PROCEDIMIENTO DE DISEÑO DE SISTEMAS CIBERFÍSICOS DE TIEMPO REAL TOLERANTES A ATAQUES CIBERNÉTICOS

CARLOS MARIO PAREDES VALENCIA

2171993

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE - UAO
PROGRAMA DE DOCTORADO EN INGENIERÍA
LÍNEA EN AUTOMÁTICA
CALI, COLOMBIA
2022

PROCEDIMIENTO DE DISEÑO DE SISTEMAS CIBERFÍSICOS DE TIEMPO REAL TOLERANTES A ATAQUES CIBERNÉTICOS

CARLOS MARIO PAREDES VALENCIA

Tesis para optar al título de Doctor en Ingeniera

Director
DIEGO MARTÍNEZ CASTRO
PhD. Automática, Robótica e Informática Industrial

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE - UAO
PROGRAMA DE DOCTORADO EN INGENIERÍA
LÍNEA EN AUTOMÁTICA
CALI, COLOMBIA
2022

Declaro que esta presentación es mi propia obra y que, a mi leal saber y entender, no contiene ningún material previamente publicado o escrito por otra persona ni material que en gran medida haya sido aceptado para la concesión de cualquier otro título o diploma de la Universidad u otro instituto de enseñanza superior, excepto cuando se haya hecho el debido reconocimiento en el texto.

Carlos Mario Paredes Valencia

Diciembre 9, 2021

AGRADECIMIENTOS

Expreso mis agradecimientos a mi familia por todo el apoyo brindado a lo largo de este proceso. A mi Director de tesis doctoral, el Dr. Diego Martínez Castro por todo su aporte tanto profesional y personal que me permitió avanzar en el desarrollo de esta propuesta. Del mismo modo agradezco al Dr. Apolinar González Potes que me permitió evaluar el desempeño de la propuesta en un contexto real a través de la pasantía. De igual manera agradezco a la institución académica Universidad Autónoma de Occidente, por brindarme la oportunidad para llevar a cabo mi estudio doctoral y a la Universidad de Colima por permitirme lleva a cabo el proceso de pasantía. Finalmente al grupo de compañeros de trabajo del grupo GITCoD que de una u otra forma hicieron parte de este proceso, brindándome el apoyo necesario cuando lo requerí.

CONTENIDO

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

LISTA DE ACRÓNIMOS

AML: Adversarial Machine Learning ANN: Artificial Neural Networks
CCE: CCECategorical crossentropy
CMMS: Computerized Maintenance Management System
CNN: Convolutional Neural Network
CPS: Cyber Physical System
CPU: Central Processing Unit
CSI: Current Source Inverter
DBN: Deep Belief Networks
DCS: Distributed Control System
DDOS: Distributed Deniel of Service
DDS: Data Distribution Service
DMZ: Demilitarized Zone
DoS: Deniel of Service, Deniel of Service
EDF: Earliest deadline first
ERP: Enterprise Resource Planning
FDI: Fault Detection and Isolation
FMEA: Failure Mode and Effect Analysis
FPR: False positive rate
FTC: Fault Tolerant Control
HMI: Human Machine Interface
HTTPS: Hypertext Transfer Protocol Secure
ICS-CERT: Industrial Control Systems Cyber Emergency Response Team
IDS: Intrusion Detection System
IEC: International Electrotechnical Commission
ISO: International Organization for Standardization
KNN: k-nearest neighbors
LAN: Local Area Network
LO: Luenberger Observer
LSTM: Long short-term memory
OPC: Open Platform Communications
MLP: Multiple Layer Perceptron
MMU: Memory Management Unit
MSE: Mean squared error
NIST: National Institute of Standars and Technology
NMS: Network Management System
PCA: Principal Component Analysis
PCC: Point of common coupling

PLC: Programmable Logic Controller
ReLU: Rectified Linear Unit
RL: Reinforcement Learning
RNN: Recurrent Neural Network
RTAI: Real-Time Application Interface
SCADA: Supervisory Control And Data Acquisition
SOAP: Simple Object Access Protocol
SSH: Secure Shell
SSL: Secure Sockets Layer
SVM: Support Vector Machine
SWaT: Secure Water Treatment
TIC: Tecnologías de la información y comunicación
TNR: True negative rate
TPR: True positive rate
UIO: Unknown Inputs Observer
VLAN: Virtual Local Area Network
VMM: Virtual Machine Manager
VPN: Virtual Private Network
VSI: Voltage Source Inverter
WSAN: Wireless sensor and actuator network

RESUMEN

Las aplicaciones emergentes de automatización industrial demandan gran flexibilidad en los sistemas, lo cual se logra con en el aumento de la interconexión entre sus módulos, permitiendo el acceso a toda la información del sistema y la reconfiguración en función de los cambios que se presentan durante su funcionamiento, con el propósito de alcanzar puntos óptimos de operación. Para ello se soportan en el concepto de sistemas ciberfísicos (CPSs, Cyber-physical Systems por sus siglas en inglés), los cuales se caracterizan por la integración de sistemas físicos y digitales para crear productos y procesos inteligentes capaces de transformar las cadenas de valor convencionales, lo que ha dado origen al concepto de Smart Factory.

Esta flexibilidad abre una gran brecha que afecta a la seguridad de los sistemas de control ya que los nuevos enlaces de comunicación pueden ser utilizados por personas para generar ataques que produzcan riesgo en estas aplicaciones. Este es un problema reciente en los sistemas de control, que originalmente estaban centralizados y posteriormente se implementaron como sistemas interconectados a través de redes aisladas. Actualmente, para proteger estos sistemas se ha optado por utilizar estrategias que han presentado resultados destacables en otros ambientes, como por ejemplo los ambientes de oficina. Sin embargo, las características de estas aplicaciones no son las mismas y los resultados alcanzados no son los deseados. Esta problemática ha motivado varios esfuerzos que pretenden contribuir desde diferentes enfoques a aumentar la seguridad de los sistemas de control.

Se realizó una revisión de las estrategias utilizadas actualmente para el diseño de redes de control seguras, y de las técnicas y tecnologías que buscan detectar ataques en los sistemas de control y contribuir a mitigar el efecto de los mismos. Esta revisión permitió identificar los ataques que tienen mayor frecuencia e impacto en estos sistemas, a partir de lo cual se seleccionaron los ataques que comprometen la integridad de las variables de los sistemas de control y los retrasos en el envío de los mensajes que contienen estos valores, para ser abordados en esta propuesta.

Con base en lo anterior, en este trabajo se propuso un procedimiento de diseño de aplicaciones de control soportadas en sistemas ciberfísicos que posibilita la implementación de estrategias de detección y tolerancia de ciberataques, el cual integra un enfoque modular y de fácil adaptación. Este procedimiento permite identificar los diversos componentes del sistema los cuales se establecen como microservicios, e integra un planteamiento para evaluar la planificabilidad de los componentes de la aplicación de control. Las etapas del procedimiento detallan el desarrollo de sistemas de detección y aislamiento de ciberataques que son usados para generar alarmas, a partir de las cuales

es posible definir qué elemento del sistema está siendo afectado, posibilitando el uso de estrategias soportadas en réplicas de componentes que permitan el reemplazo de los mismos para tolerar ciberataques.

La arquitectura y el procedimiento propuesto permiten el cumplimiento de los requisitos del sistema, y presentan un enfoque modular y de fácil adaptabilidad para el diseño.

Palabras clave: ciberataques, sistema ciberfísico, sistema de detección de ciberataques, virtualización.

ABSTRACT

Emerging industrial automation applications demand great flexibility in the systems, which is achieved by increasing the interconnection between its modules, allowing access to all system information and reconfiguration according to changes that occur during operation, in order to achieve optimal operating points. This is supported by the concept of Cyber-physical Systems (CPSs), which are characterized by the integration of physical and digital systems to create intelligent products and processes capable of transforming conventional value chains, which has given rise to the concept of Smart Factory.

This flexibility opens a big gap that affects the security of control systems because the new communication links can be used by individuals to generate attacks that produce risk in these applications. This is a recent problem in control systems, which were originally centralized and later implemented as interconnected systems through isolated networks. Currently, to protect these systems have chosen to use strategies that have presented remarkable results in other environments, such as office environments. However, the characteristics of these applications are not the same and the results achieved are not the desired ones. This problem has motivated several efforts that aim to contribute from different approaches to increase the security of control systems.

A review was made of the strategies currently used for the design of secure control networks, and of the techniques and technologies that seek to detect attacks on control systems and contribute to mitigate their effect. This review made it possible to identify the attacks that have the greatest frequency and impact on these systems, from which attacks that compromise the integrity of control system variables and delays in sending messages containing these values were selected to be addressed in this proposal.

Based on the above, this work proposed a procedure for the design of control applications supported by cyber-physical systems that enables the implementation of cyber-attack detection and tolerance strategies, which integrates a modular and easily adaptable approach. This procedure identifies the various components of the system, which are established as microservices, and integrates an approach to evaluate the plannability of the components of the control application. The stages of the procedure detail the development of cyber-attack detection and isolation systems that are used to generate alarms, from which it is possible to define which element of the system is being affected, enabling the use of strategies supported by replicas of components that allow their replacement to tolerate cyber-attacks.

The proposed architecture and procedure allow the system requirements to be met, and present a modular and easily adaptable approach to design.

Keywords: cyber-attacks, cyber-physical system, cyber-attack detection system, virtualization.

INTRODUCCIÓN

En los últimos años se han desarrollado varios tipos de sistemas ciberfísicos, los cuales han tenido un gran impacto en diversos sectores, como por ejemplo el energético, automovilístico, industrial, dispositivos en el sector médico, entre otros [1–4]. El concepto de sistema ciberfísico (CPS, Cyber-Physical System por sus siglas en inglés), surgió a partir del intento de unificar las aplicaciones emergentes de computadoras integradas y de las tecnologías de comunicación a una variedad de dominios y sectores físicos, cuyo objetivo es monitorear y controlar los procesos físicos [5].

Estos sistemas consisten en una combinación de dispositivos móviles, sistemas integrados y computadoras que se usan para monitorear, detectar y actuar sobre elementos físicos del mundo real para cumplir una tarea específica. Las partes informáticas que conforman este tipo de sistemas suelen interconectarse, usando redes de comunicación para compartir información y datos que interactúan entre ellos, y en ocasiones con servicios de computación en la nube [2–4].

Esto permite tener mayor flexibilidad en estos sistemas, en donde se tolera la integración de nuevos nodos, lo cual ha contribuido al aumento en la capacidad de cómputo, la cobertura y adaptabilidad de las aplicaciones. Sin embargo, al mismo tiempo plantea nuevos desafíos que se relacionan con la seguridad y la confiabilidad de las aplicaciones derivada de la vulnerabilidad que presentan frente a ciberataques, los cuales pueden generar afectaciones en la infraestructura física, generar impactos negativos en el medio ambiente y los costos de producción, alterar la calidad de los productos involucrados en los procesos, incluso hasta atentar contra la vida humana, y en general generar diversas afectaciones criticas [6, 7].

Además, varias aplicaciones de control soportadas en estos sistemas se pueden etiquetar como de seguridad crítica en relación al cumplimiento de plazos estrictos de tiempo real, asociados a la generación de acciones a partir de la interacción entre los sistemas computacionales y los sistemas físicos relacionados con la aplicación, debido a que el no cumplimiento de estos requisitos puede causar un daño irreparable al sistema físico que se controla, así como a las personas que dependen de ello [8].

Por otro lado, las mediciones y acciones de control pueden ser alteradas mientras se transmiten a través de las redes de comunicaciones. De esta manera se requieren nuevos algoritmos de control o arquitecturas de diseño que en presencia de situaciones adversas puedan llevar al sistema a estados seguros y estables [9, 10].

Existen reportes de ciberataques que han afectado el funcionamiento de estos siste-

mas. Uno de los más conocidos es el caso de Stuxnet, el cual fue un virus informático que tenía la capacidad de reprogramar los controladores lógicos programables de las centrífugas de una planta nuclear en Irán en el año 2010. Del mismo modo han ocurrido incidentes de carácter económico, como por ejemplo un caso de venta de combustible reportado en Malasia en el 2013, donde se alteraban el consumo para obtener ganancias ilegalmente [11].

Según el informe publicado por el equipo de preparación para emergencias informáticas de los sistemas de control industrial (ICS-CERT, Industrial Control Systems Cyber Emergency Response Team por sus siglas en inglés) [12], el número de incidentes relacionados con las violaciones de seguridad que involucran sistemas embebidos y sistemas ciberfísicos en el 2012 fue más de cinco veces comparado con el 2010. Este tipo de situaciones ha generado gran interés en la comunidad científica en los últimos años, en busca de alternativas en los diseños de los sistemas automáticos que tengan en consideración requisitos asociados al intercambio de información a través de redes de comunicaciones [13, 14].

