo. Como resultado, se afirma que el computador "aprende", lo cual significa que se mejorarán los resultados obtenidos teniendo como base la experiencia de resultados anteriores (Zhang, 2020).

2.2.2.1 Clasificación de Machine Learning

Según los datos de entrenamiento utilizados para un algoritmo de aprendizaje automático se podría clasificar de la siguiente forma:



Figura 7. Clasificación de Machine Learning.

Fuente. Datos tomados de Barrios Arce (2019a).

Aprendizaje supervisado:

Se distingue porque el dataset (conjunto de datos) que contiene la clasificación o "etiqueta" de cada uno de los datos es utilizado como datos de entrenamiento. Está basado en la noción de un "profesor" o "supervisor" que calcula el error después de clasificar un dato, de modo que el modelo se ajusta automáticamente para disminuir el error general. Como resultado, con cada repetición del modelo, la precisión con la que puede clasificar los datos se incrementará. La regresión y la clasificación son modos de aprendizaje supervisado. La regresión usa etiquetas continuas o numéricas y la clasificación utiliza etiquetas discretas o categóricas. El aprendizaje supervisado puede usarse para:

- Reconocimiento de voz
- Detectar spam
- Clasificación de imágenes

Aprendizaje no supervisado:

En este tipo de aprendizaje se usa un conjunto de datos sin clasificación (conocido como "datos sin clasificación") para su entrenamiento, y el modelo decide el modo de agrupar o clasificar los datos. Es ventajoso en situaciones en las que se necesita conocer cómo están agrupados el conjunto de elementos en función de su similitud. Hay dos categorías de aprendizaje no supervisado:

- **El agrupamiento o clustering:** tiene como objetivo crear grupos a los que se les asignan los datos de acuerdo con su similitud.
- **La reducción de dimensionalidad** tiene como objetivo hacer que el conjunto de datos nuevo sea tan difícil de manejar como sea posible,

eliminando los aspectos del conjunto de datos que no son tan importantes.

El aprendizaje no supervisado puede utilizarse para:

- Marketing
- Sistemas de recomendación
- Detección de anomalías

Aprendizaje semisupervisado:

Hace uso de un conjunto de datos con una porción de datos clasificados y otra porción de datos no clasificados como datos de entrenamiento.

Aprendizaje por refuerzo:

La retroalimentación sobre las acciones del modelo en un entorno es utilizada como datos de entrenamiento. Debido a que el modelo tiene como objetivo obtener la mayor recompensa como resultado global, esta realimentación está relacionada con la idea de recompensa. El aprendizaje por refuerzo tiene múltiples usos, como:

- Robótica
- Inteligencia artificial de videojuegos

2.2.2.2 Algoritmos de Machine Learning

Según los autores Ramírez Gil (2023) el aprendizaje automático utiliza una amplia gama de algoritmos. No obstante, algunos algoritmos no se pueden usar con el mismo acierto en todo tipo de problemas, aunque en términos de precisión, rapidez de entrenamiento y análisis para proporcionar los resultados deseados algunos funcionan mejor que otros. El tipo de

algoritmo que se utiliza depende del tipo de problema que se quiere resolver, el número de variables involucradas, el tipo de modelo que se adapta mejor a él, etc.

Algoritmos de aprendizaje supervisado:

A continuación, se describen algunos algoritmos de aprendizaje supervisado:

Naïve Bayes Classifier:

El teorema de Bayes es utilizado por este algoritmo para realizar el aprendizaje y la clasificación de manera probabilística, asumiendo que las características de un dataset son independientes (por eso el término "naïve").

Decision Tree Classifier (Árboles de decisión):

Está basado en un árbol de decisión invertido en el que cada uno de los nodos tiene la responsabilidad de tomar una decisión acerca de uno de los valores de los atributos del conjunto, lo que lo convierte en uno de los algoritmos más utilizados en la minería de datos.

Random Forest Classifier (Bosques aleatorios):

Creado por Tin Kam Ho, combina una gran cantidad de árboles de decisión que poseen la misma distribución aleatoria y permiten seleccionar el valor más popular de las clasificaciones, es considerado un algoritmo de mayor confiabilidad de los disponibles al trabajar con conjuntos de datos de gran tamaño además de corregir en los árboles de decisión los efectos del sobreajuste.

Artificial Neural Network (Redes neuronales artificiales):

Se inspiran en la forma de cómo funcionan las neuronas de un cerebro

humano. Millones de neuronas artificiales están conectadas entre sí en sus varias capas ocultas. El peso es un número que muestra la conexión entre un nodo y otro e indica si un nodo estimula a otro o si lo suprime. Los nodos que tienen valores de peso más altos tienen un mayor impacto en los demás nodos.

Support Vector Machine (Máquinas de Vectores de Soporte):

Utiliza hiperplanos de separación; se describen como el vector que conecta dos elementos de diferentes clases, para separar los elementos del conjunto de entrenamiento en el espacio. Cuando se agregan nuevos elementos a la clasificación, se les asigna una clase en función del punto del espacio al que pertenecen respecto al hiperplano de separación seleccionado. Estos modelos se entrenan seleccionando uno de los hiperplanos de separación potenciales en cada iteración, verificando el porcentaje de aciertos y luego eligiendo el que obtenga los mejores resultados.

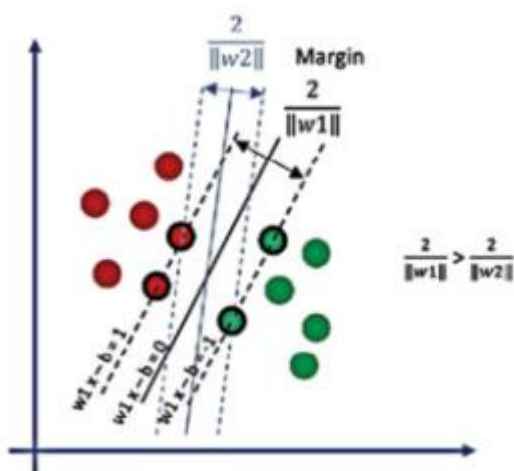


Figura 8. Máquina de Vectores de Soporte.

Fuente. Datos tomados de Ramírez Gil y Ramírez Gil (2023).

2.2.3. Ataques ciberfísicos

2.2.3.1 Clasificación de Ataques ciberfísicos

Los ataques ciberfísicos se pueden clasificar en muchos tipos según los ataques a dispositivos físicos y los ataques a los componentes de comunicación/redes/sistemas de los sistemas de distribución de agua. En Addeen et al. (2021), clasifican los ataques en los siguientes tipos:

Ataques de sensores: Este tipo de ataque ataca sensores, como alterar la lectura de un sensor para datos importantes.

Ataques de actuadores: Este tipo de ataque consiste en atacar actuadores que realizan tareas incorrectas, como vaciar o llenar una bomba con agua suficiente para interrumpir los servicios.

Ataques al sistema de control: Este tipo de incidente afecta el sistema de control, particularmente el sistema SCADA, y puede causar daños a todo el sistema.

Ataques de PLC: Este tipo de ataque tiene que ver con atacar al PLC, como realizar operaciones engañosas del sistema.

Ataques de comunicación / red: Este tipo de ataque ataca los sistemas de comunicación y las redes para interrumpir o dañar el sistema. Las comunicaciones entre sensores, actuadores, PLC, SCADA y otros componentes de hardware y software son algunos ejemplos.

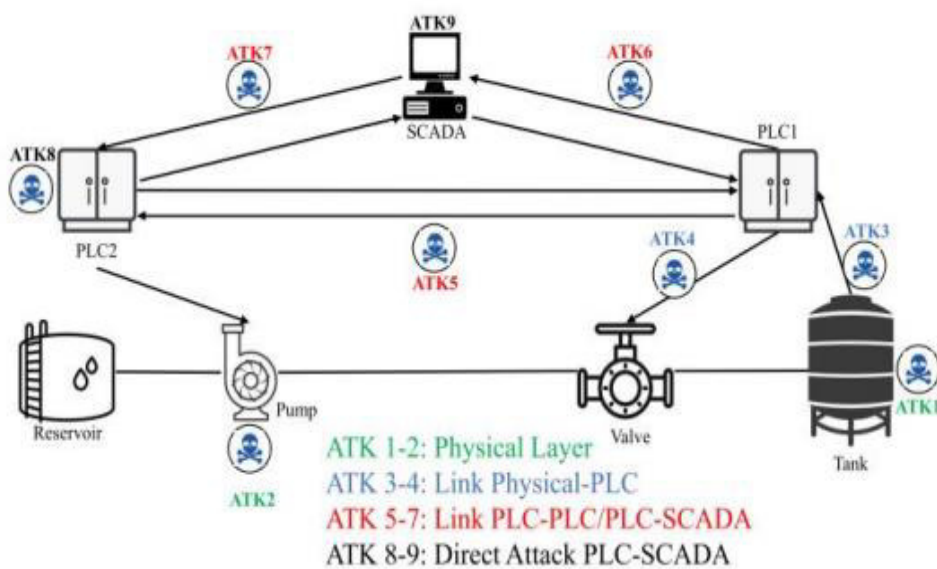


Figura 9. Tipos de ataques.

Fuente. Datos tomados de Taormina et al. (2018).

A continuación, se detallan los tipos de ataques según Taormina et al. (2018):

ATK 1: Ataque físico a un sensor

Se supone que el atacante puede acceder al sensor físicamente para poder llevar a cabo este ataque. Por lo tanto, este sensor se puede manipular, dañar o incluso reemplazar. En consecuencia, las lecturas de PLC asociadas pueden ser erróneas con ese sensor, lo que puede comprometer el control de una parte de la instalación.

ATK 2: Ataque directo contra un actuador

El atacante puede activar o desactivar el actuador a voluntad o cambiar su lugar de operación (como cuando una válvula se abre o si la bomba trabaja a mayor velocidad), lo que puede afectar la confiabilidad del sistema.

ATK 3: Ataque conexión PLC-sensor

PLC y sensores se pueden conectar por cableado o sin cable. El atacante podría llevar a cabo sus acciones con acceso físico al sistema, dependiendo del tipo de conexión. Estas están generalmente destinadas a manipulación de paquetes de datos y recopilación de información sobre la situación del sistema o interrumpir la conexión (denegación de servicio).

ATK 4: Ataque conexión PLC-actuador

Teniendo en cuenta los mismos factores que el ataque PLC-sensor sobre las diversas maneras de conexión, esta forma de ataque consigue tener un impacto en el control del equipo de campo. La completa pérdida de control del equipo, así como la modificación de las órdenes del PLC, pueden tener un impacto en los componentes de última etapa como bombas y válvulas, lo que puede tener consecuencias significativas para la actividad de instalación.

ATK 5: Ataque conexión PLC- PLC

Normalmente los PLC se conectan entre sí a través de una red privada para que puedan comunicarse sobre la situación del sistema. Uno de los casos posibles es cuando un PLC puede controlar una bomba de agua que se utiliza para bombear agua a un reservorio a través de sensores conectados a otros PLC. Este es el caso en el que ambos PLCs podrían comunicarse directamente entre sí, lo que permite al asaltante concentrarse en esa conexión para manipular, obtener o cortar la información que circula.

ATK 6: Ataque conexión PLC-SCADA

Este tipo de ataque, donde la conexión PLC-SCADA, que normalmente se realiza por medio de internet o una red propietaria, se impide transitoriamente, se manobra o se escucha en secreto, haciendo que el canal de comunicación quede saturado. En consecuencia, el SCADA recibe información acerca del estado del sistema inconclusa o equivocada.

ATK 7: Ataque conexión SCADA-PLC

Este tipo de ataque es comparable al tipo de ataque PLC-SCADA. El atacante puede haber manipulado (ataque de engaño), bloqueado (denegación de servicio), o interceptado las señales que el SCADA envía a un PLC.

ATK 8: Ataque contra un PLC

El ataque contra un PLC se da cuando el control directo del PLC que está en la red lo posee el atacador. El atacante puede alterar la lógica de control del PLC, detener completamente las operaciones habituales del proceso de control o informar deliberadamente al SCADA con datos incorrectos, dependiendo del nivel de control obtenido. Aunque se relaciona con algunos de los ataques mencionados (ATK 6 y ATK 7), este modo de ataque en específico es más constante.

ATK 9: Ataque al sistema SCADA

El ataque al SCADA se da cuando un atacante ha entrado en el sistema de supervisión control y adquisición de datos mediante un ataque tanto a nivel local como remoto. Por lo tanto, puede alterar sin restricciones todas las configuraciones del sistema y recopilar completamente cada uno de los datos que los sensores midieron.

2.2.3.2 Escenarios de ataques ciberfísicos

Los autores en Taormina et al. (2018) estudian las características de los ataques dependiendo del tiempo de duración, por ejemplo, la hora de inicio y finalización de un ataque específico, el número de ataques que ocurrieron y las ubicaciones de los componentes que serán atacados. Los investigadores utilizan diferentes herramientas para poder diseñar los ataques y personalizar los atributos para cada tipo de ataque. Se presentan los ataques en diferentes escenarios de la siguiente manera:

Escenario 1: Desbordamiento del tanque como resultado de un ataque directo a una estación de refuerzo. Si el atacante está físicamente en la estación de bombeo (ATK2) o si altera la señal de control enviada por PLC1 (ATK4) encendiendo las bombas o evitando que reciban una señal de parada, esta acción puede ocurrir. El atacante también puede controlar por completo PLC1 y controlar las señales de activación a voluntad (ATK8).

Escenario 2: Bajo nivel en un tanque como resultado de lecturas manipuladas del nivel del agua. Las lecturas del nivel de agua en el tanque T2 cambian, lo que evita que el tanque se llene de nuevo. La manipulación física del sensor de nivel de agua (ATK1) o la modificación del enlace de comunicación entre el sensor y PLC3 (ATK3) puede lograr esto.

Escenario 3: Desbordamiento del tanque como resultado de una modificación en la conexión entre PLC y PLC. Este es un ejemplo del ataque ATK5, en el que el atacante interrumpe la conexión PLC2 a PLC1 y cambia los valores del nivel de agua del tanque T1, lo que activa las bombas PU1 y PU2, lo que aumenta el nivel de agua del tanque T1.

Escenario 4: proteger la conexión PLC-SCADA de un ataque de reproducción. Un ataque destinado a ocultar sus efectos al SCADA se agrega al escenario 3. El enlace de comunicación PLC2 a SCADA (ATK6)

se puede atacar para lograr esto.

Escenario 5: Desbordamiento del tanque como resultado de configuraciones enviadas incorrectamente por SCADA. Este escenario de ataque implica modificar los paquetes enviados por SCADA para alterar las operaciones de un PLC supervisado (ATK7).

Escenario 6: Múltiples ataques aleatorios al PLC. El objetivo de este experimento último es provocar un desbordamiento en los tanques T2, T3 y T4 alterando las tendencias de nivel de agua que llegan al PLC3, que regula todos los actuadores que desvían el agua a estos tanques.

2.2.4. Conjuntos de datos

WADI

Para Mishra et al. (2019), el conjunto de datos de WADI (Water Distribution) se obtiene de la planta de distribución de agua (WADI) que representa una versión reducida de una gran red de distribución de agua en una ciudad. Está diseñado para suministrar 10 US galones/minuto de agua filtrada y es utilizado como banco de pruebas para la experimentación. Consta de 3 subprocesos: red primaria (P1) , red secundaria (P2) y red de agua de retorno (P3). Cada etapa es controlada por un controlador lógico programable. WADI consta de 103 sensores y actuadores para medir el nivel del agua, la calidad, el caudal, la presión y el estado de las válvulas y bombas.

C-Town

La red de C-Town fue introducida por primera vez en la "Batalla de las Redes de Calibración de Agua" por Ostfeld et al. (2012). Se basa en una red de distribución de agua real.

BATADAL

C-Town volvió a ser una red de referencia en la Batalla de algoritmos de detección de ataques (BATADAL), una competencia internacional sobre seguridad cibernética de los sistemas de distribución de agua (Taormina et. al. 2018). Los conjuntos de datos proporcionados son algunos de los pocos casos de conjuntos de datos de código abierto disponibles para la investigación en ciberseguridad de redes de agua, y siguen estando disponibles para el público después del concurso BATADAL. Más investigadores utilizaron esos conjuntos de datos como base para investigar los ataques ciberfísicos en las redes de distribución de agua (WDN) (Water Distribution Networks).

E-Town

E-Town se basó en una WDN real de una ciudad colombiana. En 2016, durante la International Conference on Water Distribution System Analysis (WDSA), el modelo se presentó en el concurso Battle of Water Networks District Metered Areas (BWNDMA). E-Town es un WDN complejo y a gran escala con cinco fuentes de agua principales, 17 tanques, tres estaciones de bombeo, 14 válvulas de control, 11107 nodos y 13920 tuberías (Rodríguez Martínez et al., 2023).

Cuadro 1. Datasets más usados por los investigadores

Dataset	Artículos	Conteo
BATADAL	Brentan et al. (2021)	6
	Choi et al. (2020)	
	Kravchik et al. (2022)	
	Mahmoud et al. (2022)	
	Ramotsoela et al. (2020)	
	Tsiami et al. (2021)	

C-town	Kadosh et al. (2020) Moazeni et al. (2021a) Moazeni et al. (2021b) Nikolopoulos et al. (2022) Housh et al. (2022) Rodríguez-M. et al. (2023)	6
E-town	Kadosh et al. (2020) Housh et al. (2022) Rodríguez-M. et al. (2023)	3
WADI	Ahmed et al. (2022) Kravchik et al. (2022)	2

En el cuadro 1 se evidencian los datasets más usados por los autores mencionados en los antecedentes de este estudio.

2.2.5. Métricas para evaluación del modelo

La matriz de confusión

Para Barrios, Juan (2019b), la matriz de confusión es un instrumento en el campo de la inteligencia artificial y el aprendizaje automático que permite la visualización del rendimiento de un algoritmo de aprendizaje supervisado. En términos prácticos, cada fila representa una clase en lo real, mientras que el número de predicciones de cada clase se muestra en cada columna de la matriz. De esta manera, se puede observar que nuestro modelo se basa en una variedad de tipos de errores y aciertos al proceder por el proceso de aprendizaje con los datos.

		Valores Actuales	
		1(Positivo)	0 (Negativo)
Valores de Predicción	1(Positivo)	TP (True Positive)	FP (False Positive) <i>Error tipo I</i>
	0 (Negativo)	FN (False Negative) <i>Error tipo II</i>	TN (True Negative)

Figura 10. **Matriz de confusión.**

Fuente. Datos tomados de Barrios Arce (2019b).

- + **Verdadero positivo (TP):** Cuando el sistema está bajo ataque y la predicción del modelo también es que el sistema está bajo ataque.
- + **Verdadero negativo (TN):** Cuando el sistema no está bajo ataque y la predicción del modelo es también que el sistema no está bajo ataque.
- + **Falso positivo (FP):** Cuando el sistema no está siendo atacado y la predicción del modelo es que el sistema está siendo atacado.
- + **Falso negativo (FN):** Cuando el sistema está bajo ataque y la predicción del modelo es que el sistema no está bajo ataque.

Recall

Recall o exhaustividad es una medida que describe cuántas de las predicciones "verdaderas" para todos los puntos de datos eran realmente "verdaderas". En otras palabras, mide la capacidad del modelo para que clasifique correctamente el estado del sistema de distribución de agua como atacado. La exhaustividad también se conoce como sensibilidad o tasa de verdaderos positivos.

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

Precisión

La precisión es una medida que describe cuántas de las predicciones verdaderas son realmente verdaderas. En otras palabras, mide la capacidad del modelo para evitar falsas alarmas.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Accuracy

Accuracy o exactitud es una medida que describe cuántas predicciones se clasificaron correctamente en todo el conjunto de datos. Esta métrica solo aporta valor real si el número de muestras entre cada clase está balanceado. Además, esta métrica tampoco es indicada cuando el peso real de cada clase no es el mismo.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (3)$$

F1-Score

La puntuación F1 es una media armónica ponderada de precisión y recall (recuerdo o sensibilidad), con la puntuación más alta de 1,0 y la más baja de 0,0. En general, las medidas de precisión tienen puntajes F1 más bajos porque incorporan precisión y recuerdo en su cálculo.

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (4)$$

Score of Time-To-Detection (s_{TTD})

El s_{TTD} indica qué tan rápido el algoritmo puede detectar ataques y es una medida de cuándo se marcó el ataque por primera vez en relación a cuándo comenzó.

CAPÍTULO 3: METODOLOGÍA

3.1 Identificación de variables

3.1.1 Variable independiente

Modelo de detección inteligente basado en autoencoders no supervisados con restricciones de aseguramiento.

3.1.2 Variable dependiente

Detección de ataques cibernéticos en sistemas de distribución de agua SCADA.

3.2 Operacionalización de las variables

Cuadro 2. Operacionalización de las variables

Variable	Tipo de Variable	Definición Conceptual	Dimensión	Indicador	Descripción del indicador
Modelo de detección inteligente basado en autoencoders no supervisados con restricciones de aseguramiento.	Independiente	Es el resultado del proceso de entrenamiento y se define como la representación matemática del proceso del mundo real. Los algoritmos de aprendizaje automático encuentran los patrones en el conjunto de datos de entrenamiento, que se utiliza para aproximar la función de destino y es responsable de asignar las entradas a las salidas del conjunto de datos disponible.(Pedamkar, Priya, 2023).	Tiempo para la detección (Time-to-detection)	Cantidad de tiempo para la detección.	Es la diferencia entre el tiempo en el que se detecta el ataque y el tiempo en el que comienza el ataque.
			Exactitud (Accuracy)	Porcentaje de predicciones correctas.	Es la proporción de predicciones realizadas correctamente entre el total de predicciones realizadas.
			F1 Score (Índice de Predicción)	Porcentaje de predicciones positivas correctas.	Es el porcentaje de predicciones positivas correctas, usa la precisión y la sensibilidad.
			Precisión	Porcentaje de predicciones positivas	Es la precisión de las predicciones positivas.
			Exhaustividad	Porcentaje de predicciones positivas reales.	Es el porcentaje de predicciones positivas reales o la tasa de positivos verdaderos.
Detección de ciberataques en sistemas SCADA de distribución de agua.	Dependiente	Permite identificar eventos no deseados : robar, exponer, alterar, inhabilitar o destruir información mediante el acceso no autorizado a los sistemas (IBM.com).			

3.3 Matriz de Consistencia

Cuadro 3. Matriz de Consistencia

Problemas	Objetivos	Hipótesis
Problema General ¿De qué manera el diseño de un modelo de detección inteligente basado en algoritmos y técnicas de Machine Learning, permite detectar tempranamente ataques cibernéticos en sistemas SCADA de distribución de agua, con un desempeño óptimo en comparación con modelos autoencoder o no supervisados?	Objetivo general Diseñar un modelo de detección inteligente, basado en algoritmos y técnicas de Machine Learning, que permita detectar tempranamente ataques cibernéticos en sistemas SCADA de distribución de agua, con un desempeño óptimo en comparación con modelos autoencoder o no supervisados.	Hipótesis general El diseño de un modelo de detección inteligente, basado en algoritmos y técnicas de Machine Learning, permite detectar tempranamente ataques cibernéticos en sistemas SCADA de distribución de agua, con un desempeño óptimo en comparación con modelos autoencoder o no supervisados.
Problemas Específicos PE1: ¿De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el menor tiempo de detección posible en comparación con estudios basados en modelos autoencoder o modelos no supervisados? PE2: ¿De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mayor exactitud en comparación con estudios basados en modelos autoencoder o modelos no supervisados? PE3: ¿De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mejor puntuación F1, es decir, con mejores predicciones de detección de ataques en comparación con estudios basados en modelos autoencoder o modelos no supervisados? PE4: ¿De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor precisión en comparación con estudios basados en modelos autoencoder o modelos no supervisados? PE5: ¿De qué manera el diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor exhaustividad en comparación con estudios basados en modelos autoencoder o modelos no supervisados?	Objetivos específicos OE1: Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el menor tiempo de detección posible en comparación con estudios basados en modelos autoencoder o modelos no supervisados. OE2: Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mayor exactitud en comparación con estudios basados en modelos autoencoder o modelos no supervisados. OE3: Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mejor puntuación F1, es decir, con mejores predicciones de detección de ataques en comparación con estudios basados en modelos autoencoder o modelos no supervisados. OE4: Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor precisión en comparación con estudios basados en modelos autoencoder o modelos no supervisados. OE5: Diseñar un modelo de detección inteligente que permita detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor exhaustividad en comparación con estudios basados en modelos autoencoder o modelos no supervisados.	Hipótesis específicas HE1: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el menor tiempo de detección posible en comparación con estudios basados en modelos autoencoder o modelos no supervisados. HE2: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mayor exactitud en comparación con estudios basados en modelos autoencoder o modelos no supervisados. HE3: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mejor puntuación F1, es decir, con mejores predicciones de detección de ataques en comparación con estudios basados en modelos autoencoder o modelos no supervisados. HE4: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor precisión en comparación con estudios basados en modelos autoencoder o modelos no supervisados. HE5: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor exhaustividad en comparación con estudios basados en modelos autoencoder o modelos no supervisados.

3.4 Tipo y Diseño de investigación

Según Vara (2012), el tipo de investigación a desarrollar es aplicada porque identifica una situación-problema y busca una de las posibles soluciones, que puede ser la más adecuada para el contexto específico. Además, el enfoque es cuantitativo. Este enfoque cuantitativo se caracteriza por plantear las relaciones entre las variables con el objetivo de obtener propuestas precisas y hacer sugerencias específicas Hernández et al. (2014). Asimismo, es importante destacar que esta investigación es de nivel no experimental, puesto que busca comprender por qué ocurre el problema y la relación de causa-efecto entre las variables. (Ñaupas Paitán et al., 2014).

3.5 Unidad de análisis

Base de datos con datos históricos de un sistema SCADA de tamaño mediano diseñada para parecerse a las operaciones de las redes de distribución de agua del mundo real.

3.6 Población de Estudio

Comprende datos SCADA históricos generados durante un período de un año de lecturas de 43 sensores por hora.

3.7 Tamaño de muestra

Se utilizarán tres conjuntos de datos independientes para entrenar, validar y probar el algoritmo, respectivamente. Se describen a continuación los 3 conjuntos de datos de lecturas de 43 sensores por hora.

Dataset de entrenamiento : Conjunto de datos que proporciona 1 año de lecturas sin ataques.

Dataset de validación: El conjunto de datos incluye lecturas con siete ataques simulados etiquetados dentro de un período de 6 meses.

Dataset de prueba : Conjunto de datos que consta de 3 meses de lecturas de sensores, con siete ataques simulados.

En este trabajo de investigación, la muestra es de tipo no probabilístico. Según Hernández et al. (2014), una muestra es no probabilística cuando la elección de los elementos no depende de la probabilidad, sino de causas que tienen relación con las características de la investigación y depende del proceso de toma de decisiones del investigador.

3.8 Selección de muestra

Los tres conjuntos seleccionados pertenecen a un simulador de una red de abastecimiento o distribución de agua potable (WDN) realista de tamaño mediano llamado C-town y se probaron en la competencia batalla de algoritmos de detección de ataques (Battle of the Attack Detection Algorithms (BATADAL)) que demuestra un estudio de caso casi realista de un WDS.

3.9 Técnicas de recolección de datos

Los conjuntos de datos de C-town se obtuvieron al realizar una revisión sistemática de la literatura, en la cual C-town ha tenido la mayor cantidad de menciones (6) que los otros conjuntos de datos utilizados por los investigadores.

3.10 Análisis e interpretación de la información

Un componente importante del balance hídrico en los sistemas de distribución es la demanda de agua. Para obtener un buen rendimiento del modelo, se entrena un algoritmo clasificador para aprender la variabilidad de la demanda diaria. Se propone un clasificador dedicado para cada hora del día; por ello, se eligieron los conjuntos de datos C-town tanto para entrenar como para validar y probar el modelo. Se toma como base el estudio de investigación de (Kadosh et al., 2020).

3.11 Metodología de implementación

En este paso se describe la metodología a usarse para la implementación del diseño del modelo.

3.11.1 Descripción del Dataset

En este paso se realiza una descripción de los datasets obtenidos del sistema de distribución de agua C-town seleccionado.

3.11.2 Preprocesamiento de datos

En este paso se realiza la normalización de los datos y luego el balanceo de estos.

3.11.3 Entrenamiento

En este paso se entrena el dataset de validación que contiene datos etiquetados, así como se entrena el conjunto de datos de prueba sin datos

etiquetados.

3.11.4 Evaluación

En este paso se hace uso de las métricas descritas en el marco teórico, además, se realiza una comparación del rendimiento con otros algoritmos de estudios de investigación similares a este estudio.

3.11.5 Modelo de detección

En este paso se describe el algoritmo de detección, así como los parámetros que se utilizaron y se muestra parte del código.

CAPÍTULO 4: IMPLEMENTACIÓN

4.1 Propuesta Técnica para la integración

En este apartado se describe una propuesta técnica para la integración del modelo en un sistema SCADA de distribución de agua real. Esto incluye la arquitectura de software, metodología de desarrollo de software, identificación de stakeholders y herramientas tecnológicas.

Arquitectura de Software

Se propone la siguiente arquitectura de software que incluye el desarrollo de un modelo de detección de ciberataques en sistemas SCADA de distribución de agua utilizando autoencoders .

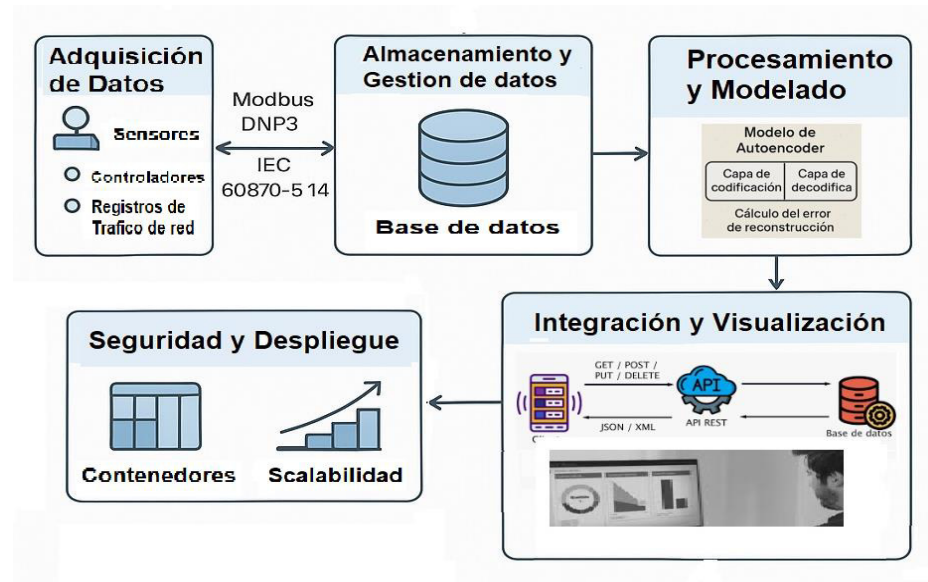


Figura 11. Arquitectura de software para integración.

Esta arquitectura mostrada en la figura 11 se divide en varios componentes los cuales se describen a continuación:

Adquisición de Datos

- **Fuentes de datos:** Sensores, controladores, registros de tráfico de red SCADA.
- **Protocolo de comunicación:** Modbus, DNP3, IEC 60870-5-104, u otros utilizados en SCADA.
- **Módulo de preprocesamiento:** Filtrado y limpieza de datos, conversión de datos brutos a formatos adecuados para el modelo (tablas, series temporales), eliminación de valores atípicos no relevantes.

Almacenamiento y Gestión de Datos

- **Bases de datos:** PostgreSQL, InfluxDB (para datos de series temporales), MongoDB (si se manejan estructuras no relacionales).
- **Módulo de etiquetado:** Identificación de eventos normales y anómalos si se cuenta con datos etiquetados.
- **Pipeline de datos:** Uso de herramientas como Apache Kafka o RabbitMQ para manejar grandes volúmenes de datos en tiempo real.

Procesamiento y Modelado

- **Arquitectura de autoencoder:** Red neuronal con codificador y decodificador, entrenamiento con datos normales para detectar anomalías en el comportamiento de la red SCADA.
- **Técnicas adicionales:** Algoritmos de reducción de dimensionalidad (PCA, t-SNE) para optimizar el input, uso de redes neuronales recurrentes (LSTM) si se trabaja con series temporales.
- **Módulo de detección de anomalías:** Comparación entre entrada y salida del autoencoder, cálculo de error de reconstrucción para definir umbrales de detección de ataques.