En el pasado, los sistemas de control industrial enfrentaban amenazas como daños/-fallas físicas en sus diversas partes, que comprometían el funcionamiento normal del sistema sin cumplir de manera adecuada el fin para el cual fue diseñado y concebido. Hoy en día, debido al uso extensivo de la información y de las tecnologías de la comunicación en estos sistemas, lo hacen vulnerables a los ciberataques, como se comentó anteriormente.

Independientemente de la naturaleza de los ataques, estos incidentes muestran que los controles preventivos de seguridad, como las zonas desmilitarizadas tradicionales, la fuerte segregación de red y los múltiples firewalls, no siempre son suficientes para proteger los equipos en los sistemas de control involucrados en los CPSs. Por lo que los esfuerzos no solo deben ponerse en la prevención de ataques sino también en la detección y corrección o mitigación de estos, así como en el diseño de arquitecturas que permitan tener cierto grado de tolerancia frente a este tipo de situaciones.

Con base en lo anterior, el propósito de esta investigación se orientó al desarrollo de nuevas estrategias de diseño para aportar a la seguridad de las aplicaciones de control soportadas en CPSs, con relación a ciberataques que afecten la integridad de las mediciones y acciones de control, o los retrasos en el envío de los mensajes que contienen esta información.

Este trabajo se compone de siete capítulos organizados de la siguiente manera. En el primer capítulo se expone el problema de investigación. En el segundo capítulo se

plantean los objetivos del proyecto de investigación. El capítulo tres presenta el marco conceptual y los antecedentes que enmarcan la propuesta. En el cuarto capítulo se presenta el análisis de los efectos de ciberataques en las redes de control. El quinto capítulo presenta el procedimiento de diseño propuesto. El sexto capítulo presenta el uso de la propuesta para el abordaje de dos casos de estudio. Finalmente, se realiza una síntesis de las conclusiones del trabajo desarrollado y las líneas futuras de investigación.

1. PROBLEMA DE INVESTIGACIÓN Y JUSTIFICACIÓN DEL PROYECTO

El presente capítulo tiene el propósito de presentar el problema de investigación, así como las preguntas de investigación, el planteamiento de la hipótesis y la justificación de la investigación.

El capítulo se organiza de la siguiente manera. En la primera sección se mencionan diferentes aspectos que conllevan al planteamiento del problema de investigación. En la siguiente sección se plantean las preguntas de investigación, continuando con la hipótesis y finalizando con la justificación de la propuesta.

1.1 PROBLEMA DE INVESTIGACIÓN

En las últimas décadas, los avances en las tecnologías de la información y de las comunicaciones se han trasladado a los procesos de automatización, dando origen a nuevos sistemas y esquemas de control, lo que condujo al desarrollo de CPSs integrados en entornos industriales. Estos sistemas ofrecen grandes ventajas relacionadas con su alta flexibilidad y sus capacidades de adaptabilidad a diversos escenarios de funcionamiento de las aplicaciones. Sin embargo, han surgido nuevo desafíos relacionados con su funcionamiento, particularmente se requiere un avance en los enfoques de seguridad, debido a que las herramientas clásicas de ciberseguridad han presentado vulnerabilidades en estos escenarios [15].

La investigación en seguridad informática se ha centrado tradicionalmente en la protección de la información corporativa. Sin embargo aún quedan desafíos por resolver en lo que respecta a la protección en los sistemas de control. Por lo tanto, si bien los métodos actuales de seguridad de la información pueden proporcionar los mecanismos necesarios para la seguridad de los sistemas de control, estos mecanismos por sí solos no son suficientes para la defensa en profundidad de estos sistemas. La ejecución de software desconocido y malicioso sobre los algoritmos de control, puede desencadenar comportamientos que alteren el funcionamiento del sistema, además del acceso a datos confidenciales [16].

Si bien es cierto que los dispositivos integrados han transformado la manera como se crea, se comparte, se procesa y se administra la información, la seguridad en estos dispositivos son tareas desafiantes, debido a la naturaleza limitada de los recursos (cantidad de memoria para su uso, unidades de cómputo, consumo de energía, entre otras). Un ejemplo de esto se puede encontrar en el sector de la salud, en donde los dispositivos embebidos se usan diariamente para procesar datos médicos confidenciales y que realizan funciones críticas para múltiples pacientes. Aunque se han desarrollado méto-

dos de protección de información (algoritmos criptográficos y protocolos de seguridad), las soluciones son incompatibles con muchas arquitecturas integradas y no se pueden usar, debido a un sistema operativo o firmware personalizado, presupuestos de energía limitados y recursos computacionales altamente restringidos [17].

Se hace necesario destacar que uno de los principales problemas en la seguridad de estos sistemas es la heterogeneidad que se tienen en las diversas partes que lo componen. Existen diferentes componentes de hardware tales como sensores, actuadores y sistemas embebidos. Además, hay diversos distribuidores de software, para controlar y monitorear los procesos. Como resultado, cada componente y su integración puede contribuir a los factores que hacen que un CPS pueda ser vulnerable a ataques cibernéticos [1]. Un software o un conjunto de sensores comprometidos, puede causar daños potenciales conduciendo a los actuadores a estados que son incompatibles con las condiciones físicas [18].

Las técnicas utilizadas actualmente para reducir los riesgos de seguridad tienen básicamente dos enfoques: las herramientas tradicionales de seguridad de la información y el uso de teorías de análisis de señales y sistemas. Las herramientas tradicionales de seguridad de la información incluyen mecanismos de autenticación y control de acceso, sistemas de detección de intrusos en la red, administración de parches de seguridad, entre otros. En la práctica, la efectividad de estas herramientas es limitada, dado que no se puede proporcionar una defensa en tiempo real y eficiente para ataques cibernéticos a través de las actualizaciones de parches de seguridad y criterios de certificación. Mientras que las técnicas que se soportan en el análisis de señales y sistemas, utilizan enfoques similares a los métodos de detección y aislamiento de fallas y control tolerante a fallos, donde asumen el ciberataque como un fallo en el sistema. En términos generales, al alterar ciertos datos significativos los atacantes pueden desestabilizar la planta.

De esta manera se requieren nuevas estrategias y algoritmos que aporten a la detección y mitigación de los impactos generados por estos ataques. Las estrategias deben permitir el desarrollo de sistemas de control confiables, con capacidades de detección de anomalías y reconfiguraciones de operación en presencia de incertidumbres, fallas en los componentes y ataques de adversarios [9].

Se cree que comprender las interacciones del control del sistema con el mundo físico conlleve a la capacidad de desarrollar marcos generales y sistemáticos que permitan asegurarlos frente a ciberataques [2, 19].

Con base en lo anterior, el propósito de esta investigación es contribuir con nuevas estrategias y procedimientos al desarrollo de CPSs para aplicaciones de control que tengan

un comportamiento seguro, en relación a ciberataques que afecten la integridad de las mediciones y acciones de control, o los retrasos en el envío de esta información.

Es por esta razón que se plantea la siguiente pregunta que abarca la problemática mencionada: ¿Cómo diseñar CPSs de tiempo real para aumentar su tolerancia frente a ataques cibernéticos que afecten los valores de las variables del proceso?.

1.2 PREGUNTA(S) DE INVESTIGACIÓN

A partir de lo anterior se plantearon las siguientes preguntas de investigación:

- ¿Cómo detectar ciberataques a partir del conocimiento de la dinámica de los ataques y de los sistemas?
- ¿Cómo detectar y aislar múltiples ciberataques que se presenten de manera simultánea?
- ¿Cómo diseñar arquitecturas de sistemas que permitan detectar y aislar ciberataques, y la reconfiguración de CPSs para aumentar su tolerancia a ataques cibernéticos?

1.3 HIPÓTESIS DE INVESTIGACIÓN

La hipótesis en la que se soportó este trabajo fue la siguiente:

La coordinación de métodos de detección y aislamiento de ataques, junto con estrategias de reconfiguración soportadas en técnicas de virtualización, permitirá desarrollar CPSs tolerantes a ciberataques.

1.4 JUSTIFICACIÓN DE LA INVESTIGACIÓN

Actualmente la sociedad depende de múltiples sistemas automáticos soportados en sistemas ciberfísicos. Como se ha mencionado, estas aplicaciones se encuentran hoy día en sectores tales como el industrial, energético, salud, medioambientales, etc. La seguridad y confiabilidad son requisitos fundamentales en estos sistemas.

Debido a esto, intervenciones de agentes ajenos a los procesos que generen comportamientos inadecuados pueden tener impactos catastróficos en el mundo físico, causando daño tanto en la infraestructura del sistema e incluso atentar contra vidas humanas [20]. Esto ha causado que esta área de la investigación haya estado muy activa en los últimos años. Ataques a smartgrids, sistemas de aviación, plantas de agua, plantas químicas, sistemas de distribución de petróleo y gas natural, son cada vez más frecuentes.

Hoy en día asegurar los sistemas de control industrial es una tarea que cada vez tiene mayor importancia. Estándares como NERC CIP, ISA 95, ISA-99(IEC-62443) y NIST 800-82, se han elaborado con la idea de poder ayudar a identificar e implementar mejores prácticas de seguridad en estos sistemas [21].

En sistemas de control de procesos, por ejemplo, la presión en un tanque podría acumular niveles peligrosos si el sistema SCADA es afectado por un intruso, pudiendo incluso producir una explosión. También desde un punto de vista económico los ingresos de las entidades podrían estar en juego si se manipula el sistema SCADA; un ejemplo de esto puede ser influir sobre una reacción química que altere la calidad o la productividad en un proceso industrial.

Por lo tanto pueden ocurrir consecuencias económicas o problemas de seguridad si los sistemas SCADA que administran estos procesos se ven comprometidos [6]. La razón por la cual la investigación en la seguridad de los sistemas SCADA se ha vuelto importante es porque de hecho han ocurrido incidentes en los que los sistemas SCADA se han visto comprometidos [22].

Otros casos han evidenciado las vulnerabilidades que presentan estos sistemas. En el 2010 se descubrió el malware Stuxnet, diseñado para infiltrarse en los sistemas SCADA en componentes específicos de hardware y software; éste fue el primer malware conocido, diseñado especialmente para comprometer el software de un PLC (Programmable Logic Controller por sus siglas en inglés), causando daños físicos en maquinarias pesadas como turbinas a vapor y centrífugas de gas, presentes en plantas de procesos industriales. Una de las principales características de este malware fue su capacidad de propagación y afectación a las acciones de control del proceso, y capacidad de ser detectado [23].

Aunque revistas como la IEEE Transactions on Control Systems han dedicado números especiales a la seguridad informática en los sistemas de control, el trabajo en esta área se encuentra en etapas ascendentes. Problemas de cómo detectar ataques y cómo ajustar el sistema del controlador para contrarrestar los efectos de un ciberataque aún no se han resuelto en su totalidad [24], lo que limita garantizar la confiabilidad del sistema, y evitar daños tanto en las estructuras y procesos que estos controlen; así como genera efectos negativos hacia el mundo exterior, impactando los costos de producción, el medio ambiente y/o la integridad de las personas.

Seguirán surgiendo nuevas amenazas [25], y de ahí la necesidad de diseñar nuevos métodos de protección de información en estos sistemas, así como nuevos sistemas de detección de anomalías donde se tenga en cuenta estos eventos no deseados.

2. OBJETIVOS DE LA TESIS

El presente capítulo tiene el propósito de presentar el objetivo general y los específicos de la propuesta de investigación.

2.1 OBJETIVO GENERAL

Desarrollar un procedimiento de diseño de sistemas ciberfísicos de tiempo real para tolerar ataques cibernéticos que modifiquen los valores de las señales en lazos de control y afecten los retrasos en el lazo de realimentación.

2.2 OBJETIVOS ESPECÍFICOS

- Definir los ataques a considerar en la investigación.

- Desarrollar un procedimiento de diseño de sistemas ciberfísicos para tolerar los ataques definidos, soportado en una nueva arquitectura del sistema.

- Desarrollar un caso de estudio que permita experimentar el uso del procedimiento de diseño propuesto.

Con el fin de cumplir con los objetivos planteados se propuso desarrollar las actividades descritas en la Tabla I.

TABLA I.
Actividades metodológicas.

Objetivo específico		Actividad
1	1.1	Desarrollar modelos que permitan simular diferentes tipos de ataques cibernéticos reportados sobre sistemas ciberfísicos.
	1.2	Evaluar mediante simulación el impacto de los ataques modelados
2	2.1	Proponer una nueva arquitectura de nodo que permita la reconfiguración de sistemas ciberfísicos para tolerar los ataques cibernéticos definidos a ser considerados en la investigación.
	2.2	Proponer una estrategia para detectar los ataques cibernéticos definidos y reconfigurar el sistema para tolerarlos.
	2.3	Desarrollar modelos y métodos de validación del diseño de sistemas ciberfísicos que toleren los ataques cibernéticos definidos con base en la arquitectura y estrategia de reconfiguración propuestas.
	2.4	Desarrollar un procedimiento de diseño de sistemas ciberfísicos con base en las propuestas realizadas.
3	3.1	Especificar el caso de estudio a implementar.
	3.2	Modelar e implementar un prototipo del caso de estudio.
	3.3	Evaluar mediante simulación y experimentación el desempeño del sistema frente a la presencia de los ataques definidos.