Integración y Visualización

- **API de acceso a resultados:** Desarrollo de una API REST con Flask o FastAPI para consultar los eventos detectados.
- **Dashboard de monitoreo:** Visualización en tiempo real con Grafana o Streamlit.
- **Alertas y notificaciones:** Sistema de notificación por correo, Slack, o integración con SIEMs (Splunk, ELK Stack).

Seguridad y Despliegue

- **Contenedorización:** Docker para facilitar el despliegue.
- **Escalabilidad:** Kubernetes para manejar cargas de datos en entornos industriales.
- **Autenticación y control de acceso:** Seguridad basada en roles para evitar accesos no autorizados.

Metodología de desarrollo de software

Debido a que el proyecto involucra aprendizaje automático y un entorno crítico como SCADA, una combinación de Desarrollo Ágil y CRISP-DM (Cross-Industry Standard Process for Data Mining) permite construir un sistema robusto y adaptable para la detección de ataques en SCADA, asegurando que el modelo de autoencoder se optimice continuamente con nuevos datos y responda a los desafíos de seguridad en infraestructuras críticas.

➤ Enfoque Ágil (Scrum o Kanban)

Este enfoque permite la iteración continua, la adaptación a nuevos hallazgos en los datos y la integración con los sistemas SCADA en ciclos cortos.

Ventajas: Flexibilidad, mejora continua y entrega incremental.

Aplicación: Se desarrollan módulos en iteraciones, probando cada parte del sistema (adquisición de datos, preprocesamiento, autoencoder, detección y respuesta).

➤ **CRISP-DM para la parte de Machine Learning**

Este estándar es ampliamente usado en proyectos de análisis de datos y modelos de inteligencia artificial. Se compone de las siguientes fases:

- **Comprensión del negocio:** Definir los requisitos de seguridad y los tipos de ataques a detectar.
- **Comprensión de los datos:** Analizar datos de sensores SCADA para identificar patrones normales y anómalos.
- **Preparación de los datos:** Preprocesamiento, normalización y selección de características.
- **Modelado:** Entrenamiento del autoencoder y ajuste de hiperparámetros.
- **Evaluación:** Validación del modelo con datos reales y ajuste de umbrales de detección.
- **Despliegue:** Integración con el sistema SCADA y monitoreo continuo.

Identificación de stakeholders

En el desarrollo del software que incluye un modelo de autoencoder para la detección de ataques en un entorno SCADA, participan varios actores o stakeholders, cada uno con roles y responsabilidades específicos, que se describen a continuación.

➤ **Administrador del Sistema SCADA**

- Responsable del funcionamiento del sistema SCADA.
- Asegura que la integración del modelo de autoencoder no afecte

el rendimiento del sistema.

- Supervisa la implementación de medidas de seguridad.

➤ **Equipo de Desarrollo de Software**

- Diseña y desarrolla el software del modelo de autoencoder.
- Integra el modelo con los sistemas SCADA existentes.
- Asegura la compatibilidad y optimización del código.

➤ **Ingenieros de Ciberseguridad**

- Definen los requisitos de seguridad para la detección de ataques.
- Analizan amenazas y patrones de comportamiento anómalo en SCADA.
- Evalúan la efectividad del modelo de detección.

➤ **Científicos de Datos / Ingenieros de Machine Learning**

- Diseñan y entrenan el modelo de autoencoder.
- Seleccionan las mejores técnicas de preprocesamiento y ajuste de hiperparámetros.
- Evalúan la precisión y detección de anomalías del modelo.

➤ **Operadores del Sistema SCADA**

- Monitorean el sistema en tiempo real.
- Interactúan con la interfaz del software de detección de anomalías.
- Actúan ante alertas de posibles ciberataques.

➤ **Gerentes o directores de Tecnología**

- Toman decisiones estratégicas sobre la implementación del sistema.
- Asignan presupuesto y recursos para el desarrollo del software.
- Supervisan la alineación del proyecto con los objetivos organizacionales.

➤ **Audidores y Reguladores de Seguridad**

- Verifican el cumplimiento de normativas y estándares de ciberseguridad.
- Evalúan el impacto del modelo en la seguridad del sistema

SCADA.

- Emiten recomendaciones y certificaciones de seguridad.

➤ **Usuarios Finales y Clientes**

- Empresas o entidades gubernamentales que usan SCADA en sectores críticos (agua, energía, manufactura).
- Reciben informes y alertas sobre anomalías detectadas.
- Evalúan la eficacia del sistema en la protección de infraestructuras críticas.

Herramientas tecnológicas

Para la implementación real del software para la detección de ataques en un entorno SCADA, es crucial seleccionar herramientas tecnológicas adecuadas tanto en **hardware** como en **software**.

❖ **Hardware**

Dado que el sistema SCADA opera en tiempo real, el hardware debe garantizar alta disponibilidad, procesamiento eficiente y seguridad.

Infraestructura de Procesamiento

- **Servidores Edge Computing** (Para análisis en tiempo real sin depender de la nube). Ejemplo: Dell Edge Gateway 5200, HPE Edgeline EL4000, Siemens IPC227E.
- **Servidores On-Premise o en la Nube** (Para entrenamiento y almacenamiento del modelo). Ejemplo: HPE ProLiant DL380, Dell PowerEdge R740, Google Cloud AI, AWS EC2 con GPU.

Dispositivos de Monitoreo y Control

- **Controladores Lógicos Programables (PLC)**. Ejemplo: Siemens S7-1200, Rockwell Automation ControlLogix, Schneider Electric Modicon.

- **Dispositivos IoT para Sensores.** Ejemplo: Raspberry Pi 4, NVIDIA Jetson Nano (para inferencia en el borde).

❖ **Software**

Para el desarrollo del software del autoencoder y su integración con SCADA, se deben considerar herramientas de desarrollo, entrenamiento, monitoreo y seguridad.

Frameworks y Librerías de Machine Learning

- **TensorFlow / Keras** (Para entrenar el modelo de autoencoder).
- **PyTorch** (Alternativa potente con mayor flexibilidad).
- **Scikit-learn** (Para preprocesamiento de datos y selección de características).

Bases de Datos

- **InfluxDB / TimescaleDB** (Optimizadas para series temporales en SCADA)
- **MongoDB** (Para almacenar eventos de detección de anomalías)
- **PostgreSQL** (Para reportes e integración con otras herramientas)

Plataformas de Integración con SCADA

- **Ignition SCADA** (Compatible con Python y bases de datos avanzadas)
- **Wonderware System Platform** (Para entornos industriales grandes)
- **OpenPLC / Node-RED** (Para simulaciones y pruebas de integración)

Herramientas de Seguridad y Monitoreo

- **Wireshark** (Para analizar el tráfico de la red SCADA)
- **Snort / Suricata** (Sistemas de detección de intrusos para SCADA)

- **Grafana / Kibana** (Para visualizar anomalías detectadas en tiempo real).

4.2 Metodología de Implementación del Modelo

La metodología para la implementación de este estudio está asentada en las metodologías de la revisión sistemática de la literatura, en las fases descritas en cada uno de los artículos y también en una metodología para la detección de anomalías en sistemas de control industrial llamada MADICS, por sus siglas en inglés (Methodology for Anomaly Detection in Industrial Control Systems), propuesta por (Perales Gómez et al., 2020).

Se respalda la metodología con el siguiente cuadro donde se visualizan los pasos a seguir de los diferentes estudios encontrados en la literatura.

El cuadro 4 muestra cada uno de los artículos y la selección de las siguientes fases que serán desarrolladas en esta tesis.

Cuadro 4. Fases de las metodologías de artículos del estado del arte

Artículos Relacionados	Recogida de datos	Preproce_ samiento de datos	Extracción de caracte_ rísticas	Selección de caracte_ rísticas	Etiquetado	Entrena_ miento	Evaluación	Detección
Suaboot et al. (2021)	✓		✓	✓	✓	✓	✓	✓
Mokhtari et al. (2021)	✓			✓		✓	✓	✓
Priyanga et al. (2019)	✓			✓		✓	✓	✓
Selim et al. (2021)	✓	✓	✓	✓		✓	✓	✓

Rodríguez et al. (2023)	✓				✓	✓	✓
Kravchik et al. (2022)	✓		✓		✓	✓	✓
M. R. et al. (2022)	✓	✓			✓	✓	✓
Almajed et al. (2022)	✓	✓	✓		✓	✓	✓
Wang, W., et al. (2022)	✓	✓		✓	✓	✓	✓

MADICS

MADICS es una metodología de detección de anomalías basada en técnicas semisupervisadas de aprendizaje automático y aprendizaje profundo para el campo de los sistemas de control industrial.

Dado que un ataque cibernético es un evento que rara vez ocurre y, por lo tanto, puede considerarse como una anomalía, generalmente se utiliza el paradigma de detección de anomalías para detectar ataques cibernéticos.

MADICS está dividido en cinco pasos:

- Preprocesamiento del conjunto de datos
- Filtrado y de características
- Extracción de características
- Selección de método de detección de anomalías
- Validación

La Metodología propuesta selecciona la mayoría de los pasos de MADICS, pero en algunas fases como recogida de datos, preprocesamiento de datos

y evaluación o validación de datos se utilizó la guía de Selim et al. (2021), Almajed et al. (2022) y Rodríguez et al. (2023).

A continuación, se muestra un gráfico de la metodología propuesta

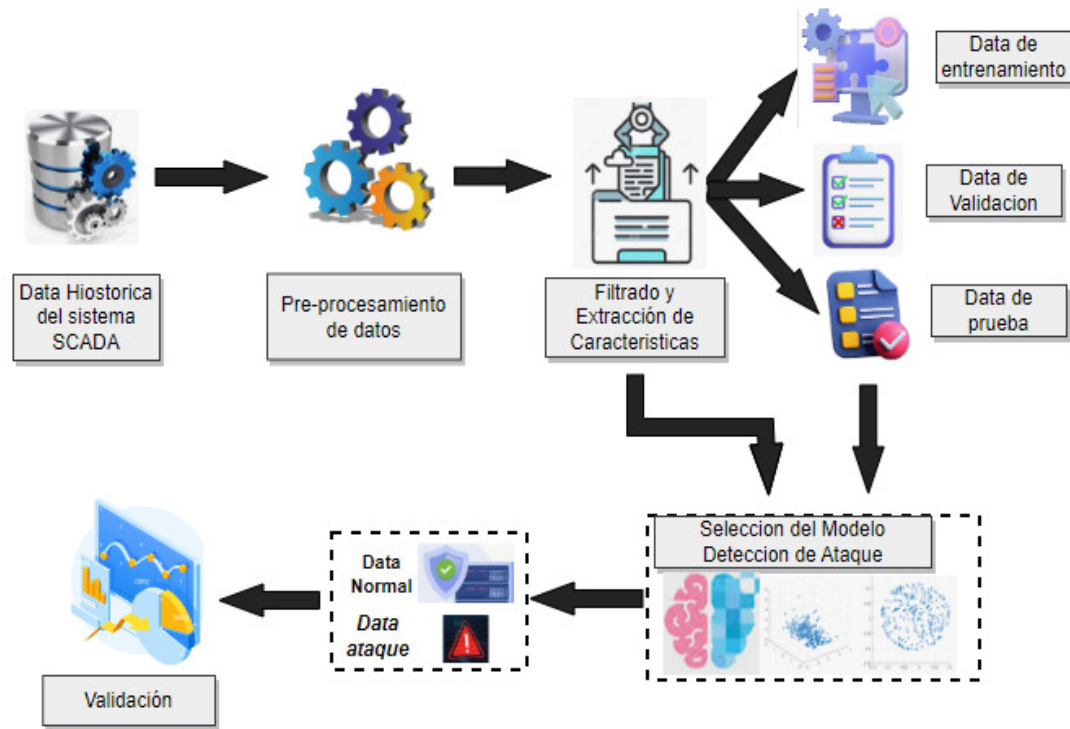


Figura 12. Metodología para la detección de ataques.

4.3 Descripción del Conjunto de Datos (Dataset)

El principal motivo para elegir este conjunto de datos es que se ajusta al alcance de esta tesis, enfocada en los sistemas de distribución de agua. Además, se eligió C-Town por las siguientes razones:

- ❖ Es de tamaño mediano ya que tiene un número suficiente de datos

para realizar el entrenamiento y las pruebas.

- ❖ El conjunto de datos está validado y es de los años 2016-2017.
- ❖ Contiene una descripción más detallada y precisa de los ataques y anomalías.

En este estudio se utilizaron 3 conjuntos de datos de la red de distribución de agua C-town que se muestra en la figura 13.

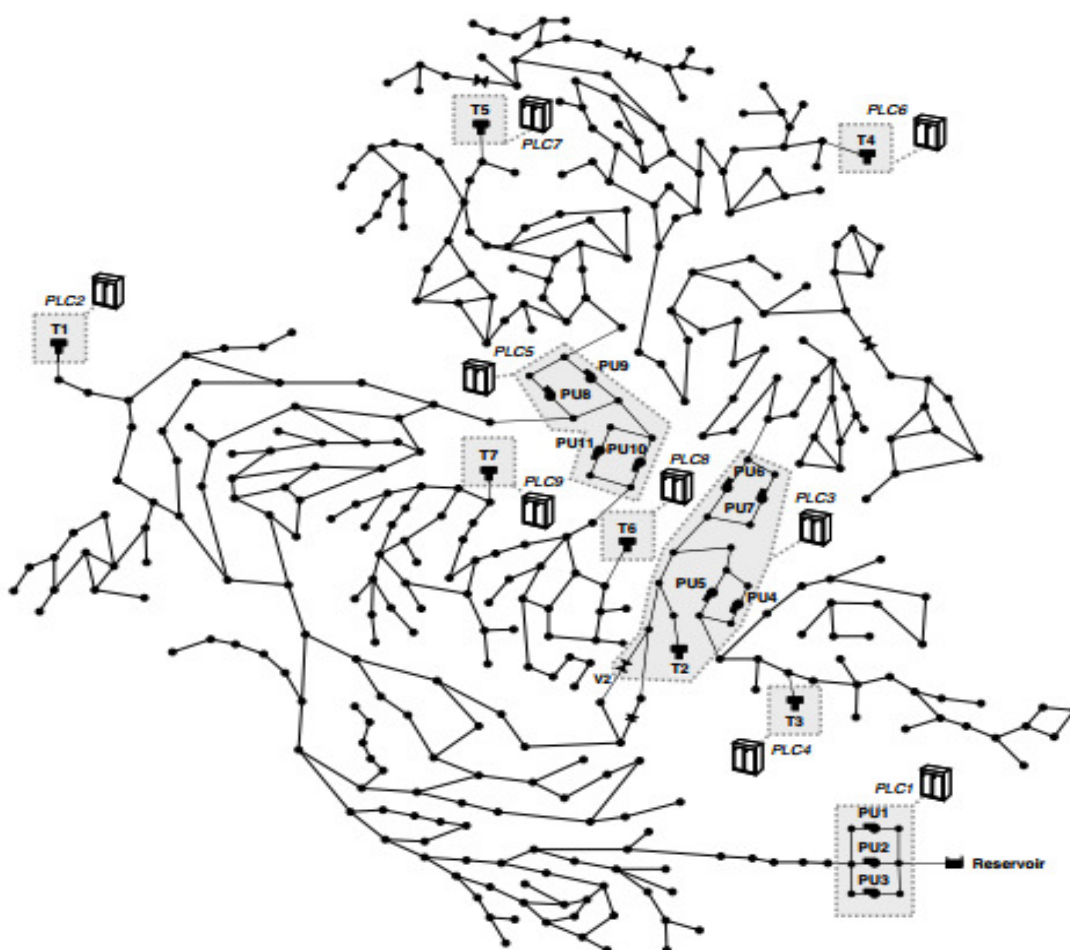


Figura 13. Red de distribución de agua C-town.

Fuente. Datos tomados de Taormina et al. (2018).

La red de distribución de agua C-town consta de un depósito, siete tanques

Cuadro 5. Variables de los conjuntos de datos C-town

ID	VARIABLE	DESCRIPCIÓN	UNIDAD	PLC	TIPO DATO
1	L_T1	Nivel del tanque T1	m	PLC2	Float
2	L_T2	Nivel del tanque T2	m	PLC3	Float
3	L_T3	Nivel del tanque T3	m	PLC4	Float
4	L_T4	Nivel del tanque T4	m	PLC6	Float
5	L_T5	Nivel del tanque T5	m	PLC7	Float
6	L_T6	Nivel del tanque T6	m	PLC8	Float
7	L_T7	Nivel del tanque T7	m	PLC9	Float
8	F_PU1	Caudal en bomba PU1	L/s	PLC1	Float
9	S_PU1	Estado de bomba PU1	-	PLC1	Boolean
10	F_PU2	Caudal en bomba PU2	L/s	PLC1	Float
11	S_PU2	Estado de bomba PU2	-	PLC1	Boolean
12	F_PU3	Caudal en bomba PU3	L/s	PLC1	Float
13	S_PU3	Estado de bomba PU3	-	PLC1	Boolean
14	F_PU4	Caudal en bomba PU4	L/s	PLC3	Float
15	S_PU4	Estado de bomba PU4	-	PLC3	Boolean
16	F_PU5	Caudal en bomba PU5	L/s	PLC3	Float
17	S_PU5	Estado de bomba PU5	-	PLC3	Boolean
18	F_PU6	Caudal en bomba PU6	L/s	PLC3	Float
19	S_PU6	Estado de bomba PU6	-	PLC3	Boolean
20	F_PU7	Caudal en bomba PU7	L/s	PLC3	Float
21	S_PU7	Estado de bomba PU7	-	PLC3	Boolean
22	F_PU8	Caudal en bomba PU8	L/s	PLC5	Float
23	S_PU8	Estado de bomba PU8	-	PLC5	Boolean
24	F_PU9	Caudal en bomba PU9	L/s	PLC5	Float
25	S_PU9	Estado de bomba PU9	-	PLC5	Boolean
26	F_PU10	Caudal en bomba PU10	L/s	PLC5	Float
27	S_PU10	Estado de bomba PU10	-	PLC5	Boolean
28	F_PU11	Caudal en bomba PU11	L/s	PLC5	Float
29	S_PU11	Estado de bomba PU11	-	PLC5	Boolean
30	F_V2	Caudal en válvula V2	L/s	PLC3	Float
31	S_V2	Estado de válvula V2	-	PLC3	Boolean
32	P_J280	Presión en nudo J280	m	PLC1	Float
33	P_J269	Presión en nudo J269	m	PLC1	Float
34	P_J300	Presión en nudo J300	m	PLC3	Float

35	P_J256	Presión en nudo J256	m	PLC3	Float
36	P_J289	Presión en nudo J289	m	PLC3	Float
37	P_J415	Presión en nudo J415	m	PLC3	Float
38	P_J302	Presión en nudo J302	m	PLC5	Float
39	P_J306	Presión en nudo J306	m	PLC5	Float
40	P_J307	Presión en nudo J307	m	PLC5	Float
41	P_J317	Presión en nudo J317	m	PLC5	Float
42	P_J14	Presión en nudo J14	m	PLC3	Float
43	P_J422	Presión en nudo J422	m	PLC3	Float

En los cuadros 6 y 7 se muestran los 7 ataques del conjunto de datos denominado conjunto de datos de validación y del conjunto de datos de prueba, respectivamente.

Cuadro 6. Ataques del conjunto de datos BATADAL_dataset04

ID	Fecha de inicio (DD/MM/YYYY HH)	Fecha de fin (DD/MM/YYYY HH)	Duración (horas)	Descripción	Ocultación SCADA	Etiqueta (horas)
1	13/09/2016 23	16/09/2016 00	50	El atacante altera la transmisión SCADA a PLC9 y cambia los umbrales de L_T7 que determinan cuándo se encienden/apagan las bombas PU10/PU11. Niveles bajos en T7.	Se repite el ataque en L_T7.	42
2	26/09/2016 11	27/09/2016 10	24	Igual al ataque Nro.1	Igual al ataque Nro.1 pero el ataque de repetición en el flujo y estado de	0

					PU10/PU1 1	
3	09/10/2016 09	11/10/2016 20	60	El ataque altera lecturas de L_T1 enviadas por el PLC2 al PLC1, que lee un nivel bajo constante y mantiene las bombas PU1/PU2 activado. Rebosadero en T1.	Polilínea para compensar el aumento de L_T1.	60
4	29/10/2016 19	02/11/2016 16	94	Igual al ataque Nro.3	Ataque de repetición en L_T1, flujo y estado de PU1/PU2, así como en presión a la salida de las bombas (P_J269).	37
5	26/11/2016 17	29/11/2016 04	60	La velocidad de trabajo de PU7 se redujo a 0,9 de la velocidad nominal. Niveles de agua más bajos en T4.		7
6	06/12/2016 07	10/12/2016 04	94	Igual al ataque Nro.5, pero la velocidad se reduce hasta 0.7.	Repetición del ataque en L_T4.	73
7	14/12/2016 15	19/12/2016 04	110	Igual al ataque Nro.6	Repetición del ataque en L_T4, así como en el flujo y el estado de PU6/PU7.	0

Cuadro 7. Ataques del conjunto de datos BATADAL_test_dataset

ID	Fecha de inicio (DD/MM/YYYY HH)	Fecha de fin (DD/MM/YYYY HH)	Duración (horas)	Descripción	Ocultación SCADA
1	16/01/2017 09	19/01/2017 06	70	El atacante obtiene el control del PLC3 y cambia los umbrales de L_T3 que determinan cuándo se encienden/apagan las bombas PU4/PU5. Niveles bajos en T3.	Ataque de repetición en L_T3, así como en el flujo y estado de PU4/PU5.
2	30/01/2017 08	02/02/2017 00	65	Se altera las de lecturas de L_T2 que llegan al PLC3, que lee un nivel bajo y mantiene la válvula V2 Abierta. El ataque lleva a T2 a desbordarse	Polilínea para compensar el aumento en L_T2.
3	09/02/2017 03	10/02/2017 09	31	Activación maliciosa de la bomba PU3.	-
4	12/02/2017 01	13/02/2017 07	31	Igual al ataque Nro.3.	-
5	24/02/2017 05	28/02/2017 08	100	Igual al ataque Nro.2.	Ataque de repetición en L_T2, flujo y estado V2, así como en lecturas de presión de entrada y salida V2 (P_J14, P_J422)
6	10/03/2017 14	13/03/2017 21	80	El atacante obtiene el control del PLC5 y cambia los umbrales de	Ataque de repetición en L_T7, flujo y estado de

				L_T7 que determinan cuándo se encienden/apagan las bombas PU10/PU11. Las bombas se ven obligadas a encenderse/apagarse continuamente durante el ataque.	PU10/PU11, así como en lecturas de presión de entrada y salida de bombas (P_J14, P_J422). La ocultación de la presión de entrada termina antes que la de otras variables
7	25/03/2017 20	27/03/2017 01	30	Alteración de la señal T4 que llega al PLC6. Rebosadero en T6.	-

4.4 Preprocesamiento de los datos

Para realizar el preprocesamiento de datos se ha utilizado una laptop con las siguientes características: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz con 8 GB de RAM, un sistema operativo de 64 bits, procesador x64 con una edición de Windows 11 Home Single Language.

Además, se ha usado el lenguaje de programación Python, que es un lenguaje de programación multiparadigma de código abierto, el cual permite una programación funcional, orientada a objetos e imperativa. Ha sido elegida debido a que es uno de los lenguajes más usados en el campo de machine learning, siendo uno de sus usos la seguridad. Las librerías para usar en este paso son pandas, que sirve para la manipulación de datos de alto nivel, scikit-learn, que provee de algoritmos de machine learning, herramientas de preprocesamiento de datos y métricas para evaluar los modelos entrenados.

El objetivo primordial de este paso es preparar el conjunto de datos que se va a utilizar con el modelo, siguiendo la metodología MADICS se requiere 3

conjuntos de datos los cuales se describe en la sección 4.1 los cuales son: BATADAL_dataset03 (entrenamiento), BATADAL_dataset04 (validación), y BATADAL_test_dataset (prueba). El primero como su nombre indica sirve para entrenar el modelo teniendo solo datos normales sin ataques, mientras que el segundo conjunto de datos sirve para evaluar el rendimiento del modelo durante la optimización de hiperparámetros y contiene datos normales y con ataque, por último, el tercero, que es el conjunto de datos de prueba, se utiliza para validar la metodología con datos no vistos antes.

En este paso es de suma importancia tener el conjunto de datos limpio y libre de ruido, a continuación, se muestra los pasos a seguir en el preprocesamiento:

- ❖ Carga de datos: En Python se realiza usando la librería Pandas.
- ❖ Limpieza de datos: Es necesario eliminar los valores duplicados, nulos, y las columnas innecesarias.
- ❖ Transformación de datos: Convertir los datos categóricos en numéricos, normalizar los datos y reducir la dimensionalidad.
- ❖ Selección de características: Se seleccionan las características más significativas para el modelo.

Para el conjunto de datos C-twon se han eliminado las columnas ATT_FLAG y DATETIME que no son necesarias. No hay datos nulos ni duplicados, pero sí se han encontrado datos con el valor de 0 que presentan, por ejemplo, ausencia de caudal en bomba PU9 (F_PU9) y caudal en bomba PU11 (F_PU11).

Los conjuntos de datos BATADAL_dataset04 y BATADAL_test_dataset se encuentran altamente desbalanceados: de 4178 y 2090 instancias, solo 7 son ataques, lo que representa 0.167 % del total para el conjunto de datos de validación, mientras que para el de prueba son 0.334% de registros,

respectivamente.

Para una mejor comprensión de las correlaciones de las variables, se realizó un análisis visual del conjunto de datos BATADAL_dataset03 de entrenamiento. Se realiza un cálculo de correlación de estos datos, como se muestra en la figura 17. Los colores más claros representan menos correlación, mientras que los colores más oscuros representan más correlación. En algunas variables se visualiza una alta correlación, mientras que en otras no tanto. Como en los estados de las bombas, que son variables discretas, es decir, solo tienen dos valores: encendido (1) y apagado (0). Estas se muestran en color blanco, es decir, no hay correlación. También en algunas variables que contienen valores de cero en sus datos.

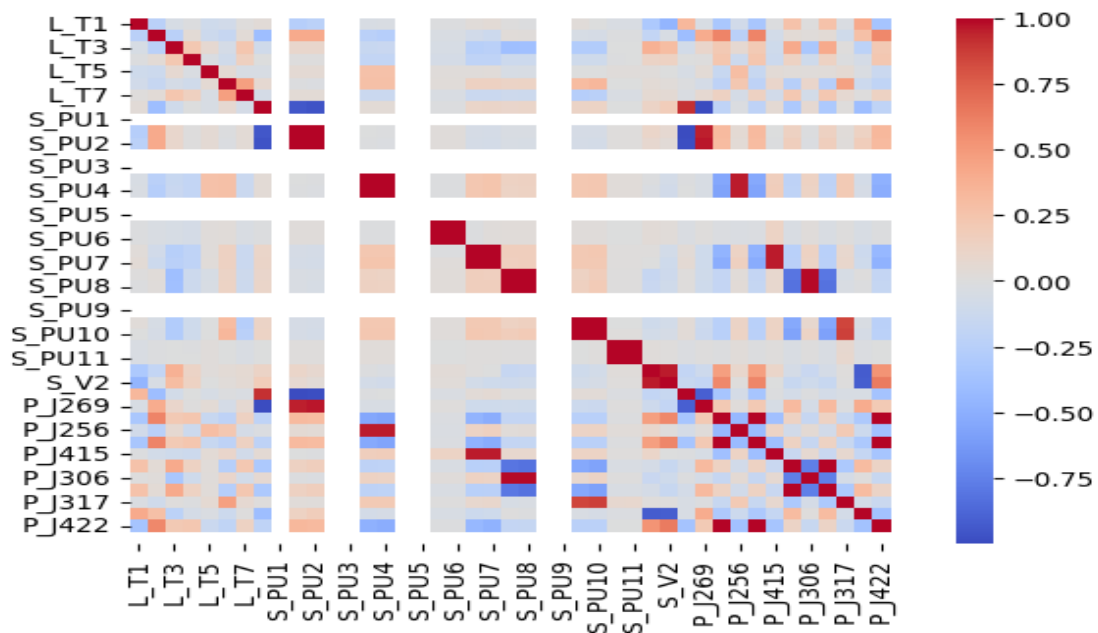


Figura 17. Correlación en BATADAL_dataset03.

4.5 Filtrado y Extracción de características

El filtrado y selección de características es un paso importante en el preprocesamiento de datos, especialmente cuando se trabaja con modelos no supervisados como los autoencoders. Al eliminar el ruido en los datos, no solo es beneficioso para el rendimiento cuando se entrena el algoritmo, sino que puede incluso ser positivo para aumentar la confianza y obtener mejores resultados.

Los motivos para usar la selección de características son principalmente: puede ayudar a reducir el tiempo dedicado al entrenamiento del proceso, reducir la complejidad del modelo y hacerlo más fácil de entender, aumentar la precisión (si se selecciona el subconjunto adecuado) y reducir el sobreentrenamiento.

Para nuestro estudio utilizaremos la plataforma de software destinada al aprendizaje automático y la minería de datos WEKA , usada para el análisis de conocimiento, versión 3.8.6 (c) 1999–2022 de la Universidad de Waikato, Hamilton, Nueva Zelanda. Con ello se decide qué atributos vamos a seleccionar y usamos la herramienta de ranking para la selección de atributos en dos métodos: la selección basada en correlación y el análisis de principales componentes.

La selección de atributos basada en correlación: este método evalúa los subconjuntos de atributos teniendo en cuenta que los subconjuntos de atributos buenos tienen características altamente correlacionadas con la clasificación, pero no están correlacionadas entre sí. En la figura 18 se muestra el resultado de la selección de atributos con el conjunto de datos de validación.

```

Evaluator: weka.attributeSelection.CorrelationAttributeEval
Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation: dataset04-weka.filters.unsupervised.attribute.Remove-R1,45-weka.filters.unsupervised.instance.RemoveDuplicates
Instances: 4177
Attributes: 43

```

Ranked attributes:					
0.99665	36	P_J289	-0.06246	18	F_PU6
0.99584	34	P_J300	-0.06893	19	S_PU6
0.64461	31	S_V2	-0.11087	5	L_T5
0.5812	2	L_T2	-0.13776	41	P_J317
0.54134	30	F_V2	-0.15358	39	P_J306
0.31916	10	F_PU2	-0.1647	23	S_PU8
0.30289	11	S_PU2	-0.16858	22	F_PU8
0.22628	3	L_T3	-0.18557	8	F_PU1
0.21109	4	L_T4	-0.20818	6	L_T6
0.18833	33	P_J269	-0.22935	27	S_PU10
0.08652	7	L_T7	-0.23609	26	F_PU10
0.07686	40	P_J307	-0.28383	32	P_J280
0.07497	38	P_J302	-0.34495	37	P_J415
0.00231	29	S_PU11	-0.35833	1	L_T1
0.00212	28	F_PU11	-0.37649	35	P_J256
0	16	F_PU5	-0.38858	42	P_J14
0	17	S_PU5	-0.39099	20	F_PU7
0	24	F_PU9	-0.4295	21	S_PU7
0	13	S_PU3	-0.49254	14	F_PU4
0	25	S_PU9	-0.51652	15	S_PU4
0	9	S_PU1			
0	12	F_PU3			

Figura 18. Selección de atributos del conjunto de datos de validación.

A continuación, se muestra en la figura 19 la correlación de la variable **P_J289** que es la presión en nudo J289 que está en el primer lugar del ranking.