3. MARCO REFERENCIAL

En el presente capítulo se exponen los conceptos, las teorías y los antecedentes relacionados con el proyecto de investigación.

El capítulo se organiza de la siguiente manera. Primero se presentan los conceptos más afines al desarrollo del proyecto, seguido se exponen las teorías utilizadas para el desarrollo de la propuesta. En la tercera sección se presentan los antecedentes y el estado actual de las temáticas. Finalmente se presentan las conclusiones del capítulo.

3.1 MARCO CONCEPTUAL

Esta sección tiene el propósito de presentar los diversos conceptos que se requieren para desarrollar el proyecto de investigación. Inicialmente se presenta el concepto de CPS y su arquitectura. Posteriormente se presenta la definición de sistema de tiempo real. Por último se describen los sistemas de detección de intrusos.

3.1.1 Sistema Ciberfísico

Las nuevas aplicaciones de sistemas de control se enmarcan en su mayoría, dentro de la categoría de CPS, los cuales son sistemas físicos controlados por algoritmos de computación que se integran a través de redes de comunicaciones, pero en los cuales los componentes físicos y de software están profundamente entrelazados.

En estos nuevos entornos surgen nuevas aplicaciones para lograr dar respuesta a diversas necesidades, y esto conllevo a diferentes tipos de desafíos [26–29], como lo son los relacionados al modelamiento y diseño de las tareas, la interconexión e interoperabilidad entre dispositivos heterogéneos, uso compartido de energía, problemas de seguridad, programación y control del sistema, capacidad de procesamiento de los dispositivos, así como las conexiones de red compartidas y la capacidad de cambios dinámicos en la disponibilidad de ciertos componentes, requiriendo de esta manera gestores de administración que asignen correctamente los recursos. Estas aplicaciones en la gran mayoría de casos requieren de un alto nivel de garantía de buen funcionamiento, que se relaciona principalmente con el cumplimiento de requisitos de tiempo real y la protección de la información.

La arquitectura de los CPSs, pueden ser representadas en tres capas, como se observa en la Fig. 1. La primera de ellas es la capa física, la cual puede ser dividida en dos sub-capas. Una de ellas representa la infraestructura física y la otra está asociada a los sensores y actuadores, los cuales son los encargados de tomar las lecturas del en-

torno y desarrollan las acciones de control requeridas para llevar el sistema a un estado determinado.

Fig. 1. Arquitectura de un CPS.

La segunda capa de la arquitectura consta de la capa de comunicaciones que implementa la transmisión de datos y permite la interacción entre la capa física y la capa de aplicaciones. Dada la heterogeneidad de estos sistemas, diferentes protocolos de comunicación pueden coexistir, como, por ejemplo, redes distribuidas inalámbricas de baja potencia o redes TCP / IP, que se virtualizan a través de la capa de red subyacente. Estas redes heterogéneas están interconectadas mediante puertas de enlace (gateways). Las puertas de enlace, incluyen despachadores para coordinar las comunicaciones a través de redes inalámbricas de sensores y actuadores (WSAN, Wireless sensor and actuator network por sus siglas en inglés) o enrutadores para transmitir la información entregada a los dispositivos de destino.

Finalmente, la capa cibernética en donde se tienen todas las aplicaciones distribuidas encontradas en el CPS. Esto consiste en paquetes de software dedicado a funciones específicas, como lo son la supervisión y el control remoto, la gestión de la red y agentes, bases de datos e interfaz hombre-máquina (HMI, Human Machine Interface por sus siglas en inglés), solo por nombrar algunas. Además, se encuentran la red corporativa, que contiene las estaciones de trabajo y el servidor de aplicaciones la cual está a cargo de la administración comercial e interacción con el cliente, y la DMZ, la cual es una red segmentada aparte que se conecta directamente con los firewalls, y permite el intercambio de información de modo seguro. Los servidores que contienen datos históricos y los servidores web que contienen los datos del sistema a los que se debe acceder desde las redes corporativas, se colocan en este segmento separado para mejorar la seguridad [30]. En esencia, proporcionan al usuario varias funcionalidades, que permite la abstracción de los datos recibidos y permitiendo, de forma algo transparente, la interacción entre redes, dispositivos y la infraestructura física.

3.1.2 Sistema de Tiempo real

Una aplicación de tiempo real es una aplicación cuyo desempeño no solo depende de la lógica de su algoritmo sino también del cumplimiento de parámetros temporales. Para ello, el comportamiento de sus elementos ha de ser predecible durante todo el tiempo de ejecución de la aplicación, lo que permite verificar el cumplimiento de propiedades que posibilitan garantizar el cumplimiento de los requisitos de la aplicación [31].

El plazo de respuesta es el tiempo máximo del que dispone el sistema para haber atendido completamente una tarea. Este tiempo se contabiliza a partir del instante en el que se activa el evento asociado a la tarea, el cual puede ser periódico o no. El tiempo que tarda la ejecución de una tarea no siempre es el mismo, y para ello se define el tiempo del peor caso de ejecución como el tiempo máximo que puede tardar la tarea en ejecución una vez se activa, para ello se tienen en cuenta el tiempo de cómputo de su algoritmo y la interferencia que puede sufrir su ejecución dependiendo de la política de planificación de tareas que se tenga implementada en el sistema. Un sistema de tiempo real funciona de manera correcta siempre y cuando los plazos de todas las tareas estén garantizados, es decir, que todas las tareas se ejecutan dentro de sus respectivos plazos cada vez que se activan. Por lo tanto, para cada una de las tareas que conforman el sistema, se debe garantizar que el tiempo del peor caso de ejecución sea menor o igual al plazo de respuesta de la tarea, para lo cual se realizan test de análisis de planificabilidad [32].

En el contexto de los sistemas de control soportados en CPSs, la aplicación se distribuye entre varios componentes que interactúan a través de mensajes. Por lo tanto en estas aplicaciones tienen restricciones de tiempo real con plazos de respuesta extremo-extremo. Este tiempo es medido desde el instante en que se realiza la medición de las variables necesarias para calcular la acción de control, hasta cuando se actúa sobre el sistema, y depende de los tiempos de finalización de las tareas y del envío de los mensajes.

El problema de evaluar la planificabilidad de un sistema distribuido de tiempo real es NP-hard. Como alternativa para realizar estas evaluaciones se emplean restricciones y heurísticas. Un enfoque común es asignar las tareas estáticamente a los nodos que conforman el sistema y localmente utilizar un algoritmo de planificación como RM (Rate Monotonic) o EDF (Earliest Deadline First) [33]. Por otro lado, las aplicaciones en sistemas distribuidos están caracterizadas por poseer relaciones de precedencia entre sus tareas. Si las tareas son estáticamente asignadas a procesadores independientes, las restricciones de tiempo extremo-extremo pueden ser analizadas por una teoría que considere la relación entre los jitter [34].

Aunque la planificación por prioridades fijas es la técnica de planificación en línea más popular en sistemas de tiempo real, el uso del algoritmo EDF está tomando cada vez más relevancia debido a sus beneficios en cuanto al manejo de los recursos del sistema, y por lo tanto ya viene soportado en diversos entornos de desarrollo de aplicaciones de tiempo real.

El análisis de planificabilidad para EDF [35], se define :

• El modelo de las tareas como $\Gamma = \{\tau_1, \tau_2, \tau_3\}$, con $\tau_i = (C_i, D_i, T_i)$, donde C_i, D_i y T_i son los respectivos valores de tiempo de cómputo del peor caso, plazo y periodo de la tarea τ_i.

• $H_\tau(t) = \sum_{i=1}^{n} C_i \left\lfloor \frac{t+T_i-D_i}{T_i} \right\rfloor$, es la cantidad de tiempo de cómputo que debe ser atendido por el procesador hasta el tiempo t para cumplir con los plazos de tiempo del sistema.

• Intervalo critico inicial (ICI) es el intervalo de tiempo entre cero y el primer instante en el que no hay activaciones pendientes $[0,\mathcal{R})$. $\mathcal{R}$ puede ser calculado de forma recursiva con el siguiente método:

◦ $K_0 = 1$.

◦ $G_\tau(t) = \sum_{i=1}^{n} C_i \left\lceil \frac{t}{P_i} \right\rceil$.

Termina cuando $K_i = K_{i+1}$. El valor de $\mathcal{R}$ es definido por el ultimo valor de K_i.

El test de planificabilidad para un conjunto de tareas ejecutadas en un mismo nodo, empleando una política de planificación EDF consiste en verificar la Ecuación 1:

$$H_\tau(t) \leq t \; \forall \; t \leq R \tag{1}$$

3.1.3 Sistema de detección de intrusos

Para evitar que se exploten vulnerabilidades de los dispositivos conectados a la red del sistema donde se vean afectaciones que involucren manipulaciones de los datos o la disponibilidad de estos, se han usado diferentes técnicas que abordan estas situaciones. Para gestionar la red y las vulnerabilidades, se usan sistemas de Detección de Intrusos (IDS, Intrusion Detection System por sus siglas en inglés), los cuales monitorean el tráfico de datos para poder identificar y proteger los sistemas de estas eventualidades.

La actividad de estos sistemas se divide en tres etapas: monitorear, analizar y detectar. La etapa de monitoreo depende de una red de sensores o de un host-basado en senso-

res. La etapa del análisis utiliza un método para identificar y extraer características. La última etapa, depende del sistema que detecta las anomalías [36, 37].

Según estudios previos, los métodos IDS basados en host, pueden detectar ciberataques en diferentes canales de comunicación. Este tipo de IDS trabajan con la instalación de software tales como antivirus y detectan actividades sospechosas sobre el tráfico de la red, analizando actividades tales como llamadas del sistema, registro de aplicaciones, archivos del sistema, entre otros. Sin embargo estos métodos no presentan un correcto funcionamiento en algunos dispositivos, debido a las funcionalidades y recursos limitados que estos presentan [36, 38].

Los IDS tradicionales pueden realizar la detección en un canal o en toda la red. Estos monitorean el tráfico entero de la red para poder detectar ataques conocidos o desconocidos, para lo cual utilizan técnicas basadas en anomalías y en firmas. El método basado en firmas demanda más recursos, este método detecta solo los registros que se encuentran almacenados en una base de datos, aunque tienen una alta precisión y efectividad con amenazas conocidas [39].

Mientras que el método basado en anomalías es más eficiente para detectar nuevos ataques [36], dado que compara las actividades del sistema en el instante contra un perfil de comportamiento normal y genera alertas siempre que se exceda un umbral definido por el comportamiento normal del sistema [40]. Sin embargo, cualquier situación que no coincida a un comportamiento normal se considera una intrusión y aprender todo el comportamiento normal es bastante complejo, por lo que este método generalmente tiene altas tasas de falsos positivos [39].

Se pueden incluir también métodos basados en especificaciones. Una especificación no es más que un conjunto de reglas y umbrales que definen el comportamiento esperado de los diferentes componentes de la red, tales como nodos, protocolos y tablas de enrutamiento. Este método detecta intrusos cuando el comportamiento del sistema se desvía del comportamiento normal. Tiene el mismo propósito de los métodos basados en anomalías, con la diferencia de que este método es definido manualmente por un experto, el cual define las especificaciones. Las especificaciones definidas manualmente suelen proporcionar tasas bajas de falsos positivos en comparación con la detección basada en anomalías y no requieren etapas de entrenamiento, después de que estén definidas puede empezar a trabajar de manera inmediata. Sin embargo, estos métodos no se pueden adaptar a diferentes entornos y pueden consumir mucho tiempo para adaptarse, además de ser propensas al error [39].

Actualmente existe interés por el desarrollo de técnicas híbridas, para maximizar las

ventajas de unas técnicas y minimizar el impacto de los diferentes inconvenientes que tienen de manera individual [39, 41].

3.2 MARCO TEÓRICO

En esta sección se exponen las teorías y técnicas consultadas para abordar la problemática de investigación.

Esta sección se organiza como sigue. En la primera parte se expondrán métodos analíticos basados en la teoría de control donde se busca formalizar la manera sobre la cual se puede detectar y ubicar una anomalía dentro de un sistema de control. En la segunda parte se expondrán algunos métodos basados en máquinas de aprendizaje que se han usado para abordar problemática de seguridad en estos sistemas. En la última sección se exponen algunas de las tecnologías utilizadas para contribuir con soluciones a esta problemática.