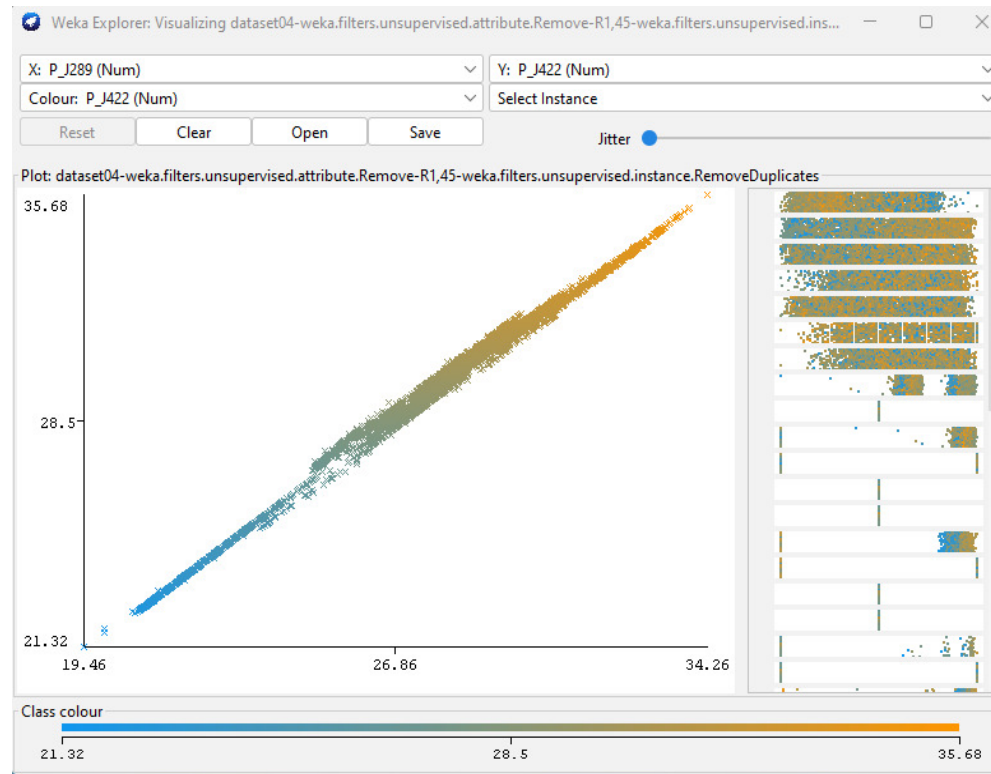


Figura 19. Correlación de la variable Presión en nudos P_J289.

En comparación con las figuras de correlación de las variables F_PU6 que es el caudal de Bomba PU6 y P_J256 que es la presión en nudo J256 que están en la parte baja del ranking. Como se visualiza en la figura 20 y 21 las correlaciones de estas variables están dispersas no están correlacionadas.

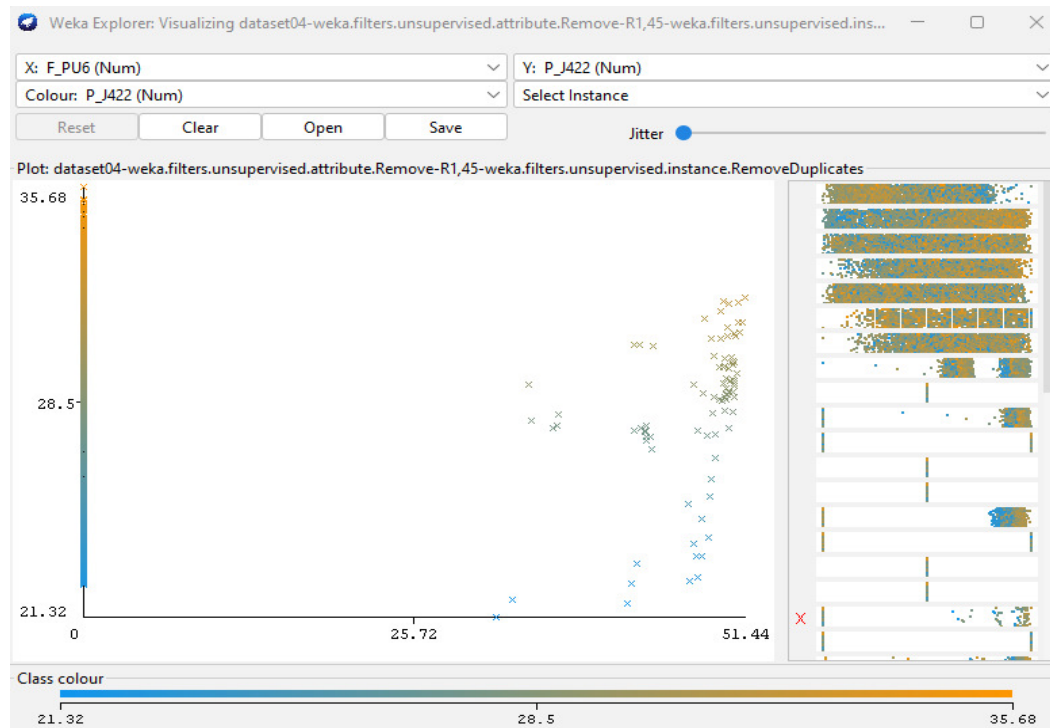


Figura 20. Correlación de la variable de Caudal F_PU6.

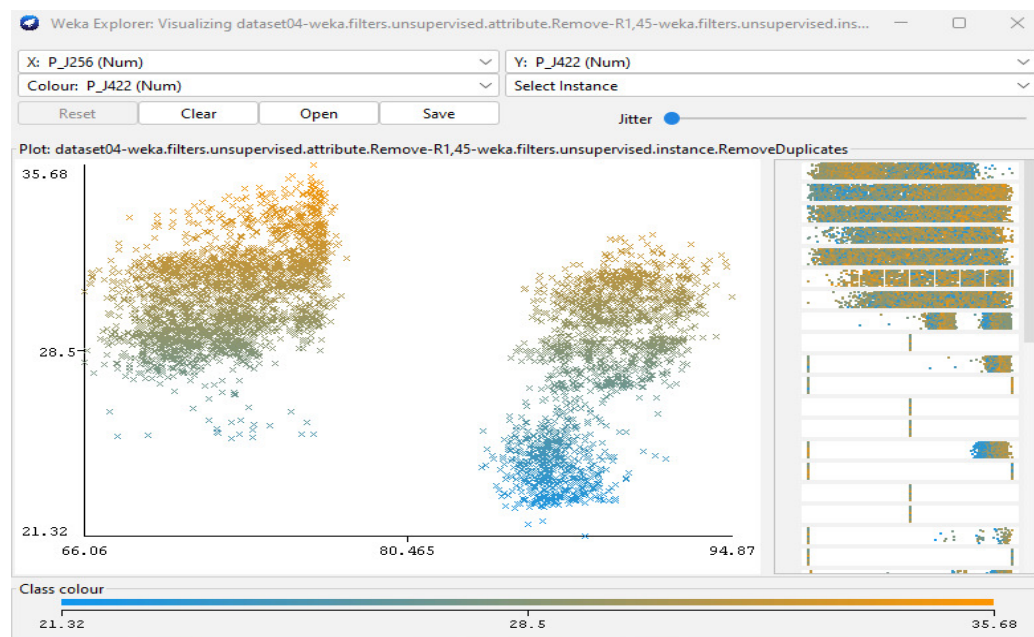


Figura 21. Correlación de la variable Presión en nudo P_J256.

Análisis de Componentes Principales: Principal Component Analysis (PCA), por sus siglas en inglés, es una técnica de reducción de dimensionalidad no supervisada que proyecta los datos en un nuevo espacio de características ortogonales ordenadas por varianza explicada. Puede usarse para seleccionar los componentes principales más importantes. La figura 22 muestra el ranking de atributos de este análisis de componentes principales, los cuales son 13.

```
Evaluator: weka.attributeSelection.PrincipalComponents -R 0.95 -A 5
Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation: dataset04-weka.filters.unsupervised.attribute.Remove-R1,45-weka.filters.unsupervised.instance.RemoveDuplicates
Instances: 4177
Attributes: 43
```

```
Ranked attributes:
0.7968 1 0.305 P_J300+0.304 P_J289+0.298 P_J422-0.211 S_PU4-0.208 F_PU4...
0.6649 2 0.358 P_J302+0.358 P_J307-0.279 S_V2+0.279 P_J14-0.265 S_PU8...
0.5354 3 0.398 P_J269-0.398 F_PU1+0.389 S_PU2+0.385 F_PU2-0.336 P_J280...
0.4388 4 -0.34 F_V2-0.314 P_J256+0.31 P_J14-0.293 S_V2+0.27 F_PU8...
0.3646 5 -0.437 F_PU7-0.422 S_PU7-0.343 P_J415+0.325 F_PU6+0.321 S_PU6...
0.295 6 0.382 P_J317+0.348 F_PU10+0.343 S_PU10-0.282 P_J256-0.256 F_PU4...
0.2392 7 -0.528 S_PU11-0.528 F_PU11+0.342 S_PU6+0.341 F_PU6+0.228 P_J415...
0.1864 8 -0.47 S_PU6-0.468 F_PU6-0.389 S_PU11-0.389 F_PU11-0.306 P_J415...
0.1436 9 0.657 L_T7+0.505 L_T6+0.241 L_T3+0.199 L_T4-0.189 L_T2...
0.1111 10 -0.484 L_T2+0.397 L_T3-0.334 L_T6-0.32 L_T7+0.309 L_T1...
0.0858 11 -0.848 L_T5+0.272 L_T4+0.211 L_T1+0.196 F_PU4+0.188 S_PU4...
0.0608 12 0.854 L_T4+0.324 L_T5+0.208 L_T2+0.186 P_J415-0.115 L_T6...
0.0411 13 -0.701 L_T1-0.316 L_T5-0.26 P_J280-0.212 L_T2-0.199 P_J422...

Selected attributes: 1,2,3,4,5,6,7,8,9,10,11,12,13 : 13
```

Figura 22. Ranking de atributos del análisis PCA.

4.6 Diseño del Modelo de detección

Aporte

El principal aporte de esta tesis es el diseño de un modelo de detección inteligente usando algoritmos y técnicas de Machine Learning que permita la detección de ataques cibernéticos en sistemas de distribución de agua SCADA de manera temprana. Para el diseño de este modelo se hace uso de un algoritmo de aprendizaje profundo llamado Autocodificador o Autoencoder (AE) y se modifica dicho algoritmo para un mejor rendimiento.

En el artículo propuesto por Taormina et al. (2018), los investigadores utilizan un algoritmo AE básico para detectar los ataques del conjunto de datos BATADAL. Tanto el codificador como el decodificador de la EA propuesta son dos ANN (redes neuronales artificiales profundas) totalmente conectadas.

En los resultados se muestra el rendimiento del modelo propuesto y se hace una comparación con el modelo que utilizó este algoritmo AE básico.

Autoencoders

Según M. Arif et al. (2020), en su libro “Avances en Deep Learning” mencionan que los autocodificadores (AEs), también conocidos como autoasociadores, están capacitados para aprender representaciones eficientes de los datos de entrada llamados codificaciones, sin ninguna supervisión (es decir, el conjunto de entrenamiento no está etiquetado). Los autocodificadores son útiles para reducir la dimensionalidad porque estas codificaciones suelen tener una dimensionalidad mucho menor que la de los datos de entrada. Y lo que es más significativo, los autocodificadores

son útiles para el preentrenamiento no supervisado de redes neuronales profundas porque son detectores de características poderosos.

Una de las principales ventajas de los autoencoders es que no existen restricciones sobre el tipo de relaciones que pueden aprender; por lo tanto, a diferencia del PCA, la reducción de dimensionalidad puede incluir relaciones no lineales. La principal desventaja radica en su alto riesgo de sobreentrenamiento (overfitting), por lo que se aconseja utilizar muy pocas épocas y evaluar constantemente la evolución del error con un conjunto de datos de validación.

Si se emplean funciones de activación lineales, las variables creadas en el cuello de botella (la capa con menos neuronas) son muy similares a las partes principales de un PCA, pero no necesariamente deben ser ortogonales entre sí.

El objetivo del proceso de entrenamiento es convertir el vector de entrada en un vector codificado de baja dimensión, con un error de reconstrucción mínimo al reconstruir los datos de entrada a partir del código correspondiente. En esencia, el proceso de codificación implica aprender características de los datos de entrada. AE extrae características útiles de cada vector de entrada y filtra la información no deseada.

La Figura 23 muestra un AE basado en ANN completamente conectado y sus componentes. Los datos del sistema de distribución de agua se introducen en el modelo AE, este clasifica las entradas como muestras normales o anómalas en función de un umbral.

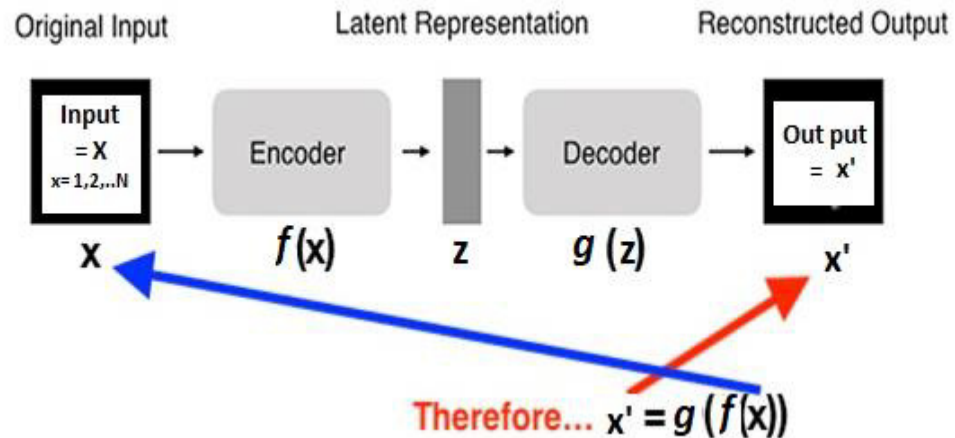


Figura 23. Arquitectura de Autoencoder.

Fuente. Datos tomados de Rosebrock (2020).

En donde:

X : Es el conjunto de datos de Entrada, desde la variable 1 hasta la variable **N**

Encoder: El codificador es el que recibe información y la comprime en el espacio latente. Si escribimos **x** para nuestros datos de entrada y **f** para el codificador, la representación del espacio latente de salida, **z**, sería **$z = f(x)$** .

Representación latente z: EL espacio latente es el espacio de dimensiones reducidas de los datos comprimidos o de entrada. La arquitectura de cuello de botella es otro nombre para el espacio latente, que contiene el número máximo de datos de compresión.

Decoder: Decodificador, la tarea del decodificador es aceptar la representación del espacio latente **z** y reconstruir la entrada inicial. Podemos representar el decodificador como **$x' = g(z)$** si representamos la

función del decodificador como $\mathbf{g}$ y la salida del detector como $\mathbf{x}'$.

$\mathbf{X}'$: Reconstructed output, es decir, son los datos de salida reconstruidos. Como se definió a $\mathbf{x}$ como datos de entrada, $\mathbf{f}(\mathbf{x})$ como la función codificadora y $\mathbf{g}(\mathbf{z})$ como la función decodificadora. Usando una notación matemática, todo el proceso de entrenamiento del autocodificador se puede escribir como lo siguiente: $\mathbf{x}' = \mathbf{g}(\mathbf{f}(\mathbf{x}))$.

Mientras que el codificador transforma las entradas en vectores de características, también conocidos como códigos, el decodificador reconstruye las entradas originales a partir de un espacio de características comprimido formado por representaciones codificadas.

El factor de compresión (cf) es la proporción entre el tamaño de la capa de entrada (o salida) y el de la capa oculta más intermedia que contiene las representaciones codificadas. Los valores de cf más altos están asociados con reconstrucciones más gruesas.

Cuando $\mathbf{X} \in \mathbf{x}$ es un patrón de entrada genérico alimentado a la red y $\mathbf{F} \in \mathbf{f}$ es el código que lo representa, matemáticamente se pueden escribir las transformaciones ϕ y ψ que realizan el codificador y el decodificador, respectivamente, como las ecuaciones (4-5) :

$$\phi : \mathcal{X} \rightarrow \mathcal{f} \quad (4)$$

$$\psi : \mathcal{f} \rightarrow \mathcal{X} \quad (5)$$

Mientras que la ecuación 6 representa la pérdida del AE que se entrena para que se pueda encontrar el conjunto de pesos de conexión que se

definen en ϕ y ψ que minimiza el error de reconstrucción que ha sido medido para todos los elementos de X en algunos patrones del entrenamiento en donde $\circ$ es el operador de composición de la función.

$$\phi, \psi = \arg \min_{\phi, \psi} \|X - (\psi \circ \phi)X\|^2 \quad (6)$$

AE para el modelo propuesto

Para mejorar el rendimiento de la detección de AE, aplicamos las restricciones de aseguramiento siguientes como: Pesos vinculados, Pesos ortogonales, Características no correlacionadas y Norma Unitaria.