3.2.1 Detección y aislamiento de fallas

Las técnicas de teoría de control aplicadas para automatizar procesos aseguran la estabilidad del sistema en lazo cerrado teniendo un rendimiento predefinido en el caso donde todos los componentes del sistema operan de forma segura. Sin embargo, en estos procesos pueden ocurrir situaciones anormales o fallas. En consecuencia, un lazo de control realimentado puede resultar en un desempeño insatisfactorio en caso de mal funcionamiento de los actuadores, sensores u otros componentes del sistema. Esto puede llevar al sistema a inestabilizarse. En muchos procesos industriales donde el mantenimiento o la reparación no siempre pueden ser de inmediato, es conveniente diseñar métodos de control capaces de lograr el rendimiento deseado teniendo en cuenta la ocurrencia de estas fallas. Estas técnicas de control se conocen con Control Tolerante a Fallos (FTC, Fault Tolerant Control por sus siglas en inglés). El diseño de estos sistemas requiere de una detección y aislamiento rápido de los fallos (FDI, Fault Detection and Isolation por sus siglas en inglés) para tomar las decisiones adecuadas. Por lo tanto, para preservar la seguridad de operadores y la confiablidad de los procesos, la presencia de fallas debe ser tenida en cuenta durante el diseño de sistemas de control [42].

La detección y el aislamiento de fallas se refiere a la tarea de inferir la ocurrencia de la falla en un proceso y encontrar sus raíces usando varias estrategias centradas en sistemas basados en el conocimiento: modelos cualitativos, modelos cuantitativos y datos históricos. Basado en el conocimiento del modelo/proceso se permite generar residuos sensibles a fallas específicas e insensibles a otras, posibilitando la detección y ubicación

del origen de la falla, o como en este caso se va a considerar, el ciberataque.

Durante el desarrollo de esta propuesta se piensa usar esta técnica para poder detectar e identificar en que parte del sistema está ocurriendo un ciberataque. Es decir, se tomará el ciberataque como si fuera una falla, que altera el comportamiento normal del sistema de control. De este modo a continuación se describe el funcionamiento de la teoría de detección y aislamiento de fallas. En sistemas FDI, generalmente se hace una distinción entre fallas aditivas y fallas multiplicativas. Sin embargo, los FTC tienen como objetivo compensar el efecto de la falla independientemente de la naturaleza de la falla.

Las fallas en el sistema se muestran a menudo como una variación de los parámetros del sistema. Así, en la presencia de las fallas el sistema puede ser modelado como se muestra en las Ecuaciones (2) y (3):

$$x(k+1) = Ax(k) + Bu(k) + F_a f_a(k) \tag{2}$$

$$y(k) = Cx(k) + F_s f_s(k) \tag{3}$$

Donde $F_a = B$ y $f_a = (\Gamma - I)U + U_{f0}$, donde ΓU y U_{f0} representa el efecto de una falla multiplicativa en el actuador y un efecto aditivo en la falla del actuador. Las Ecuaciones (4) y (5) describe el comportamiento de estos efectos.

$$\Gamma = diag(\alpha),\ \alpha = [\alpha_1 ... \alpha_i ... \alpha_m]^T \tag{4}$$

$$U_{f0} = [u_{f0} ... u_{f0i} ... u_{f0m}] \tag{5}$$

Si el i-ésimo actuador esta defectuoso, $\alpha_i \neq 1$ o $u_{f0i} \neq 0$, entonces la matriz F_a corresponde a la i-ésima columna de la matriz B y f_a corresponde a la magnitud de la falla que afecta directamente al actuador(controlador).

Así, de manera similar, en la presencia de las fallas (ataques) en el sensor, la matriz F_s es la i-ésima fila de la matriz C y el vector de fallas f_s, es la magnitud de la falla que afecta el i-ésimo sensor.

De este modo, cualquier sistema de control en donde se encuentre susceptible a que sus señales de control y/o variables medidas puedan ser hackeadas, se puede modelar como una combinación de los dos modelos definidos anteriormente, en las ecuaciones (2) y (3).

En resumen, la técnica FDI permite evitar consecuencias críticas y ayuda a tomar decisiones apropiadas cuando se presentan fallas en el sistema. Este proceso consta de dos tareas primordiales:

• Detección de la falla, que consiste en decidir si ha ocurrido una falla o no.

• Aislamiento de la falla, que consiste en decidir qué elemento(s) del sistema están siendo afectados por el fallo.

El procedimiento para realizar estas tareas se puede agrupar en tres pasos:

• **Generación residual:** este proceso consiste en comparar la salida medida con una salida estimada. La salida estimada puede ser proveniente de un modelo cualitativo, cuantitativo o basado en datos históricos. A esta señal se le denomina residual, $res(k)$, y se describe en la Ecuación (6):

$$res(k) = y(k) - \hat{y}(k) \tag{6}$$

Donde $y(k)$, son las medidas provenientes del proceso real y $\hat{y}(k)$ son el conjunto de salidas estimadas por cualquiera de los modelos mencionados. Se pueden usar otro tipo de normas como la 1 o la 2, dependiendo del proceso de evaluación.

• **Evaluación residual:** es el proceso de comparación de los residuales con algunos umbrales predefinidos. Esto se lleva a cabo con la Ecuación (7).

$$|res(k)| > \tau_{thresholds} \tag{7}$$

Los umbrales son definidos a través de una prueba y una etapa en donde se producen unos síntomas $S(r)$, los cuales permitirán detectar y aislar los ataques.

• **Toma de decisiones:** es un proceso de decisión a través de indicadores.

Relacionando está técnica con la detección e ataques, estos tres pasos implican el diseño de residuales que deben tomar valores cercanos a 0 en situaciones de que el sistema no se encuentre bajo ataque. En el caso contrario, cuando el sistema sea atacado, las señales del residuo deberán tener valores diferentes a 0.

Una sola señal residual puede alertar o detectar un ciberataque. Sin embargo, se requiere de un conjunto de residuales para poder aislarlo. Así, para ubicar el origen del ciberataque se requiere que algunos residuos tengan cierta sensibilidad solo para una parte en particular del sistema. Esto implica que el conjunto de residuales debe ser independiente de otros ciberataques definidos. De este modo, para aislar los ciberataques se considera un banco de residuos estructurados, en donde cada vector residual puede usarse para detectar un ciberataque en un lugar o parte del sistema en particular.

Con anterioridad, se ha nombrado la importancia de la etapa de la generación residual para poder detectar y aislar los ciberataques. La detección de los ataques en los sis-

temas de control se ha podido desarrollar usando observadores de estado, como por ejemplo el Observador de Luenberger (LO, Luenberger Observer). Mientras que la etapa de aislamiento se lleva a cabo con residuos estructurados que se pueden generar usando Observadores de entrada desconocidas (UIOs, Unknown Inputs Observer). A continuación se prensentan el diseño de estos sistemas.

3.2.1.1 Detección del ataque usando LO

Este observador permite obtener los estados de un sistema teniendo en cuenta las entradas y las salidas. El modelo general de este observador de estados se muestra en las Ecuaciones (8) y (9).

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k)) \tag{8}$$

$$\hat{y}(k) = C\hat{x}(k) \tag{9}$$

Este modelo permite estimar los estados $\hat{x}(k)$ y las salidas del proceso $\hat{y}(k)$, conociendo de antemano la señal de entrada $u(k)$ y la salida $y(k)$. Determina una fiel representación de lo que ocurre en la planta, en donde la salida estimada es muy cercana a la salida real del proceso, cuando las señales de medida no están comprometidas. El observador es diseñado de tal forma que los autovalores se ubiquen dentro del círculo unitario, para garantizar la estabilidad. Para detectar la anomalía entre el estimador y la planta real se puede generar el residuo como se muestra en la Ecuación (10).

$$res(k) = y(k) - \hat{y}(k) \tag{10}$$

Esta señal tendrá el valor cercano a 0 (o bien por debajo de un umbral) cuanto el sistema esté libre de ataque, y tendrá un valor lejos de 0 (por encima de un umbral) cuando ocurra lo contrario. De este modo el residual usado para la detección se debe comparar con un umbral como se observar en (11), de modo que si :

$$|res(k)| > \tau_{thresholds\ detection} \tag{11}$$

Entonces el sistema detectará que esta siendo atacado.

3.2.1.2 Aislamiento del ataque usando UIOs

La detección de anomalía sola da información acerca de la existencia de que el sistema se encuentra bajo ataque, pero no necesariamente, determina la ubicación o el origen del ciberataque. Para identificar las partes comprometidas del sistema se requiere de un banco de observadores de entrada desconocida que permitirán generar residuos

sensibles a una parte en particular e insensible a las otras partes del sistema, en donde se evaluará con otro umbral, para poder aislar el ciberataque.

Estos observadores son una generalización del LO, pero teniendo en cuenta de que hay entradas desconocidas al sistema. Su modelo se muestra en las Ecuaciones (12), (13) y (14).

$$w(k+1) = Fw(k) + TBu(k) + K_{12}y(k) \tag{12}$$

$$\tilde{x}^u k = w(k) + Hy(k) \tag{13}$$

$$\tilde{y}^u(k) = C\tilde{x}^u(k) \tag{14}$$

Las matrices F, T, H y K_{12}, son diseñadas de tal forma que las entradas desconocidas estén desacopladas de las otras entradas. Este observador permite reconstruir los estados del sistema o proceso, ignorando entradas. Para que esto se cumpla se debe solucionar el conjunto de Ecuaciones (15), (16), (17), (18), (19). La matriz F_d, es una matriz que es sensible a un ataque en específico e insensible a los otros ciberataques.

$$(HC - I)F_d = 0 \tag{15}$$

$$T = I - HC \tag{16}$$

$$F = TA - K_1C \tag{17}$$

$$K_2 = FH \tag{18}$$

$$K_{12} = K_1 + K_2 \tag{19}$$

La matriz K_1, se calcula del mismo modo que la ganancia L del LO. La matriz C es la matriz de salida correspondiente únicamente a una salida. Esto implica que el banco de observadores usa de manera independiente cada salida para poder estimar los estados. Esto hace que el aislamiento sea posible, y una vez más usando una señal residual como se muestra en la Ecuación (20).

$$res(k) = y(k) - \tilde{y}^u(k) \tag{20}$$

Y una evaluación de este, usando un determinado umbral para el aislamiento (Ecuación (21), se tiene que si:

$$|res(k)| > \tau_{thresholds\ isolation} \tag{21}$$

Ubica la salida comprometida. Así cada observador UIOs es diseñado para cada salida. Para poder llevar a cabo el diseño de estos observadores se debe garantizar dos condiciones:

• Que el rango de la matriz CF_d sea igual al rango de la matriz F_d.

• Que el par (C_c,A_1) sea detectable, donde $A_1 = F + K_1C$.

Si estas condiciones son satisfechas permite que el vector residual sea sensible solo a una determinada ubicación, logrando poder ubicar el origen del ciberataque.

Teniendo en cuenta esto se pueden optar por métodos analíticos que permiten encontrar la magnitud de estas eventualidades y diseñar sistemas que toleren las mismas [42]. En la literatura, se han aprovechado de estos métodos que en última se basan en modelos que permiten detectar y ubicar el sensor/acción de control atacada. Sin embargo, dado que estos sistemas están compuestos de diversas partes que en ocasiones es muy difícil de modelar su comportamiento dinámico, se pueden obtener discrepancias entre un entorno real y un modelo matemático llevado en una simulación, debido a la falta del conocimiento exacto de los diferentes parámetros que se encuentran en los sistemas [43].

3.2.2 Máquinas de aprendizaje automático

En los recientes años, métodos basados en datos han sido empleados para poder detectar ciberataques [27, 44–49]. Estos métodos han permitido representar relaciones entre las variables de entrada y salida sin la necesidad de modelar el sistema.

La tecnología de aprendizaje automático es uno de los métodos basados en datos, que está emergiendo como un método de detección de ataques dentro de estos sistemas. En el aprendizaje automático, las características importantes se procesan manualmente y existe una limitación para aprender un sistema relativamente complejo [50].

Teniendo esto en cuenta se pretende exponer las teorías relacionados con este método. El aprendizaje automático, se pueden ver desde dos perspectivas, el clásico y moderno. El enfoque de esta propuesta usa el aprendizaje automático moderno por lo tanto a continuación se describe la teoría relacionada con este.

3.2.2.1 Aprendizaje automático moderno

Estos métodos de aprendizaje se pueden agrupar en tres grandes grupos, aprendizaje por refuerzo, métodos de ensamble y redes neuronales y aprendizaje profundo. Del mismo modo, estos grupos, presentan diversos algoritmos, algunos de ellos se muestran en la Fig. 2.

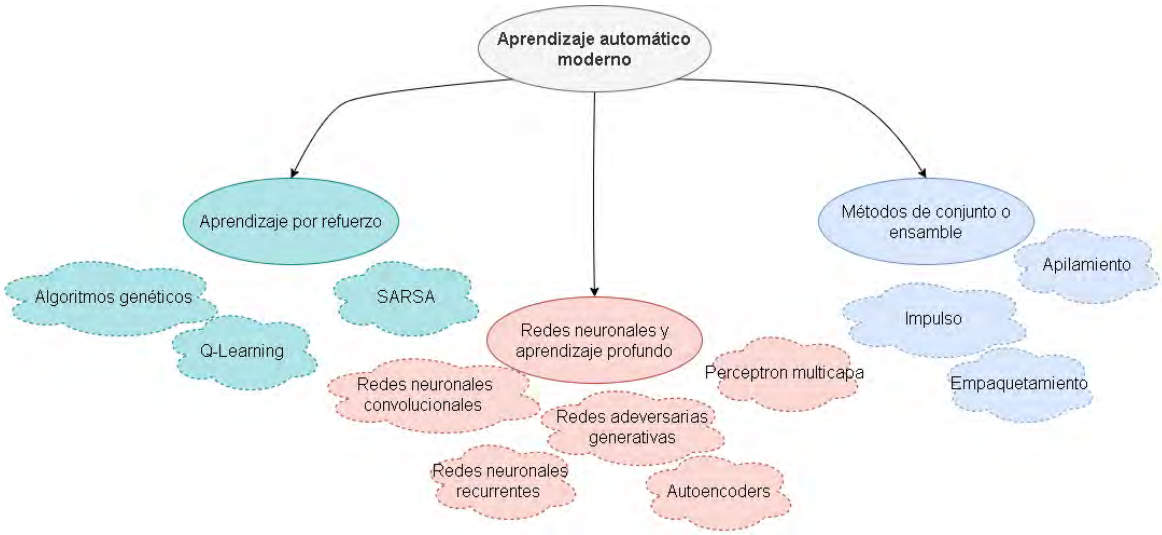


Fig. 2. Aprendizaje automático moderno.

Por un lado se tiene el aprendizaje por refuerzo (RL, Reinforcement Learning por sus siglas en inglés), el cual se refiere a la capacidad de poder aprender a través de la exploración y de este modo obtener experiencias de la práctica, frente a objetos desconocido [51]. El aprendizaje por refuerzo puede ser entendido en términos de 5 elementos: entorno, agente, estado, acción y recompensa, cómo se muestra en la Fig. 3. El entorno es todo lo que se encuentra por fuera del agente que va a aprender. El agente aprende a mapear los estados a acciones y al llevar a cabo estas acciones observa su efecto en el entorno. El estado es la representación interna del agente de una situación en particular o configuración del entorno, tal como este lo percibe. Una acción permite cambiar algún atributo o aspecto del entorno para producir un nuevo estado. Un agente en estado S_t puede seleccionar y realizar una acción A_t que cambia el entorno a un nuevo estado $S_{(t+1)}$ y produce una recompensa $R_{(t+1)}$. La recompensa es una respuesta externa del entorno e indica si la acción efectuada fue positiva o negativa con respecto al objetivo del agente. De este modo el agente busca seleccionar acciones que permitan maximizar su recompensa acumulada a largo plazo [52].

Los métodos de ensamble o de aprendizaje en conjunto se compone de varias bases que se originan principalmente a partir de datos de formación por algún algoritmo de aprendizaje clásico [18].

Por último, se tienen todo lo relacionado con las redes neuronales y aprendizaje profundo. Las redes neuronales artificiales (ANN, Artificial Neural Networks por sus siglas en inglés), es básicamente una cantidad de “neuronas” y “conexiones” entre las mismas. Una neurona no es más que una función que tiene varias entradas y una salida. Su fun-

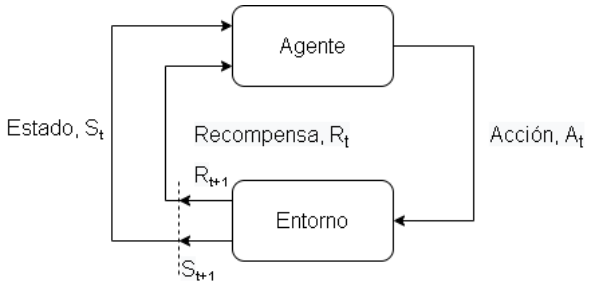


Fig. 3. Aprendizaje por refuerzo.

ción es tomar las variables de entrada, realizar una serie de operaciones entre estas y así poder obtener un resultado como salida. Por otro lado, la conexión se refiere a los canales que permiten comunicación entre las diferentes neuronas. De este modo las salidas de una neurona son las entradas de otra. Cada conexión tiene un parámetro que se denomina peso. Este peso es como una fuerza que permite establecer que tan fuerte o débil es la conexión para una señal. Estos pesos, son las relaciones sinápticas entre las neuronas, que permite de esta forma que la neurona responda más frente a una entrada y menos a otras. Estos pesos, se van ajustando en el periodo de entrenamiento, y a lo largo de millones de épocas, se logra que la red aprenda.

3.2.2.2. Perceptron de múltiples capas

El aprendizaje profundo se refiere a entrenar y probar múltiples capas de ANN, que son capaces de aprender complejas estructuras y lograr así un alto nivel de abstracción. Dentro de este ámbito se pueden encontrar diferentes arquitecturas. La más simple de todas es las que se conocen con el nombre de perceptron de múltiples capas (MLP, Multiple Layer Perceptron por sus siglas en inglés). La cual consta de una capa de entrada, múltiples capas ocultas y una capa de salida [53, 54]. La arquitectura básica se observa en la Fig. 4.

Cada entrada x_i se le asocia un peso w_i. El término w_0 es el bias. La suma de todas las entradas ponderadas $x_i w_i$ ingresan a una función de activación no lineal f que la transforma de este modo a la salida y_j. De este modo, la ecuación de salida de la neurona j viene dada por la Ecuación (22).

$$y_j = f\left(w_0 + \sum_{i=1}^{n} x_i w_i\right) \tag{22}$$

Existen muchas funciones de activación, la más común es una función rectificadora, donde las neuronas usan la función de unidad lineal rectificadora (ReLU, Rectified Linear Unit por sus siglas en inglés). También se encuentran la función tangente hiperbólica,

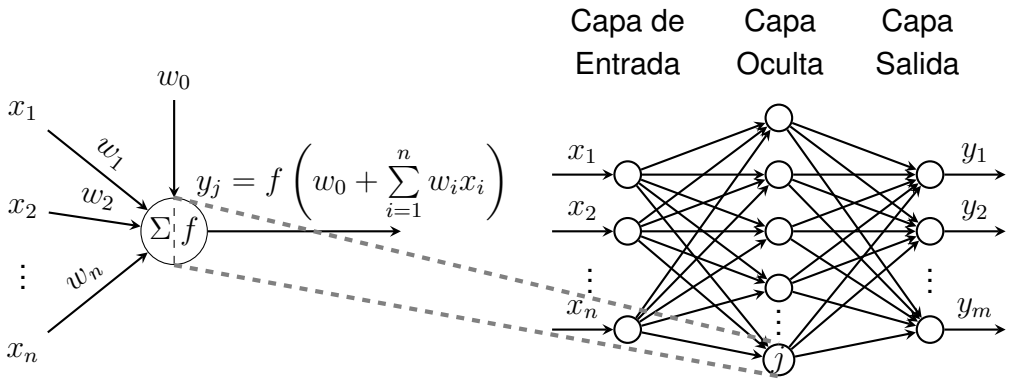


Fig. 4. Arquitectura del MLP.

la función sigmoidal y la función softmax o función exponencial normalizada. Lo más común es que la capa de salida tenga la función softmax, para problemas de clasificación y la función ReLU para problemas de regresión. Estas funciones son descritas en las Ecuaciones (23) y (24).

$$R(z) = max\,(0,z) \tag{23}$$

$$\sigma : \Re^K \longrightarrow [0,1]^K$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \; \forall \; j = 1,...,K \tag{24}$$

La salida y_j, corresponde a la salida de la neurona j de la primera capa oculta. Así las salidas de esta primera capa serán la entrada de la segunda capa, donde se realiza los mismos pasos mencionados anteriormente.

Para el entrenamiento de estos sistemas, son usados diferentes algoritmos tales como el gradiente descendente. Este algoritmo tiene como objetivo encontrar los valores de los pesos de la red que permite minimizar el error (diferencia entre las salidas estimadas y las salidas reales).

3.2.2.3. Redes Neuronales Convolucionales

Otro ejemplo de estas arquitecturas de aprendizaje profundo, son las redes neuronales convolucionales (CNN, Convolutional Neural Network por sus siglas en inglés), las cuales se han usado especialmente en aplicaciones donde se vea involucrado el procesamiento digital de imágenes. Estas toman una imagen como entrada y permiten determinar algunos aspectos característicos de las imágenes para diferenciarlas de otras. El procesamiento previo que requieren estas arquitecturas es mucho menor a otros algoritmos.

Las CNN son una construcción matemática que generalmente está compuesta de tres capas: convolución, agrupación (pooling) y capas completamente conectadas, que suelen tener una arquitectura similar al MLP. Las primeras dos, se encargan de extraer las características y la tercera, permite mapear estas características dentro de una salida, tal como una clasificación [54, 55].

La capa de convolución está compuesta por una pila de operaciones matemáticas, tales como la convolución, el cual es un operador lineal. Esta capa permite extraer características de manera óptima a partir de kernels o filtros. Esto permite reducir las dimensiones de la imagen, además de empezar a extraer propiamente características, tales como los bordes de la imagen de entrada. La operación de convolución permite que los patrones de características que se han extraído localmente sean invariantes a medida que el kernel se traslade a través de todas las posiciones de la imagen.

Cabe destacar que no es necesario limitarse a una sola capa convolucional. Desde la primera capa se busca capturar las características de bajo nivel, tales como bordes, color, orientación del degradado, entre otras. Agregar más capas, permite que la arquitectura se adapte a otras características de alto nivel, brindando una red con una muy buena comprensión. La salida proveniente de una neurona en una capa de convolución viene dada por la Ecuación (25).

$$\begin{aligned} z_{i,j,k} =& b_k + \sum_{u=0}^{f_h^{-1}} \sum_{v=0}^{f_w^{-1}} \sum_{k'=0}^{f_{n'}^{-1}} x_{i',j',k'} \cdot w_{u,v,k',k} \\ i^{'} =& i \times s_h + u \\ j^{'} =& j \times s_w + v \end{aligned} \tag{25}$$

Donde $z_{i,j,k}$ es la salida de la neurona localizada en la fila i, columna j en un mapa de características k de la capa convolucional l, s_h y s_w son el desplazamiento (stride) vertical y horizontal, f_h y f_w son la altura y ancho del campo receptivo, $f_{n'}$ es el numero de mapa de características en la capa previa (capa $l-1$), $x_{i',j',k'}$ es la salida de la neurona localizada en la capa $l-1$, fila $i^{'}$, columna $j^{'}$, y el mapa de característica $k^{'}$, b_k es el término bias para el mapa de característica k en la capa l y $w_{u,v,k',k}$ es el peso de conexión entre cualquier neurona y el mapa de característica k de la capa l y su entrada localizada en la fila u, columna v (relativo al campo receptivo de la neurona) y el mapa de característica $k^{'}$, Fig. 5.

Una de las características de este proceso de convolución es la reducción de dimensionalidad, comparada con la entrada, aunque también se pueden usar capas que permita aumentar la dimensionalidad o que permanezca igual. En el primer caso se aplican Valid Padding, o Same Padding en el otro caso. Así, el padding consiste simplemente en

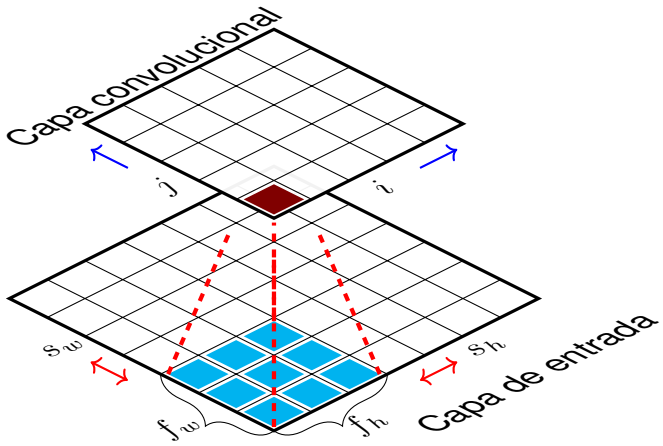


Fig. 5. Conexión entre capas.

agregar pixeles.

La capa de agrupación proporciona una operación de reducción de resolución que permite reducir la dimensionalidad de los mapas de características, para introducir una invariancia que permite reducir el número de parámetros aprendidos en capas siguientes. Dentro de estas capas se pueden encontrar capas que promedian u obtienen los valores máximos de las salidas.