1.- Pesos vinculados (Tied Weights)

Según Zhou et al. (2021) menciona que los pesos de las conexiones entre las neuronas del codificador (encoder) y el decodificador (decoder) pueden estar vinculados, esto significa que los pesos del decodificador son compartidos con los del codificador, es decir, las conexiones son simétricas.

La idea detrás de los pesos vinculados es que la representación latente aprendida por el codificador sea útil para la reconstrucción precisa en el decodificador, al compartir los pesos, se reduce el riesgo de sobreajuste y se mejora la generalización del modelo.

Los pesos vinculados aseguran que los pesos del codificador y del decodificador sean iguales. Esta restricción facilita el aprendizaje, particularmente la regularización y reducción de dimensionalidad que se asemejan a las de PCA.

Cuando los errores de reconstrucción son moderados, la codificación

genera características latentes ortogonales para los datos entregados siendo esto útil para la reducción de dimensionalidad y, de manera eventual para la detección de anomalías. La ponderación de los vectores de la capa I de un codificador y un decodificador en un AE multicapa se traslada en la siguiente ecuación (7) donde W son los pesos.

$$W_I = W_{-I}^T \quad (7)$$

2.- Pesos ortogonales

Los pesos de cada capa de codificación son ortogonales porque cada vector de peso es independiente. Las restricciones de ortogonalidad funcionan como una regularización de EA.

Cuando se aplica esta restricción, se penalizan los pesos no ortogonales. El usuario toma la decisión de elegir pesos ortogonales o no ortogonales; esto depende del conjunto de datos de entrada. De esta manera, la aplicación de esta restricción está supeditada a la regularización. La ecuación (8) representa la condición de ortogonalidad.

$$W_{encoder}^T W_{encoder} = I \quad (8)$$

3.- Características no correlacionadas

Las representaciones latentes deben estar no correlacionadas si la salida del codificador es ortogonal. La ecuación matemática (9) muestra la salida del AE en características no correlacionadas.

$$correlation(O_{encoder\ i}, O_{encoder\ j}) = 0 | i \neq j \quad (9)$$

4.- Norma Unitaria

Es necesario establecer normas unitarias para los pesos de cada capa. Esta característica ayuda en el manejo de los gradientes de explosión y desaparición. Todas las capas de EA deben estar equipadas con la restricción de la norma unitaria como se muestra en la ecuación (10).

$$\sum_{j=1}^p w_{ij}^2 = 1; i = 1, \dots, k \quad (10)$$

Algoritmos de Optimización

En Berahmand et al. (2024) estos algoritmos son métodos de optimización que se utilizan para actualizar los pesos de las redes neuronales, como los autoencoders, durante el entrenamiento. Estos algoritmos ajustan los pesos y sesgos de la red para entrenar el autocodificador de manera efectiva. Existen varias técnicas de optimización para entrenar autocodificadores, siendo las más notables el (SGD) (Stochastic Gradient Descent) descenso de gradiente estocástico, Adam y Adagrad.

En nuestro caso para el modelo propuesto se ha utilizado la técnica de optimización Adam debido a las ventajas que muestran diversos estudios:

- **Ajustes automáticos de la tasa de aprendizaje:** Adam ajusta la tasa para cada parámetro individualmente, basándose en estimaciones del primer y segundo momento de los gradientes.
- **Eficiencia computacional:** Requiere poca memoria y es computacionalmente eficiente.

- **Apto para datos y parámetros grandes:** Funciona bien con grandes volúmenes de datos y modelos con muchos parámetros.
- **Desempeño consistente:** Generalmente, Adam tiene un buen desempeño en la práctica y converge rápidamente.

Selección de Hiperparámetros

Un método para encontrar los hiperparámetros ideales para un modelo de aprendizaje automático es la búsqueda aleatoria de hiperparámetros o random search por sus siglas en inglés. La búsqueda aleatoria evalúa el rendimiento del modelo seleccionando combinaciones al azar dentro de un rango predefinido en lugar de probar todas las combinaciones posibles de hiperparámetros. En el cuadro 8 se muestran los hiperparámetros seleccionados.

Cuadro 8. Selección de Hiperparámetros

Hiperparámetros	Búsqueda aleatoria	Hiperparámetro Seleccionado
Tasa de aprendizaje del optimizador Adam Adam Optimizer learning rate	0.0001, 0.001, 0.01, 0.1	0.0001
Tamaño del lote Batch size	8, 16, 32, 64, 128, 264	32
Número de épocas Number of epochs	500, 1000, 2500, 5000	500
Número de capas ocultas Number of hidden layers	3, 5, 7, 9, 11	5

Establecer el Umbral

Después de entrenar el modelo, el siguiente paso es establecer un umbral de error entre la entrada y la salida del algoritmo. Se elige el umbral theta (θ_{th}) de manera empírica y la longitud de la ventana n. Para lograrlo, se observa y suma el rango de errores de reconstrucción promedio entre todas las características de una muestra. Establecemos el umbral en función de la estimación final del rango.

El umbral theta está relacionado con varios percentiles de distribución de errores en el conjunto de datos de validación. El modelo clasifica como bajo ataque cuando el autocodificador emite consecutivamente un error de reconstrucción mayor que theta en una longitud de ventana de n horas. A continuación, se muestra la ecuación matemática del umbral theta.

$$\theta_{th} = \max \left\{ f(x) : \sum_m \frac{|x - x'|}{m} \text{ for } N_{\text{training-samples}} \right\} \quad (11)$$

Para probar el modelo en nuevas muestras, el proceso de calibración es crucial. Si un objeto de prueba se clasifica como "ATTACK", localizamos las entidades relacionadas con las características atacadas, como bombas, sensores y válvulas, utilizando errores de reconstrucción.

Estructura del algoritmo

A continuación, en el cuadro 9 se visualiza el algoritmo para la detección de ataques.

Cuadro 9. Estructura del algoritmo

Entrenamiento y pruebas del algoritmo
Entradas: Conjunto de datos (X), argumentos del modelo que incluyen nl, nh, cf, activación y capas personalizadas. Ejecutar: detección de ataques ocultos
1.- Inicializar el AE con el conjunto de datos X ya preprocesado 2.- Aplicar restricciones en el codificador y capas de decodificador que utilizan una combinación de pesos vinculados (Ecuación 07), pesos ortogonales (Ecuación 08), características no correlacionadas (Ecuación 09), norma unitaria (Ecuación 10). 3.- Para $k = 1, 2, 3, 4, \dots, N$ hasta el entrenamiento N hacer el entrenamiento. 4.- Pasar la muestra a través de las capas de codificación, representación latente y decodificación (Ecuaciones 4,5,6). 5.- Utilizar Adam Optimizer para minimizar la función de pérdida. 6.- Retornar los parámetros de AE. 7.- Seleccionar el umbral θ_{th}. 8.- Para $k = 1, 2, 3, 4, \dots, N$ hasta la prueba N hacer la prueba. 9.- Pasar la muestra a través de las capas de codificación, representación latente y decodificación (Ecuaciones 4,5,6). 10.- Retornar los errores de reconstrucción 11.- Aplicar el umbral θ_{th} en los errores de reconstrucción para la detección de ataques o anomalías.

Las ecuaciones (07-10) representan las restricciones de aseguramiento de la IA, incluidas las restricciones de pesos vinculados, pesos ortogonales,

características no correlacionadas y normas de unidad, que se aplican al AE. Se ha seleccionado la combinación de estas restricciones y se aplican a las capas ocultas. El objetivo es obtener una representación latente significativa y no correlacionada, un requisito previo para la reducción de la dimensionalidad.

Se usó Adam Optimizer con una tasa de aprendizaje = 0.0001, un factor de decaimiento de 0.5 (que significa que la tasa de aprendizaje disminuirá a la mitad después de cada actualización de parámetros) y momentos (β_1 y β_2) (0.9, 0.99), siendo estos coeficientes los que controlan cómo se calculan las medias móviles exponenciales de los gradientes y sus cuadrados. β_1 generalmente se establece en 0.9 y β_2 en 0.99. Estos valores influyen en la suavidad de las actualizaciones y en la adaptación de la tasa de aprendizaje.

Además, se seleccionaron 500 épocas, lo cual significa que el modelo se ha entrenado durante 500 iteraciones completas a través de los datos de entrenamiento, con un “minibatch size” (tamaño del minibatch) de 32. Un minibatch es un subconjunto de los datos de entrenamiento. En este caso, se está utilizando un tamaño de minibatch de 32, lo que significa que, en cada paso de actualización, el modelo se ajusta utilizando 32 ejemplos de entrenamiento a la vez.

Para los modelos base AE y el Modelo Propuesto, se han elegido de manera aleatoria los hiperparámetros ideales y se han maximizado las puntuaciones de rendimiento de la clasificación binaria, incluidas la precisión (precision), la recuperación (recall), la puntuación F1 (F1 score), la exactitud (accuracy) y la especificidad (specificity).

Entrenamiento del modelo

Primero se define los parámetros del modelo

```
# define model parameters
params = {
    'nI' : X.shape[1],
    'nH' : 3,
    'cf' : 2.5,
    'activation' : 'tanh',
    'verbose' : 1,
}
```

Figura 24. Definición de Parámetros.

Cuadro 10. Parámetros del modelo

```
nI = self.params['nI'] # número de entradas

nH = self.params['nH'] # número de capas ocultas en codificador (decodificador)

cf = self.params['cf'] # factor de comprensión

activation = self.params['activation'] # función de activación de AE

optimizer = self.params['optimizer'] # Optimizador Keras

verbose = self.params['verbose'] #echo on screen
```

En el cuadro 10, se detalla cada uno de los parámetros para el entrenamiento del modelo para su mejor comprensión.

El Modelo Propuesto difiere del AE de referencia o base en el diseño de la definición de la capa oculta. Se aplica “early stopping” con “patience” = 3 para una mejor regularización, early stopping o parada precoz es una técnica que te permite especificar un número grande de épocas de entrenamiento de forma arbitraria y detener el entrenamiento una vez que el rendimiento del modelo deja de mejorar en un conjunto de validación, y el valor de patience (paciencia) determina cuántas épocas esperar sin mejoras antes de detener el entrenamiento, en nuestro caso hay que esperar 3 épocas. En la figura 25 se visualiza esta parte del modelo.

```
# train models with early stopping and reduction of learning rate on plateau
earlyStopping= EarlyStopping(monitor='val_loss', patience=3, verbose=0, min_delta=1e-4, mode='auto')
lr_reduced = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=1, verbose=0, epsilon=1e-4, mode='min')
```

Figura 25. Entrenamiento del modelo con early stopping.

Se crea el autoencoder para eventos de detección con los parámetros ya descritos líneas arriba, como se muestra en la figura 26.

```
# create AutoEncoder for Event Detection (AEED)
autoencoder = AEED(**params)
autoencoder.initialize()
```

Figura 26. Crear Autoencoder para detección.

Se obtuvo el número de neuronas en cada capa de la siguiente manera: capas codificadoras: [layer_0, layer_1, layer_2] = [43, 34, 25]; Capa de cuello de botella: [layer_3] = [17]; y capa decodificadora como: [layer_4, layer_5, layer_6] = [25, 34, 43]. A continuación, se muestra este resultado en la figura 27.

```
Created autoencoder with structure:
layer_0: 43, layer_1: 34, layer_2: 25, layer_3: 17, layer_4: 25, layer_5: 34, layer_6: 43
```

Figura 27. Resultado de la creación de autoencoder.

CAPÍTULO 5: RESULTADOS Y DISCUSIÓN

5.1. Presentación de resultados

Al aplicar restricciones de aseguramiento de pesos vinculados, pesos ortogonales, características no correlacionadas y norma unitaria en el modelo propuesto, no todas las cuatro restricciones funcionan de manera ideal, pero una combinación de estas restricciones mejora la clasificación y reduce la dimensionalidad.

En los siguientes gráficos se muestran los resultados de la detección de ataques del modelo propuesto en los conjuntos de datos de validación y de prueba, los cuales contienen 7 diferentes ataques cada uno.

Como se visualiza en la figura 28, el modelo produce 5 conjuntos de falsas alarmas.

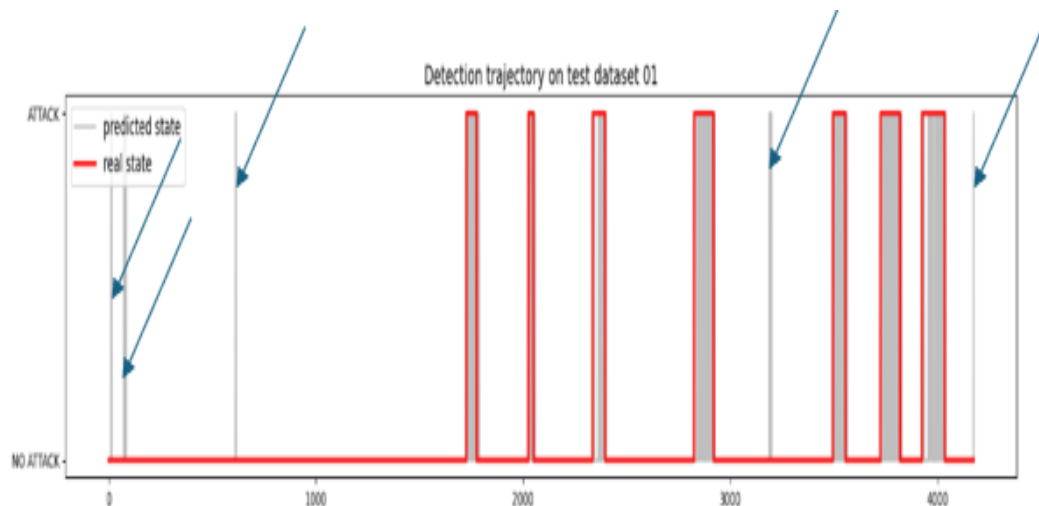


Figura 28. Detección en el dataset test de validación.

Mientras que para el conjunto de datos de prueba que contiene 7 ataques no etiquetados solo se muestra una sola falsa alarma como se visualiza en la figura 29 .

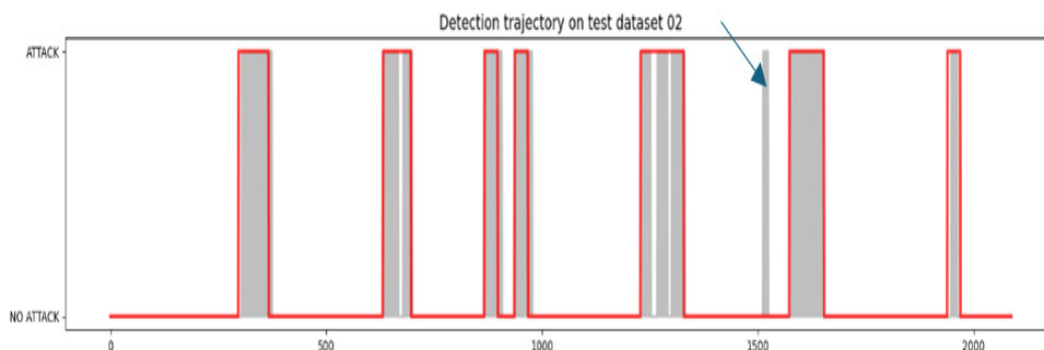


Figura 29. Detección en el dataset test de prueba.

5.1.1. Métricas de rendimiento del modelo

Para probar las hipótesis planteadas en este trabajo de estudio se hace uso de las siguientes métricas como los son: la puntuación de tiempo de detección (STTD) para evaluar que el modelo tenga la capacidad de detectar un ataque en el menor tiempo posible, la exactitud (Accuracy) para evaluar cuantas predicciones correctas se ha realizado de todas las predicciones realizadas, la puntuación F1 (F1 score) que evalúa cuantas mejores predicciones se ha obtenido y por último la precisión (Precision) que evalúa las predicciones positivas cuanto menor sea las falsas alarmas es mejor.

Hipótesis General

El diseño de un modelo de detección inteligente basado en algoritmos y técnicas de Machine Learning permite detectar tempranamente ataques cibernéticos en sistemas SCADA de distribución de agua, con un desempeño óptimo en comparación con modelos autoencoder o no supervisados.

Se logró diseñar el modelo de detección usando el algoritmo Autoencoder, junto con restricciones de aseguramiento, pesos vinculados, pesos ortogonales, características no correlacionadas y norma unitaria, además de las métricas ya mencionadas líneas arriba. Los resultados evidencian que el modelo permite la detección temprana y es eficaz para la detección de ataques cibernéticos.

Hipótesis Específicas

HE1: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el menor tiempo posible en comparación con estudios basados en modelos autoencoder o modelos no supervisados.

Se diseñó un modelo de detección que permitió alcanzar el menor tiempo posible; el modelo alcanzó una puntuación de tiempo de detección (STTD) del 94%, es decir, es bastante significativo en términos porcentuales. Por lo tanto, se acepta la hipótesis planteada.

HE2: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mayor exactitud en comparación con estudios basados en modelos autoencoder o modelos no supervisados.

Se diseñó un modelo de detección que permite una mayor exactitud de las

predicciones correctas que ha realizado el modelo. De todas las predicciones realizadas, se ha obtenido un rendimiento de exactitud (Accuracy) del 97% para el conjunto de datos de validación, mientras que para el conjunto de datos de prueba se obtuvo un Accuracy del 96%. Esto significa que el modelo es fiable en lo que respecta a realizar predicciones correctas. Por lo tanto, se acepta la hipótesis planteada.

HE3: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con la mejor puntuación F1, es decir, con mejores predicciones de detección de ataques en comparación con estudios basados en modelos autoencoder o modelos no supervisados.

Se diseñó un modelo de detección con una puntuación F1 que está relacionada con las mejores predicciones. Se obtuvo una puntuación F1=83% para el conjunto de datos de prueba, mientras que para el conjunto de datos de validación se obtuvo una puntuación F1=98%. Se concluye que un 83% en un rendimiento aceptable. Por lo tanto, se acepta la hipótesis planteada.

HE4: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor precisión en comparación con estudios basados en modelos autoencoder o modelos no supervisados.

El diseño de un modelo de detección con un algoritmo de mayor precisión respecto a las predicciones positivas: el modelo alcanzó una puntuación de precisión del 100% trabajando con el conjunto de datos de validación, mientras que para los datos de prueba logró una puntuación de precisión del 83%. Por lo tanto, no se acepta la hipótesis planteada.

Se muestra el resultado en el cuadro 11 de las muestras obtenidas tanto del conjunto de datos de validación como del de prueba y se realiza una comparación con el modelo que utilizó el algoritmo autoencoder básico de los autores (Taormina & Galelli, 2018).

Cuadro 11. Comparación del modelo propuesto con el modelo AE de referencia

		Accuracy	F1 score	Precision	Recall
Taormina & Galelli	Dataset Validación	N/A	0.704	0.835	0.608
	Dataset Prueba	N/A	0.715	0.881	0.602
Modelo Propuesto	Dataset Validación	0.977	0.988	1	0.977
	Dataset Prueba	0.960	0.831	0.838	0.825

Analizando el dataset de prueba en donde el modelo que utilizó AE básico tiene una precisión de 0.881 trabajando con datos de prueba etiquetados, mientras que el modelo propuesto tiene una precisión de 0.838 trabajando con datos de prueba no etiquetados.

HE5: El diseño de un modelo de detección inteligente permite detectar ataques cibernéticos en sistemas SCADA de distribución de agua, con el algoritmo de mayor exhaustividad en comparación con estudios basados en modelos autoencoder o modelos no supervisados.

Se diseñó un modelo de detección con un algoritmo de mayor exhaustividad relacionada con las predicciones positivas reales que se clasificaron correctamente como predicciones positivas. Si analizamos el Recall del

modelo base, se tiene 0.602, a diferencia del modelo propuesto, que es de 0.825. Aquí se tiene una diferencia de 0.223; en términos de porcentaje, es un 22% más de predicciones positivas reales que ha realizado el modelo propuesto. Por lo tanto, se acepta la hipótesis planteada.

A continuación, se analizarán las métricas junto con otros estudios del marco teórico que hayan realizado su investigación con el conjunto de datos C-town.

5.1.2. Tiempo de detección de los ataques

Según Taormina & Galelli (2018) Time-to-Detection Score s_{TTD} o tiempo hasta la detección es el tiempo que necesita un algoritmo para revelar un ataque cuanto menor sea ese valor mejor será el rendimiento del algoritmo, se muestra la ecuación que se formula del s_{TTD} .

$$0 \leq s_{TTD} = t_d - t_0 \leq \Delta t \quad (12)$$

En donde:

s_{TTD} : Es la puntuación de tiempo hasta la detección de un ataque.

t_d : Es el tiempo en el que se detecta el ataque y corresponde a la primera marca de tiempo marcada como bajo ataque mientras el ataque está en curso.

t_0 : Es el tiempo en la que comienza el ataque.

Δt : Duración total del ataque.

En la batalla de algoritmos (BATADAL) al comparar todos los algoritmos en diferentes escenarios de ataque, se calculó la siguiente puntuación de rendimiento (s_{TTD}):

$$s_{TTD} = 1 - \frac{1}{n_a} \sum_i^{n_a} \frac{TTD_i}{\Delta t_i} \quad (13)$$

En donde:

n_a : Número de ataques contenidos en un dataset.

TTD_i : Tiempo hasta la detección del i-ésimo ataque.

Δt_i : Duración correspondiente.

El valor de s_{TTD} varía entre 0 y 1, siendo $s_{TTD} = 1$ el caso ideal en el que todos los ataques se detectan inmediatamente y $s_{TTD} = 0$ siendo el caso en el que no se detecta ninguno de los ataques.

En nuestro caso usaremos la ecuación 13, en el cuadro 12 muestra los datos usados para hallar la métrica s_{TTD} .

En donde:

n_a : 7 ataques

TTD_i : Se obtuvo de los resultados del modelo.

Δt_i : Obtenido del Cuadro 5 Ataques del conjunto de datos BATADAL_test_dataset de la columna duración de los ataques.

Cuadro 12. Valores para el cálculo de la métrica s_{TTD}

	Δt_i	TTD_i	$\frac{TTD_i}{\Delta t_i}$
Ataque 1	70	3	0.042857143
Ataque 2	65	1	0.015384615
Ataque 3	31	0	0
Ataque 4	31	0	0
Ataque 5	100	2	0.02
Ataque 6	80	2	0.025
Ataque 7	30	8	0.266666667
		Σ	0.36990842

$$s_{TTD} = 1 - 0.36990842 / 7 = \mathbf{0.94715594}$$

5.2. Discusión

A continuación, se realizará una comparación de los resultados obtenidos de las métricas con los resultados que han obtenido los otros estudios que utilizaron la misma métrica. Cabe mencionar que estos estudios en su gran mayoría han trabajado con conjuntos de datos de agua como C-town, E-town y Batadal.

5.2.1. Métrica s_{TTD}

En el cuadro 13 se muestra la comparación de la métrica s_{TTD} obtenida con

los valores de otros estudios.

Cuadro 13. Comparación de la métrica s_{TTD} con otros estudios

Modelo	Descripción	STTD
OCDS Kadosh et al. (2020)	One-Class Cyber-Attack Detection System	0.967
ANN Abokifa et al. (2019)	Artificial Neural Networks	0.963
QDA/LDA Ramotsoela et al. (2019)	Multistage Model	0.958
SVDD Housh et al. (2022)	Support Vector Data Description	0.956
Mod_Propuesto	Autoencoder con Restricciones	0.947
CNN Ramotsoela et al. (2020)	Convolutional Neural Network	0.946
AEWMA Rodríguez-M. et al. (2023)	Autoencoder	0.945
OS ELM Choi et al. (2020)	Online Sequential ELM	0.940
TGCN Tsiami et al. (2021)	Temporal Graph Convolutional Neural Networks	0.934
FastICA Brentan et al. (2021)	Fast Independent Component Analysis	0.913
iForest Mahmoud et al. (2022)	Isolation Forest	0.860

Entre en el s_{TTD} más alto con 0.967 y del modelo propuesto de 0.947 existe una diferencia de 0.02 no es un porcentaje considerable teniendo en cuenta que los otros modelos como de Kadosh et al. (2020) y Ramotsoela et al. (2019) son modelos o métodos de machine learning supervisados a diferencia del modelo propuesto que es no supervisado.

5.2.2. Exactitud del modelo

Para esta métrica de accuracy o exactitud se ha trabajado con los estudios que reportan estos valores y con el conjunto de datos Batadal para Choi et al. y Mahmoud et al. También la investigación de Selim et al. que utilizó una

base de datos de Infraestructura Crítica de agua. A continuación, se muestra esta comparación en el siguiente cuadro 14.

Cuadro 14. Comparación de la métrica Accuracy con otros estudios

Modelo	Descripción	Accuracy
Mod_Propuesto	Autoencoder con Restricciones	0.960
iForest Mahmoud et al. (2022)	Isolation Forest	0.948
CART Selim et al. (2021)	Classification And Regression Tree	0.940
OS ELM Choi et al. (2020)	Online Sequential ELM	0.924

Como se muestra en el cuadro 14 el modelo propuesto tiene un 96% de rendimiento de Accuracy ocupando el primer lugar como también se muestra para una mejor visualización en la figura 30.

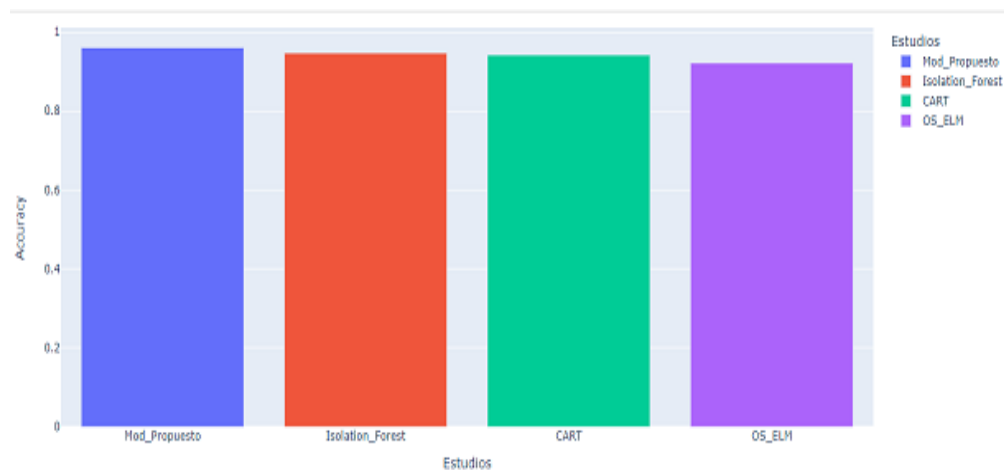


Figura 30. Rendimiento de la métrica Accuracy por estudio.

5.2.3. Puntuación F1

Para esta métrica, F1 Score o Puntuación F1 se han utilizado los estudios que reportan estos valores, habiendo trabajado solo con el conjunto de datos C-town. Se muestra además el rendimiento tanto en el conjunto de

datos de validación como en el de prueba.

A continuación, se muestra esta comparación en el siguiente cuadro 15.

Cuadro 15. Comparación de Puntuación F1 con otros estudios

Modelo	Descripción	F1 Score	
		Validación	Prueba
SVDD Housh et al. (2022)	Support Vector Data Description	0.855	0.884
Mod_Propuesto	Autoenconder con Restricciones	0.988	0.831
Taormina & Galelli (2018)	Autoencoder Base	0.704	0.715

La métrica puntuación F1 en el dataset de validación a diferencia de Housh et al. con un porcentaje de 85% contra un 98% del modelo propuesto es de 13% mientras que el dataset de prueba con 83% existe una diferencia de 5% menos con respecto a 88%. En tanto para la comparación de esta métrica con el estudio de Taormina & Galelli que usa el algoritmo Autoenconder la diferencia con el del modelo propuesto es de 12% con respecto al dataset de prueba.

5.2.4. Precisión del modelo

En lo que respecta a la precisión, se presenta a continuación el cuadro 16 con los estudios que utilizaron esta métrica. Ambos estudios, tanto el de Kravchik et al. como el de Mahmoud et al., los realizaron con el conjunto de datos Batadal. Tanto el dataset C-town como Batadal son datasets de redes de distribución de agua.

Cuadro 16. Comparación métrica Precisión con otros estudios

Artículos	Descripción	Precisión
UAE Kravchik et al., 2022	Autoencoders Undercomplete	0.957
iForest Mahmoud et al., 2022	Isolation Forest	0.840
Mod_Propuesto	Autoenconder con Restricciones	0.839

Si bien el valor de precisión del modelo propuesto es de 12% menos que el del estudio de Kravchik et al. con la precisión más alta con un 95% de precisión. El modelo con mayor rendimiento utiliza autoencoders undercomplete que son una de las arquitecturas más simples para autoencoders.

Los autoencoders undercomplete tienen una capa oculta pequeña en comparación con la capa de entrada; esto puede ser una desventaja ya que pueden aprender representaciones de baja calidad y la capacidad de representación se puede limitar, lo que puede dificultar la reconstrucción precisa de los datos originales. Con los otros modelos hay una diferencia baja y se tiene que mencionar que para el modelo propuesto de los tres conjuntos de datos de C-town se ha utilizado el dataset de prueba con data de ataques no etiquetados que es más real en un sistema SCADA de distribución de agua.

5.2.5. Exhaustividad del modelo

En lo que respecta a la exhaustividad del modelo, se presenta a continuación el cuadro 17 con los estudios que utilizaron esta métrica. Ambos estudios, tanto el de Mahmoud et al. como el de Kravchik et al., lo realizaron con el conjunto de datos Batadal.

Cuadro 17. Comparación métrica exhaustividad con otros estudios

Artículos	Descripción	Exhaustividad
Mod_Propuesto	Autoencoders con restricciones	0.825
VAE Kravchik et al., 2022	Autoencoder Variacional Estándar	0.762
XGBoost Mahmoud et al., 2022	Extreme Gradient Boosting	0.650

Como se muestra en el cuadro 17 el modelo propuesto tiene un 82% de rendimiento de Exhaustividad ocupando el primer lugar, a diferencia de Kravchik et al. que alcanzo un 76% con el modelo Autoencoder Variacional Estándar.

CONCLUSIONES

1. Con respecto a la detección de ataques cibernéticos en sistemas de distribución de agua SCADA, con el menor tiempo de detección posible, se logró el objetivo y se verifica que, en comparación con otros modelos autoencoders o no supervisados, se obtuvo una puntuación de tiempo de detección (STTD) = 0.947, con una diferencia de 0.02 respecto al valor más alto obtenido en otro estudio que utilizó un modelo supervisado.
2. En relación con el objetivo de detectar ataques cibernéticos en sistemas de distribución de agua SCADA con la mayor exactitud (accuracy), la cual nos brinda el porcentaje de acierto del modelo, se alcanzó la mejor performance respecto a otros modelos autoencoders o no supervisados, con un accuracy del 96%. La comparación se realizó con los modelos que utilizaron el conjunto de datos C-town.
3. En relación con el objetivo de detección de ataques cibernéticos en sistemas de distribución de agua SCADA, con la mejor puntuación F1, es decir, con mejores predicciones de detección, se obtuvo una mejor eficacia, alcanzando un 12% más de rendimiento respecto al modelo base, mientras que el que utilizó un modelo supervisado es casi similar en rendimiento. La métrica de puntuación F1 considera tanto la precisión como la recuperación.
4. Con respecto al objetivo de detección de ataques cibernéticos con el algoritmo de mejor precisión, se obtuvo una precisión del 84% no logrando el objetivo propuesto. Esta se efectuó con el conjunto de datos de prueba que no tiene datos etiquetados. En comparación con el modelo de mayor rendimiento, que utiliza autoencoders undercomplete

y obtuvo un 96%, el cual usó datos de prueba etiquetados, además, el algoritmo usado tiene una capa oculta pequeña en comparación con la capa de entrada del modelo propuesto. Esto puede ser una desventaja, ya que pueden aprender representaciones de baja calidad y la capacidad de representación se puede limitar.

5. En relación con el objetivo de detección de ataques cibernéticos en sistemas de distribución de agua SCADA, con la mejor métrica de recall o exhaustividad, es decir, con mejores predicciones positivas reales de detección en comparación con el modelo de referencia que utilizó autoencoders, se obtuvo un buen rendimiento del modelo; se alcanzó un 22% más de predicciones positivas reales.
6. El análisis de los trabajos relacionados mostró que el enfoque de la investigación se ha desplazado hacia la aplicación de aprendizaje automático profundo para la detección de ataques cibernéticos; es por ese motivo que se eligió Autoencoders, la cual, aplicando las cuatro restricciones de aseguramiento, mejora la detección del algoritmo.
7. El uso de métodos de aprendizaje automático requiere conjuntos de datos de alta calidad, lo que significa que son relevantes para el dominio, significativos y confiables. En nuestro caso, se seleccionó C-town, un conjunto de datos de tamaño mediano que ha sido utilizado en diversos estudios de investigación, sobre todo cuando se trata de un aprendizaje automático no supervisado en el cual se entrena y se prueba el algoritmo con un conjunto de datos no etiquetados.