Los datos provenientes de estas dos capas generalmente se transforman a una matriz unidimensional y se conectan, finalmente a una o varias capas totalmente conectadas. La capa completamente conectada normalmente tiene el mismo número de nodos de salida que el número de salidas que se requieran. Cada de una estas neuronas presentan normalmente una función tipo ReLU como función de activación. La arquitectura básica de estas redes de se presentan en la Fig. 6. A medida que una capa genera las

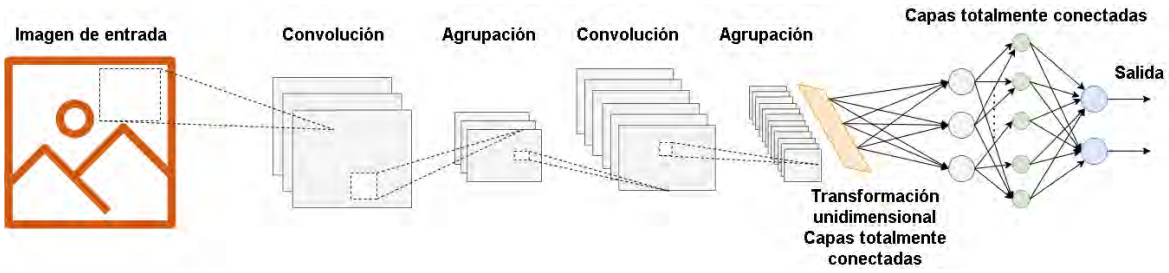


Fig. 6. Arquitectura básica de una CNN.

entradas hacia la siguiente, las características extraídas pueden volverse más complejas. El proceso de optimización de parámetros como los kernels es lo que se denomina

entrenamiento, el cual se realiza para minimizar la diferencia entre las salidas y las etiquetas verdaderas a través de un algoritmo de optimización. Entrenar una red neuronal profunda muy grande puede ser un proceso lento. Se ha visto cuatro formas de acelerar el entrenamiento (y llegar a una mejor solución): aplicar una buena estrategia para la inicialización de los pesos de conexión, usar una buena función de activación, usar normalización por lotes (Batch Normalization) y reutilizar partes de una red preentrenada (posiblemente construida en una tarea auxiliar o mediante aprendizaje no supervisado). Otro gran impulso de velocidad proviene del uso de un optimizador más rápido que el optimizador de gradiente descendente. Ejemplo de ellos son: la optimización del momento [56], Gradiente Nesterov acelerado [57], AdaGrad [58], RMSProp y finalmente Adam y Nadam mejorado [59].

3.2.2.4. Redes neuronales recurrentes

Otra arquitectura muy popular son las redes neuronales recurrentes (RNN, Recurrent Neural Network por sus siglas en inglés). Han tenido un gran impacto en el procesamiento natural del lenguaje, permitiendo generar aplicaciones tales como el reconcomiendo de voz, traducción automática y la síntesis de la misma. Están diseñadas específicamente para procesar datos de entradas temporales [60, 61].

Estas redes son una clase especial de redes neuronales caracterizadas por auto conexiones. Las RNN y sus variantes se han utilizado en muchos contextos donde la dependencia temporal de los datos es una característica implícita importante en el diseño del modelo. La arquitectura básica de estas redes de se presentan en la Fig. 7 [62].

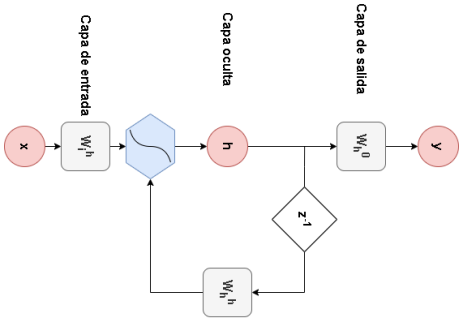


Fig. 7. Arquitectura básica de una RNN.

Estas redes son bucles que permiten que la información persista en el tiempo. Los círculos representan los nodos de entrada $\mathbf{x}$, ocultos $\mathbf{h}$ y de salida $\mathbf{y}$, respectivamente. Los bloques $\mathbf{W}_i^h$, $\mathbf{W}_h^h$ y $\mathbf{W}_h^0$, conforman las matrices que representan los pesos de entrada, ocultos y de salida respectivamente. Sus valores se sintonizan en la fase de entrenamiento a través de algún algoritmo como el gradiente descendente, por ejemplo. El

polígono representa la transformación no lineal realizada por las neuronas y z^-1, es la unidad de retardo, aunque también se pueden tener unidad de adelanto [62].

La salida de la red en el paso de tiempo t, se observa en la Ecuación (26).

$$\begin{aligned} \mathbf{h}(t) =& f\left(\mathbf{W}_i^h \mathbf{x}(t) + \mathbf{b}_i + \mathbf{W}_h^h \mathbf{h}(t-1) + \mathbf{b}_h\right) \\ \mathbf{y}(t) =& g\left(\mathbf{W}_h^o \mathbf{h}(t) + \mathbf{b}_o\right) \end{aligned} \tag{26}$$

Dentro de estas arquitecturas, una de las más conocidas son las redes de memoria a corto y largo plazo, generalmente llamadas simplemente LSTM (Long short-term memory por sus siglas en inglés), las cuales son capaces de aprender las dependencias de los datos a largo y corto plazo. Fueron introducidos por Hochreiter y Schmidhuber en 1997. Estas han sido empleadas ampliamente en el campo de procesamiento natural del lenguaje. La Fig. 8, muestra una celda de una arquitectura básica de LSTM.

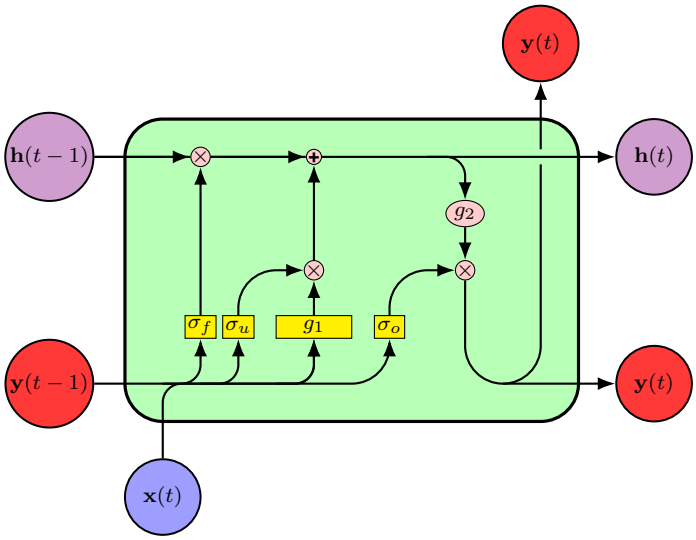


Fig. 8. Arquitectura de una celda de LSTM.

Las Ecuaciones en diferencia que definen la actualización del estado de la celda y calcular la salida son mostradas en (27).

$$\begin{aligned} \sigma_f(t) =& \sigma\left(\mathbf{W}_f \mathbf{x}(t) + +\mathbf{R}_f \mathbf{y}(t-1) + \mathbf{b}_f\right) \\ \tilde{\mathbf{h}}(t) =& g_1\left(\mathbf{W}_h \mathbf{x}(t) + \mathbf{R}_h \mathbf{y}(t-1) + \mathbf{b}_h\right) \\ \sigma_u(t) =& \sigma\left(\mathbf{W}_u \mathbf{x}(t) + +\mathbf{R}_u \mathbf{y}(t-1) + \mathbf{b}_u\right) \\ h(t) =& \sigma_u(t) \odot \tilde{\mathbf{h}}(t) + \sigma_f(t) \odot \tilde{\mathbf{h}}(t-1) \\ \sigma_o(t) =& \sigma\left(\mathbf{W}_o \mathbf{x}(t) + \mathbf{R}_o \mathbf{y}(t-1) + \mathbf{b}_o\right) \\ \mathbf{y}(t) =& \sigma_o(t) \odot \mathbf{h}(t) \end{aligned} \tag{27}$$

Donde $\sigma_f(t), \sigma_u(t)$ y $\sigma_o(t)$, representan los puntos de olvido, actualización y de salida, respectivamente, $\tilde{\mathbf{h}}(t)$ es el estado candidato, $\mathbf{h}(t)$ es el estado de celda, $\mathbf{y}(t)$ es la salida, las matrices $\mathbf{W}_f, \mathbf{W}_h, \mathbf{W}_u$ y $\mathbf{W}_o$ son los pesos aplicados a la entrada $\mathbf{x}(t)$, las matrices $\mathbf{R}_f, \mathbf{R}_h, \mathbf{R}_u$ y $\mathbf{R}_o$ definen los pesos de las conexiones recurrentes, mientras que $g_1(\cdot)$ y $g_2(\cdot)$ son las funciones de activación no lineal, usualmente del tipo tangente hiperbólica (28), $\sigma(\cdot)$ es la función sigmoidal (29). Por último $\odot$ se refiere al producto Hadamard.

$$\tanh(z) = \frac{2}{1 + e^{-2z}} - 1 \tag{28}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{29}$$

Cada punto en la celda tiene una específica y única funcionalidad. El punto de olvido decide que información podría ser descartada desde el estado de celda previo $\mathbf{h}(t-1)$. El punto de actualización opera sobre el estado previo $\mathbf{h}(t-1)$, después de haber sido modificado por el punto de olvido, y decide cuanto debe el nuevo estado $\mathbf{h}(t)$ actualizarse con un nuevo candidato $\tilde{\mathbf{h}}(t)$. Por último, el punto de salida selecciona la parte del estado que será retornado como salida. Cada punto depende de una entrada externa $\mathbf{x}(t)$ y una salida de celda previa $\mathbf{y}(t-1)$.

3.2.3 Tecnologías para abordar ciberataques en sistemas ciberfísicos

La automatización de un proceso industrial requiere la instalación de múltiples plataformas de controladores, incluidos controladores lógicos programables, sistemas de control distribuido (DCS, Distributed Control System por sus siglas en inglés), sistemas de gestión de mantenimiento computarizado (CMMS, Computerized Maintenance Management System por sus siglas en inglés), planificación de recursos empresariales (ERP, Enterprise Resource Planning por sus siglas en inglés), sistemas de monitoreo de históricos de datos, entre otros [63].

Para permitir el funcionamiento eficaz de las operaciones integradas y de fabricación se desarrolló el estándar ISA-95 [64], el cual facilita la descripción de modelos y terminología para determinar el intercambio de información entre diferentes niveles del sistema. En este se definen diferentes niveles de automatización, que van desde los procesos físicos reales en el nivel más bajo, hasta la fabricación en los niveles más altos.

En lo que respecta a las estrategias de control de procesos industriales no se han presentado cambios sustanciales en las últimas décadas. Esto se debe principalmente a que las aplicaciones de control típicas que se encuentran en los procesos industriales tienen que responder a los cambios en el mundo físico dentro de límites de tiempo predefinidos, por lo que trasladar la ejecución de las tareas de control de los dispositivos

ubicados físicamente junto con el proceso controlado a nuevas plataformas, como las plataformas de computación en la nube, requiere verificar el cumplimiento de plazos con retrasos que son difíciles de predecir. Además, si bien el hardware de control dedicado y las soluciones permiten que el diseño de control tenga plena autoridad sobre el entorno en el que se ejecutará su software, no es sencillo determinar en qué condiciones se puede ejecutar el software en plataformas de computación en la nube debido a la virtualización de recursos. Sin embargo, se esperan modificaciones en la medida en que se incorporen cada vez más a estos procesos las tecnologías de la cuarta revolución industrial [65]. Precisamente estos nuevos requisitos demandan nuevas características a los componentes principales de estas soluciones de automatización, como los PLC, los dispositivos de campo y los sistemas de control de supervisión y adquisición de datos (SCADA). Los cuales deberán volverse más flexibles, evolucionando hacia sistemas ciberfísicos reconfigurables [66].

Abordar estas soluciones en base a criterios convencionales y agregar nuevos nodos de hardware cuando se requieren nuevas funciones, demanda mayores costos de implementación, la reconfiguración de sistemas para soportar los nuevos datos transmitidos a través de la red sin afectar el cumplimiento de los requisitos en tiempo real, y dificultades en el intercambio de información entre las plataformas de diferentes fabricantes. Este contexto requiere un enfoque más flexible que agregar nuevos nodos, y es la reconfiguración de los nodos existentes de acuerdo con los requisitos de las nuevas funcionalidades del sistema.

3.2.3.1. Virtualización

Como consecuencia del incremento de recursos en los nodos hardware, tecnologías como la virtualización hacen más flexible la inclusión de nuevos componentes a muy bajos costos, lo cual es muy conveniente en cuanto a la flexibilidad e intercambio de información entre componentes. Evidentemente, como en los enfoques tradicionales, es necesario verificar el cumplimiento de los plazos en sistemas de tiempo real.