RECOMENDACIONES

1. Se recomienda el uso de otros conjuntos de datos como SWAT y WADI que poseen un mayor número de instancias y cumplen con los parámetros de calidad que caracterizan a estos sistemas SCADA para mejorar el rendimiento de las métricas.
2. Se recomienda utilizar el aprendizaje automático no supervisado, ya que es necesario desarrollar modelos que sean capaces de aprender el comportamiento normal de los sistemas SCADA y detectar cualquier variación o anomalía que pueda surgir, proporcionando así una mayor capacidad de detección y protección frente a posibles ataques cibernéticos.
3. En Lima, Perú, la entidad encargada de la administración de sistemas SCADA de distribución de agua de la ciudad es SEDAPAL (Servicio de Agua Potable y Alcantarillado de Lima). Se recomienda la aplicación de este modelo propuesto con datos reales de los sistemas SCADA de distribución de agua de esta entidad, dados los resultados obtenidos en el presente estudio, que mejoraría el tiempo de detección, la exactitud, las predicciones correctas y las predicciones positivas reales.
4. El propósito de un sistema SCADA de distribución es suministrar la suficiente cantidad de agua con la mejor calidad, todo el tiempo y llegar a todos los lugares de una ciudad. En este estudio no se han considerado esos criterios de calidad; se recomienda analizar los factores que condicionan el deterioro de la calidad del agua, considerarlos para una mejor eficiencia y precisión del algoritmo de detección.

5. Es recomendable el desarrollo de un prototipo o sistema funcional con la finalidad de prevenir y estar preparados ante ciberataques que puedan ocurrir en este tipo de sistemas SCADA de distribución de agua.
6. En cuanto a la ciberseguridad, primordialmente en sistemas de control industrial, la academia recomienda monitorear, revisar y analizar el marco legal y normativo vinculado a la ciberseguridad en cada país, proponer un plan de actualización que garantice un ecosistema nacional seguro, frente al contexto de nuevas amenazas y riesgos cibernéticos producto de las tecnologías emergentes.

REFERENCIAS BIBLIOGRÁFICAS

- Abokifa, A. A., Haddad, K., Lo, C., & Biswas, P. (2019). Real-Time Identification of Cyber-Physical Attacks on Water Distribution Systems via Machine Learning–Based Anomaly Detection Techniques. *Journal of Water Resources Planning and Management*, 145(1), 04018089.
[https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0001023](https://doi.org/10.1061/(ASCE)WR.1943-5452.0001023)
- Addeen, H. H., Xiao, Y., Li, J., & Guizani, M. (2021). A Survey of Cyber-Physical Attacks and Detection Methods in Smart Water Distribution Systems. *IEEE Access*, 9, 99905-99921.
<https://doi.org/10.1109/ACCESS.2021.3095713>
- Addeen, H. H., Xiao, Y., & Li, T. (2024). A CVAE-Based Anomaly Detection Algorithm for Cyber Physical Attacks for Water Distribution Systems. *IEEE Access*, 12, 48321-48334.
<https://doi.org/10.1109/ACCESS.2024.3384295>
- Ahmed, C. M., Palleti, V. R., & Mishra, V. K. (2022). A practical physical watermarking approach to detect replay attacks in a CPS. *Journal of Process Control*.
- Almajed, R., Ibrahim, A., Zaid Abualkishik, A., Mourad, N., & A Almansour, F. (2022). Using machine learning algorithm for detection of cyber-attacks in cyber physical systems. *Periodicals of Engineering and Natural Sciences (PEN)*, 10(3), 261. <https://doi.org/10.21533/pen.v10i3.3035>
- Barrios Arce, Juan Ignacio. (2019a). *Inteligencia Artificial y salud. Un caso práctico*.
<https://www.juanbarrios.com/inteligencia-artificial-y-salud-caso-practico/>
- Barrios Arce, Juan Ignacio. (2019b). *La matriz de confusión y sus métricas*.
<https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>
- Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y., & Xu, Y. (2024). Autoencoders and their applications in machine learning: A survey. *Artificial Intelligence Review*, 57(2), 28.
<https://doi.org/10.1007/s10462-023-10662-6>

- Brentan, B., Rezende, P., Barros, D., Meirelles, G., Luvizotto, E., & Izquierdo, J. (2021). Cyber-Attack Detection in Water Distribution Systems Based on Blind Sources Separation Technique. *Water*, 13(6), 795. <https://doi.org/10.3390/w13060795>
- Cancino, H. (2022, mayo 11). *Conti, el ciberataque que puso en aprietos a Costa Rica y amenaza con expandirse a los gobiernos de A. Latina*. <https://www.americaeconomia.com/conti-el-ciberataque-que-puso-en-aprietos-costa-rica-y-amenaza-con-expandirse-los-gobiernos-de>
- Centro Nacional de Seguridad Digital. (2022, mayo 12). *Alerta Integrada de Seguridad Digital*. www.gob.pe/cnsd alertas@cnsd.gob.pe
- Choi, Y. H., Sadollah, A., & Kim, J. H. (2020). Improvement of Cyber-Attack Detection Accuracy from Urban Water Systems Using Extreme Learning Machine. *Applied Sciences*, 10(22), 8179. <https://doi.org/10.3390/app10228179>
- Donoso, M. C. (2022). ¿Cuán importante es la seguridad cibernética para lograr la seguridad hídrica? *Revista de Ciencias Ambientales*, 56(1), 284-297. <https://doi.org/10.15359/rca.56-1.15>
- DRAGOS. (2022). *ICS/OT CYBERSECURITY YEAR IN REVIEW 2022*. <https://www.dragos.com/blog/industry-news/2022-dragos-year-in-review-now-available/>
- Gómez, J., Reyes, R., & Guzmán del Río, D. (2008). *Temas especiales de instrumentación y control*. Félix Varela.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la Investigación (6ª edición)*. McGRAW-HILL/McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.
- Housh, M., Kadosh, N., & Haddad, J. (2022). Detecting and Localizing Cyber-Physical Attacks in Water Distribution Systems without Records of Labeled Attacks. *Sensors*, 22(16), 6035. <https://doi.org/10.3390/s22166035>

- Kadosh, N., Frid, A., & Housh, M. (2020). Detecting Cyber-Physical Attacks in Water Distribution Systems: One-Class Classifier Approach. *Journal of Water Resources Planning and Management*, 146(8), 04020060. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0001259](https://doi.org/10.1061/(ASCE)WR.1943-5452.0001259)
- Kim, S., Jun, S., & Jung, D. (2022). Ensemble CNN Model for Effective Pipe Burst Detection in Water Distribution Systems. *Water Resources Management*, 36(13), 5049-5061. <https://doi.org/10.1007/s11269-022-03291-1>
- Kravchik, M., & Shabtai, A. (2022). Efficient Cyber Attack Detection in Industrial Control Systems Using Lightweight Neural Networks and PCA. *IEEE Transactions on Dependable and Secure Computing*, 19(4), 2179-2197. <https://doi.org/10.1109/TDSC.2021.3050101>
- M. Arif, W., Farooq Ahmad, B., Saduf, A., & Asif Iqbal, K. (2020). *Advances in Deep Learning* (Vol. 57). Springer. <http://www.springer.com/series/11970>
- Mahmoud, H., Wu, W., & Gaber, M. M. (2022). A Time-Series Self-Supervised Learning Approach to Detection of Cyber-physical Attacks in Water Distribution Systems. *Energies*, 15(3), 914. <https://doi.org/10.3390/en15030914>
- Mboweni, I. V., Ramotsoela, D. T., & Abu-Mahfouz, A. M. (2023). Hydraulic Data Preprocessing for Machine Learning-Based Intrusion Detection in Cyber-Physical Systems. *Mathematics*, 11(8), 1846. <https://doi.org/10.3390/math11081846>
- Mishra, V. K., Palleti, V. R., & Mathur, A. (2019). A modeling framework for critical infrastructure and its application in detecting cyber-attacks on a water distribution system. *International Journal of Critical Infrastructure Protection*, 26, 100298. <https://doi.org/10.1016/j.ijcip.2019.05.001>
- Moazeni, F., & Khazaei, J. (2021a). Formulating false data injection cyberattacks on pumps' flow rate resulting in cascading failures in smart water systems. *Sustainable Cities and Society*, 75, 103370. <https://doi.org/10.1016/j.scs.2021.103370>

- Moazeni, F., & Khazaei, J. (2021b). Sequential false data injection cyberattacks in water distribution systems targeting storage tanks; a bi-level optimization model. *Sustainable Cities and Society*, 70, 102895. <https://doi.org/10.1016/j.scs.2021.102895>
- Mokhtari, S., Abbaspour, A., Yen, K. K., & Sargolzaei, A. (2021). A Machine Learning Approach for Anomaly Detection in Industrial Control Systems Based on Measurement Data. *Electronics*, 10(4), 407. <https://doi.org/10.3390/electronics10040407>
- M.R., G. R., & P. Mathur, A. (2022). AICrit: A unified framework for real-time anomaly detection in water treatment plants. *Journal of Information Security and Applications*, 64, 103046. <https://doi.org/10.1016/j.jisa.2021.103046>
- Nikolopoulos, D., & Makropoulos, C. (2022). Stress-testing water distribution networks for cyber-physical attacks on water quality. *Urban Water Journal*, 19(3), 256-270. <https://doi.org/10.1080/1573062X.2021.1995446>
- Ñaupas Paitán, H., Mejía Mejía, E., Novoa Ramírez, E., & Villagómez Paucar, A. (2014). *Metodología de la investigación Cuantitativa—Cualitativa y Redacción de la Tesis* (4ta. Edición). Ediciones de la U.
- Ostfeld, A., Salomons, E., Ormsbee, L., Uber, J. G., Bros, C. M., Kalungi, P., Burd, R., Zazula-Coetzee, B., Belrain, T., Kang, D., Lansey, K., Shen, H., McBean, E., Yi Wu, Z., Walski, T., Alvisi, S., Franchini, M., Johnson, J. P., Ghimire, S. R., ... McKillop, R. (2012). Battle of the Water Calibration Networks. *Journal of Water Resources Planning and Management*, 138(5), 523-532. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000191](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000191)
- Pedamkar, Priya. (2023). *Machine Learning Models*. <https://www.educba.com/machine-learning-models/>

Perales Gómez, Á. L., Fernández Maimó, L., Huertas Celdrán, A., & García Clemente, F. J. (2020).

MADICS: A Methodology for Anomaly Detection in Industrial Control Systems. *Symmetry*, 12(10), 1583. <https://doi.org/10.3390/sym12101583>

Pérez López, Esteban. (2015, febrero 27). Los sistemas SCADA en la automatización industrial.

Tecnología en Marcha, 28(4), 3-14.

Pinto, A., Herrera, L.-C., Donoso, Y., & Gutierrez, J. A. (2023). Survey on Intrusion Detection Systems

Based on Machine Learning Techniques for the Protection of Critical Infrastructure. *Sensors*, 23(5), 2415. <https://doi.org/10.3390/s23052415>

Priyanga, S., Gauthama Raman, M. R., Jagtap, S. S., Aswin, N., Kirthivasan, K., & Shankar Sriram, V. S.

(2019). An Improved Rough Set Theory based Feature Selection Approach for Intrusion Detection in SCADA Systems. *Journal of Intelligent & Fuzzy Systems*, 36(5), 3993-4003.

<https://doi.org/10.3233/JIFS-169960>

Ramirez Gil, W. A., & Ramirez Gil, C. M. (2023). *Programación de Inteligencia Artificial. Curso Practico*.

Ra-ma.

Ramotsoela, D. T., Hancke, G. P., & Abu-Mahfouz, A. M. (2019). Attack detection in water distribution

systems using machine learning. *Human-Centric Computing and Information Sciences*, 9(1), 13.

<https://doi.org/10.1186/s13673-019-0175-8>

Ramotsoela, T. D., Hancke, G. P., & Abu-Mahfouz, A. M. (2020). Behavioural Intrusion Detection in

Water Distribution Systems Using Neural Networks. *IEEE Access*, 8, 190403-190416.

<https://doi.org/10.1109/ACCESS.2020.3032251>

Rodríguez Martínez, C., Quiñones-Grueiro, M., & Llanes-Santiago, O. (2023). Cyberattack Diagnosis in

Water Distribution Networks Combining Data-Driven and Structural Analysis Methods. *Journal of Water Resources Planning and Management*, 149(5), 04023013.

<https://doi.org/10.1061/JWRMD5.WRENG-5302>

Rodríguez Penin, A. (2011). *Sistemas SCADA (3ª edición)* (3ª edición). Marcombo.

Rosebrock, A. (2020, febrero 17). *Autoencoders with Keras, TensorFlow, and Deep Learning*.

<https://pyimagesearch.com/2020/02/17/autoencoders-with-keras-tensorflow-and-deep-learning/>

Selim, G. E. I., Hemdan, E. E.-D., Shehata, A. M., & El-Fishawy, N. A. (2021). Anomaly events classification and detection system in critical industrial internet of things infrastructure using machine learning algorithms. *Multimedia Tools and Applications*, 80(8), 12619-12640.

<https://doi.org/10.1007/s11042-020-10354-1>

Sevillano, F., Rodriguez, A., Valencia, A., Echave, A., Sanchez, S., Matilla, E., & De Pablo, E. (2021).

Ciberseguridad Industrial e Infraestructuras Críticas (Ediciones de la U). Ra-ma.

Sheng, C., Yao, Y., Fu, Q., & Yang, W. (2021). A cyber-physical model for SCADA system and its intrusion detection. *Computer Networks*, 185, 107677. <https://doi.org/10.1016/j.comnet.2020.107677>

Suaboot, J., Fahad, A., Tari, Z., Grundy, J., Mahmood, A. N., Almalawi, A., Zomaya, A. Y., & Drira, K.

(2021). A Taxonomy of Supervised Learning for IDSs in SCADA Environments. *ACM Computing Surveys*, 53(2), 1-37. <https://doi.org/10.1145/3379499>

Taormina, R., & Galelli, S. (2018). Deep-Learning Approach to the Detection and Localization of Cyber-Physical Attacks on Water Distribution Systems. *Journal of Water Resources Planning and Management*, 144(10), 04018065. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000983](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000983)

Taormina, R., Galelli, S., Tippenhauer, N. O., Salomons, E., Ostfeld, A., Eliades, D. G., Aghashahi, M., Sundararajan, R., Pourahmadi, M., Banks, M. K., Brentan, B. M., Campbell, E., Lima, G., Manzi, D., Ayala-Cabrera, D., Herrera, M., Montalvo, I., Izquierdo, J., Luvizotto, E., ... Ohar, Z. (2018). Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks. *Journal of Water Resources Planning and Management*, 144(8), 04018048. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000969](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000969)

- Tsiami, L., & Makropoulos, C. (2021). Cyber—Physical Attack Detection in Water Distribution Systems with Temporal Graph Convolutional Neural Networks. *Water*, 13(9), 1247.
<https://doi.org/10.3390/w13091247>
- Upadhyay, D., & Sampalli, S. (2020). SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations. *Computers & Security*, 89, 101666.
<https://doi.org/10.1016/j.cose.2019.101666>
- Valdivia Miranda, C. (2019). *Comunicaciones Industriales* (Ediciones Paraninfo, SA). Ediciones Paraninfo, SA.
- Vara Horna, A. (2012). *Siete pasos para una tesis exitosa*.
- Wang, W., Harrou, F., Bouyeddou, B., Senouci, S.-M., & Sun, Y. (2022). Cyber-attacks detection in industrial systems using artificial intelligence-driven methods. *International Journal of Critical Infrastructure Protection*, 38, 100542. <https://doi.org/10.1016/j.ijcip.2022.100542>
- Zhang, X.-D. (2020). *A Matrix Algebra Approach to Artificial Intelligence*. Springer.
<https://doi.org/10.1007/978-981-15-2770-8>