La virtualización es un método que permite incrementar la tasa de uso del hardware, dividiendo este en múltiples partes [67]. La virtualización de una plataforma se refiere a la creación de máquinas virtuales que se ejecutan sobre una misma plataforma física que es administrada por un monitor de máquina virtual (VMM, Virtual Machine Manager por sus siglas en inglés) conocido como hipervisor, el cual es el middleware entre el hardware y las aplicaciones o sistemas operativos invitados. Este tipo de tecnologías permite la ejecución concurrente de múltiples máquinas virtuales en un mismo hardware [68]. Esta tecnología es ampliamente usada en infraestructuras de servidores y computación

en la nube. Permite el soporte de múltiples ejecuciones en diversos entornos de manera simultánea asignando dinámicamente recursos de hardware a cada uno de estos ambientes. En este contexto esta técnica ofrece la siguiente serie de beneficios [69]:

- Consolidar servidores: ejecución de múltiples sistemas operativos con aislamiento temporal y espacial sobre una misma plataforma hardware.

- Sistemas operativos heterogéneos: brinda soporte para la ejecución de diversos sistemas operativos en una misma plataforma.

- Asignación dinámica de recursos: se pueden agregar o quitar recursos dinámicamente entre las aplicaciones en función de los requisitos que se vayan presentando.

Varios aspectos clave motivan una exploración de los beneficios prometidos por la virtualización en los sistemas de automatización industrial. Por ejemplo, una gran cantidad de unidades de hardware interconectadas es costosa y propensa a errores, pero exhiben una fuerte independencia, las tecnologías de virtualización brindan independencia y hardware rentable al mismo tiempo; por otro lado, el tiempo de vida muy prolongado de las instalaciones de la planta requiere la integración de nuevas soluciones a medida que el hardware heredado deja de estar disponible y la virtualización puede ayudar en este sentido [63].

La tecnología de virtualización mejora la escalabilidad del sistema y se pueden llegar a tener ahorros en costos de implementación, permitiendo un uso más eficiente del hardware físico además de que contiene varias innovaciones en seguridad. Aplicaciones de software como antivirus y firewall, instalado en los servidores para garantizar la seguridad del sistema provoca una pérdida de rendimiento en la red y los servidores. Por eso, las aplicaciones de seguridad deben implementarse en la capa de virtualización [67].

Además, el uso de procesadores multicore y algunas técnicas de virtualización en los sistemas integrados permiten introducir estructuras jerárquicas de programación, en donde se requiere la garantía de comportamiento en tiempo real tanto a nivel de las aplicaciones invitadas así como a nivel del hipervisor [69].

Un hipervisor, por ejemplo, permite aislar las aplicaciones en particiones software cada una de ellas con su propio sistema operativo. El hipervisor es el responsable de la ejecución de las aplicaciones y garantiza el aislamiento espacial y temporal de estas, así como la contención y gestión de fallos.

El aislamiento temporal consiste en que las aplicaciones se ejecuten de forma independiente de manera que el tiempo de ejecución de una no afecte a las otras. Por ejemplo, si

una aplicación por un fallo no abandona la CPU (Central Processing Unit por sus siglas en inglés), otras aplicaciones no se ejecutarían. Esto permite que una partición virtual no pueda afectar la capacidad de las otras particiones para acceder a un recurso compartido [70]. El hipervisor garantiza el aislamiento temporal mediante la planificación de las particiones. Por ejemplo, una política como la cíclica en la que a cada aplicación se le reserva una ventana temporal para la ejecución y solo allí es ejecutada, garantiza que la aplicación no consuma más CPU que la especificada.

Mientras que el aislamiento espacial ofrecido por el hipervisor permite que una aplicación no pueda acceder de ninguna manera a la zona de memoria de otras aplicaciones. Es decir que las aplicaciones y/o los datos privados de las particiones no pueden modificarse desde otra partición. El aislamiento espacial se logra si cada partición tiene su propio espacio de direcciones en la memoria compartida. El mecanismo básico en la gestión de la memoria es mediante la unidad de gestión de memoria (MMU, Memory Management Unit por sus siglas en inglés), para lograr traducir direcciones virtuales de las particiones virtuales a una dirección de memoria. Estos gestores protegen el espacio de direcciones asignado de una partición por violaciones de las demás. Cuando se ejecuta una aplicación, el hipervisor se encarga de llenar la MMU con los rangos de memoria de cada aplicación [70].

El hipervisor usa un modelo de gestión de fallos, que permite detectar y manejar las fallas o eventos inoportunos que se puedan presentar en las particiones, posibilitando de esta manera que los fallos de una partición sean gestionados directamente por el hipervisor haciendo que no afecten a otras aplicaciones. Así, el propósito de la gestión de fallos en el hipervisor se encarga de descubrir e identificar los errores en una etapa temprana para tratar de resolverlos o limitar el subsistema que presenta el fallo evitando o reduciendo de este modo las posibles consecuencias que este puede generar [71].

Dentro de este campo se destacan dos tipos de virtualización: full virtualización y para virtualización.

❖Full virtualización

También conocida como virtualización completa o bare metal. En esta técnica el sistema operativo o aplicación invitada no tiene conocimiento de que se ejecuta en un ambiente virtualizado. La ventaja de este enfoque es que permite el desacoplamiento completo de software y hardware, proporcionando un aislamiento total entre las diferentes máquinas virtuales. Todas las instrucciones sensibles a la CPU deben ser privilegiadas para evitar que otras máquinas virtuales realicen ajustes de hardware, lo cual conlleva a una pérdida de rendimiento, dado que las instrucciones deben ser emuladas. La arquitectura de este

tipo de virtualización es mostrada en la Fig. 9 [72].

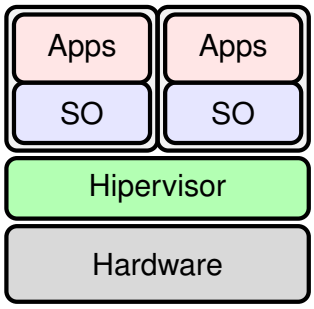


Fig. 9. Arquitectura Full Virtualización.

❖ **Para virtualización**

Esta técnica requiere de modificación al sistema operativo invitado, lo que conlleva a que el invitado es consciente de que es ejecutado en un sistema virtual. El rendimiento del sistema operativo es cercano al nativo. Se usan Hiperllamadas para solicitar servicios al hipervisor. La para virtualización ofrece ventajas en términos de eficiencia y flexibilidad en tiempos de ejecución, aunque el principal inconveniente es la necesidad de la creación de un puerto al sistema operativo invitado, lo cual impide modificaciones en partes críticas del núcleo. Aun así existen otros problemas relacionados con la virtualización en sistemas embebidos multicore, en donde temas como el apoyo de las interfaces E/S y otras tareas dependientes requieren de estudios de planificabilidad [69]. La arquitectura de este tipo de virtualización es mostrada en la Fig. 10.

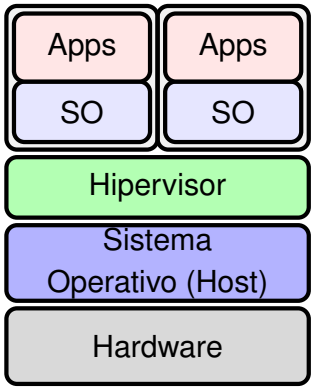


Fig. 10. Arquitectura Para Virtualización.

❖ **Contenedores**

Otra tecnología usada en procesos de virtualización, son los contenedores los cuales son relativamente nuevos. Permite facilitar la estandarización, el intercambio, el desarrollo, la implementación y la seguridad de las aplicaciones y módulos. Los contenedores y

las máquinas virtuales, proporcionan una asignación de recursos y unos beneficios de aislamiento similares. Sin embargo, dado a la arquitectura de los contenedores estos permiten ser mucho más portátiles y eficientes dado que requieren menos recursos. La arquitectura se puede observar en la Fig. 11.

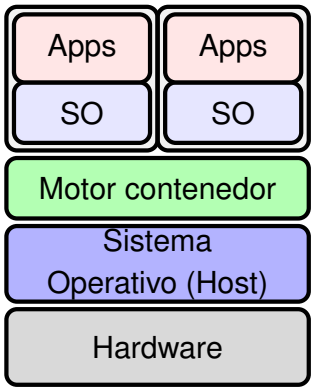


Fig. 11. Arquitectura de un Contenedor.

El concepto de arquitecturas basadas en contenedores para procesos de control automático, debe cumplir tanto con los requisitos de los controles industriales, como la capacidad en tiempo real y la alta disponibilidad, así como la requisitos que acompañan a la descomposición de un arquitectura de software monolítica en servicios [73].

Los pequeños componentes de software han surgido recientemente como una gran posibilidad para atacar problemas complejos con una interconexión de pequeñas funcionalidades. Así es como las arquitecturas de software basadas en microservicios se han utilizado recientemente de manera significativa para resolver una gran variedad de problemas de complejidad de software que deben adaptarse a los requisitos cambiantes.

Un microservicio se considera como una pequeña aplicación que se puede implementar, escalar y probar de forma independiente, con una sola responsabilidad. Las grandes aplicaciones desarrolladas con el enfoque monolítico se pueden separar en mini aplicaciones aisladas más pequeñas, brindando un servicio específico, dando la ventaja o facilidad de modificar los servicios que se requieren sin afectar otros servicios. Por lo tanto, el microservicio puede reemplazar la arquitectura monolítica, lo que permite el desarrollo de aplicaciones de servicio distribuidas, ligeras, desacopladas e independientes que trabajan juntas. Entonces, debido a que los servicios en esta arquitectura se pueden implementar por separado en diferentes nodos o procesos, puede haber una sobrecarga de comunicación entre ellos, afectando el rendimiento del sistema distribuido, debido a la comunicación de red de alta latencia, por lo tanto, es necesario proporcionar una protocolo eficaz de comunicación en red.

Aunque ya existen en el mercado varias soluciones de virtualización disponibles comercialmente, tanto para servidor (p. Ej., ESXi, XenServer) como integradas en tiempo real (p. Ej., Integrity Multivisor, Wind River Hypervisor), el despliegue de virtualización en el dominio de la automatización industrial es algo reciente, por lo que aún existen desafíos y oportunidades para implementar tecnologías de virtualización en procesos industriales, algunos de los cuales son [63, 66, 74]:

• Soporte heredado: la mayoría de los sistemas de control industrial tienen requisitos muy estrictos de disponibilidad; por lo tanto, cualquier operación que requiera desconectar el sistema debe reducirse al mínimo. En consecuencia, la capacidad de realizar mantenimiento y actualizar o reemplazar sensores y manipuladores sin ningún tiempo de inactividad será valiosa. Un desafío de la sostenibilidad de la arquitectura es poder admitir actualizaciones en tiempo de ejecución.

Por lo tanto, mientras se cambia a nuevas arquitecturas de control, también se debe ofrecer soporte para aplicaciones heredadas, incluidas estrategias de migración. Si bien existen varios enfoques que hacen evolucionar las arquitecturas de control al siguiente nivel, ninguno de ellos aborda la implementación de funciones flexibles al mismo tiempo con el soporte heredado. Uno de los principales desafíos para migrar el código de control heredado es replicar el comportamiento del controlador lo más cercano posible. Por lo tanto, transformar el código de control en el código del lenguaje de un nuevo controlador a menudo no es una solución deseable. Los desafíos generales del ciclo de vida de los sistemas de automatización y las posibles estrategias de mitigación se abordan mediante actividades de estandarización en curso, pero todavía no existe una solución genérica para integrar aplicaciones de control heredadas en nuevos diseños de controladores.

• Sensores y actuadores inteligentes. Hoy en día, a medida que los procesadores y la memoria se vuelven más baratos, ha habido un aumento en la demanda de funciones que se incorporen dentro de los dispositivos de campo para responsabilidades adicionales de monitoreo y control. Esto plantea el desafío de manejar la complejidad de la lógica de control distribuida. El software de lógica de control se distribuye más en una gran cantidad de nodos, en consecuencia la arquitectura del software debe manejar esta complejidad.

• Mayor conectividad. Los sectores industriales están experimentando un cambio hacia operaciones y sistemas más integrados. Dependiendo del tipo y propósito de los sistemas de automatización estas múltiples plataformas de controlador se pueden distribuir a escala local, de área amplia o incluso global, lo que da como resultado una mayor interconexión entre los diferentes sistemas. Esta mayor conectividad está impulsada por la proliferación y la reducción constante de los costes asociados al uso de las conexio-

nes inalámbricas y de Internet, lo que hace que la información sea accesible a través de fronteras geográficas de forma rentable.

• Seguridad de la red: los sistemas de control industrial a menudo asumen que la red en la que operan es (físicamente) segura / separada de la intervención externa. Además, se supone que muchos sistemas de control industrial se ponen en servicio una vez y rara vez tienen que modificarse durante su ciclo de vida. Sin embargo, con la aparición de Internet el supuesto de aislamiento en la realidad no siempre se cumple. El desarrollo constante de nuevos vectores de ataque conduce a numerosos problemas de seguridad. Por lo tanto, para garantizar operaciones seguras la arquitectura del software debe evitar que nuevos vectores de ataque provenientes de los canales de comunicación ingresen al sistema. Además, la arquitectura del software también debe permitir actualizaciones de seguridad sin interrupciones en el proceso de control.

La virtualización se puede utilizar para aislar el software del controlador del software que necesita conectarse a una red externa. La interacción necesaria entre el software conectado y el software de control puede estar bajo un estricto control mediante el uso de firewalls virtuales y sistemas de detección de intrusos. Toda la red subyacente a la pila de redes se puede virtualizar e implementar de forma independiente del dominio.

• Previsibilidad de las latencias de la red: debido al uso de conexiones a Internet y conexiones inalámbricas, la suposición sobre la previsibilidad de las latencias de la red del sistema de control industrial en el entorno de la red de área local (LAN, por las siglas en inglés Local Area Network) ha cambiado. Por lo tanto, desde la perspectiva del impacto de la sostenibilidad, garantizar la provisión de garantías de tiempo para controlar el software lógico se vuelve esencial.

• Cambio hacia el hardware básico. Hay un movimiento desde el hardware especializado, en el que una plataforma de hardware se diseña y construye para un propósito específico, hacia el hardware básico. El hardware básico es más asequible, a diferencia del hardware de propósito especial, por ejemplo, servidores especializados con memoria o CPU enormemente alta. A medida que el hardware básico proporciona una cantidad promedio de recursos informáticos, la industria está comenzando a desarrollar y ofrecer funcionalidad en software utilizando hardware básico, en lugar de invertir en costoso hardware de alta gama. A pesar de las ventajas de costo del hardware básico también existen desafíos que se deben enfrentar, los cuales incluyen:

◦ CPU de varios núcleos: la disponibilidad de procesadores con varios núcleos aumenta el rendimiento computacional, pero puede conducir a una mayor complejidad. Además, la mayor parte del software heredado está escrito para ejecutarse en un solo núcleo. Por

lo tanto, la arquitectura y la práctica del diseño de software deben utilizar de manera eficiente este hardware paralelo. Es posible consolidar el hardware moviendo el software de varios dispositivos a uno, por ejemplo pasar controladores de un solo núcleo a un solo controlador de varios núcleos. Esto se puede hacer virtualizando cada controlador de núcleo único, asignando el software heredado a una máquina virtual. Esto no solo reducirá la cantidad de unidades de hardware, sino que también reducirá el cableado entre las unidades, los cables de alimentación y el espacio en el piso. Al utilizar la virtualización integrada en tiempo real el software del dispositivo existente se puede presentar con una vista del hardware que se asemeja mucho al hardware heredado; por lo tanto, la virtualización permite utilizar el software existente con pocos cambios o ninguno. Además, la virtualización integrada proporciona medios para una independencia estricta entre diferentes aplicaciones de software, lo que reduce el impacto en el software de dispositivos sensibles a los recursos, por ejemplo, los controladores.

- Algunas soluciones de redundancia basadas en hardware, por ejemplo CPU sincronizadas duplicadas por seguridad, no son compatibles con hardware básico. Esto a su vez crea responsabilidades adicionales para la arquitectura del software. Las soluciones de servidor disponibles en la actualidad para la virtualización de servidores proporcionan mecanismos de software para alta disponibilidad y tolerancia a fallas. Esto se logra mediante la replicación automática de una máquina virtual (VM) en una máquina física secundaria, manteniendo los estados de la VM principal y la VM de réplica sincronizados en tiempo real. Esta es una solución basada en software y hardware básico que imita la solución de hardware especializado.

- Tecnologías mixtas. La mayor parte del legado en los dispositivos y sistemas de control que se encuentran en el nivel 1 y el nivel 2 de la pirámide de automatización generalmente se ocupan de requisitos estrictos sobre el comportamiento de sincronización del software, hasta milisegundos para los sistemas de control distribuidos y microsegundos para otros, como el control de movimiento. Para cumplir con los requisitos del control industrial se utiliza middleware especializado, por ejemplo sistemas operativos en tiempo real. Esto hace que a veces sea difícil adoptar tecnologías de otros dominios directamente en el dominio de la automatización industrial, ya que no cumplen con los requisitos de los sistemas de control ni brindan soporte heredado. Un ejemplo concreto es el uso de tecnologías de software basadas en Windows para construir una HMI para un controlador.

Con la virtualización, es posible mezclar diferentes tecnologías en el mismo hardware con un impacto controlado en el comportamiento. En el ejemplo anterior, la HMI se administra en una máquina virtual que ejecuta un sistema operativo invitado de Windows, sin que el software del dispositivo (heredado) y el sistema operativo en tiempo real se

vean afectados. De otro lado, el uso de la virtualización para administrar diferentes tecnologías también se puede utilizar para una transición controlada a una nueva plataforma tecnológica, por ejemplo, una migración paso a paso desde un sistema operativo de un solo procesador en tiempo real a Linux en tiempo real con soporte multinúcleo. Esto permite ejecutar tanto la plataforma heredada como la nueva plataforma en paralelo.

• Criticidad mixta. El cambio de hardware personalizado a hardware básico conlleva responsabilidades adicionales a la arquitectura del software. Desde la perspectiva de la seguridad, puede resultar difícil utilizar tecnologías de software (como los sistemas operativos) que se desarrollan para fines generales en un contexto de seguridad, como por ejemplo la funcionalidad de protección que pone el sistema en un estado seguro en caso de error.

Cuando se integran diferentes aplicaciones de software en un hardware común, los estándares de seguridad como IEC 61508 requieren que todo el software sea tratado como de máxima criticidad, a menos que se pueda demostrar una independencia estricta. Por lo tanto, sin independencia entre diferentes aplicaciones de software, la integración de una aplicación crítica para la seguridad y una aplicación de control heredada requiere la certificación de seguridad del software heredado, que normalmente es económicamente inviable para cualquier aplicación heredada más grande.

• Emulación de hardware heredado. Muchas instalaciones de sistemas de control industrial tienen una vida útil muy larga, a veces de 20 a 30 años. Durante esta larga vida útil, el hardware puede volverse obsoleto y necesitar ser reemplazado. De lo contrario, será cada vez más difícil y costoso con la edad a medida que los componentes de hardware se acerquen al final de su vida útil. Además, durante el ciclo de vida de los sistemas, las actualizaciones de hardware pueden generar problemas de incompatibilidad que requieran una costosa reescritura del software integrado. La tecnología de virtualización y emulación es una posibilidad atractiva para mantener el software heredado intacto mientras se reemplaza el hardware heredado físico con contrapartes virtuales que se ejecutan en hardware moderno.

Las aplicaciones que se ejecutan dentro de los contenedores están aisladas unas de otras porque no pueden ver ni acceder a los recursos de las demás. Además, el anfitrión del contenedor puede limitar la cantidad de recursos utilizados por un contenedor individual, por ejemplo, CPU, memoria, E/S de disco y red. Comparado con la virtualización basada en hipervisor, los contenedores tienen una sobrecarga más baja (ya que el kernel del sistema operativo es compartido). A diferencia de los enfoques de hipervisor, el huésped y el host deben compartir el mismo kernel de sistema operativo. La virtualización basada en contenedores, al no requerir un hipervisor, proporciona un rendimiento

casi nativo, tiempos de implementación rápidos y una sobrecarga baja al tiempo que conserva un cierto nivel de recursos y aislamiento control. A pesar de las ventajas y posibilidades descritas, los dominios industriales a menudo requieren un comportamiento en tiempo real, y estas capacidades aún no son totalmente compatibles con la virtualización basada en contenedores. Un host puede acomodar varios contenedores a la vez, proporcionando medios para el aislamiento de contenedores y el control de recursos para los contenedores [66, 74].

En Linux, el aislamiento de aplicaciones se implementa mediante una función del kernel llamada espacios de nombres, la gestión de recursos por parte de cgroups. Dado que los contenedores se inician a partir de imágenes de contenedor (similar a cómo los invitados del hipervisor pueden comenzar a partir de imágenes de disco), los sistemas de archivos que admiten la copia en escritura pueden ayudar a que la creación y administración de contenedores sea muy eficiente. Sobre la base de las características del kernel anteriores, Docker y LXC son algunas implementaciones populares de herramientas y API para crear y administrar contenedores en Linux. El sistema operativo proporciona interfaces para administrar, limitar y asignar el acceso a los recursos del sistema (CPU, memoria, etc.) así como a sus dispositivos conectados (red, E/S, etc.). Los contenedores que se ejecutan en el controlador se pueden agregar, quitar, iniciar y detener dinámicamente según las necesidades del sistema en general. Además, la asignación de recursos (por ejemplo, cambiar la distribución de recursos compartidos de CPU entre contenedores) se puede cambiar dinámicamente durante el tiempo de ejecución del sistema.

El objetivo de los contenedores es envolver microservicios simples que funcionan en red, proporcionando un enfoque modular para crear aplicaciones [75].

3.3 ANTECEDENTES

Los CPSs, han tenido un gran impacto en varios sistemas de nuestra vida diaria (sistemas de generación y distribución de energía eléctrica, distribución de gas natural, sistemas de transporte, dispositivos de salud, entre otros) [1]. Sin embargo, los sistemas de control heredados no fueron diseñados para enfrentar los niveles de amenaza actuales, debido a que los nuevos enlaces de comunicación pueden ser utilizados por personas para generar ciberataques que ponen en riesgo la funcionalidad de estas aplicaciones. Estos ataques informáticos en ambientes industriales pueden poner en riesgo personas y equipos, afectar la producción o la calidad de los productos, y en general generar diversas afectaciones críticas.

Para proteger estos sistemas se ha optado por utilizar estrategias que han presentado buenos resultados en otros ambientes, como por ejemplo los ambientes de oficina. Sin

embargo, las características de estas aplicaciones no son las mismas y los resultados alcanzados no son los esperados [76], obteniendo ciertas dificultades relacionadas con:

- El software y el hardware de los entornos de oficina suelen estar más actualizados, ya que tienen ciclos de vida de reabastecimiento de tecnología más cortos.

- Los clientes de automatización industrial exigen un soporte de 20 años, obviamente mucho más largo que la vida útil de 3-5 años de los sistemas en ambientes de oficina. Esto crea un entorno industrial, con equipos y firmware, que no pueden ser compatibles con las prácticas estándar de los ambientes corporativos, tales como parches (los sistemas operativos obsoletos simplemente ya no se pueden parchear, porque no existen parches para ellos).

- La disponibilidad de los equipos en sistemas industriales es muy alta; por lo que en muchos casos una solución sencilla de los entornos corporativos como el parcheo, simplemente no funciona porque la máquina no está disponible para apagarse hasta una interrupción planificada. También es difícil predecir cómo un parche recién introducido afectará el funcionamiento de un sistema de control, especialmente si el parche no se prueba rigurosamente, lo que aumenta la renuencia de la organización a actuar ante posibles amenazas.

- La implementación de parches de seguridad puede afectar el rendimiento de la aplicación, y por tanto la estabilidad, la disponibilidad y el comportamiento en tiempo real de las máquinas. Algo equivalente ocurre con el impacto en el tráfico de datos a través de la red de comunicaciones asociado a soluciones que evalúan el tráfico en la red, lo cual puede afectar los retrasos en las estrategias de control y a su vez el desempeño de los lazos de control.

Por otro lado, la rigidez de las aplicaciones en los entornos industriales (aplicaciones que modifican poco su comportamiento y realizan un envío de mensajes periódico), facilitan la detección de anomalías respecto al comportamiento de las aplicaciones en entornos de oficina. En general estas estrategias prácticas para abordar la problemática de seguridad se pueden agrupar en:

- Utilización de productos certificados y probados.

- Implementación de arquitecturas de sistemas que establezcan zonas seguras. Aíslan el tráfico, restringen al acceso y analizan el tráfico para detectar anomalías.

- Implementación de estrategias y desarrollo de hábitos de protección a todos los niveles,

desde el nivel operativo hasta el campo, aplicando estándares para la seguridad en la automatización industrial como el ISA99 / IEC 62443. Esto consiste en la definición e implementación de políticas y procedimientos que cubren la evaluación de riesgos, la mitigación de riesgos y los métodos para recuperarse de un desastre.

Los métodos que garantizan la seguridad en la información en estos sistemas se han orientado principalmente en la autenticación, control de acceso o asegurar la integridad del mensaje. Sin embargo, este enfoque deja varios desafíos sin abordar. En algunos casos los datos no son confidenciales y se requiere solamente verificaciones del origen de la información, mientras que en otros casos mantener la confidencialidad es importante; además algunos sistemas son críticos en el tiempo y por lo tanto los atacantes influyen en procesos de sincronización y retrasos en los lazos de control [77]. De este modo se requiere un avance en los enfoques de seguridad, en donde las herramientas clásicas de ciberseguridad son ineficaces en estos escenarios.

Aun así, casos de ataques y vulnerabilidades se han reportado [78–82]. Empresas cómo Siemens han reportado listas de avisos de algunas vulnerabilidades. Estos incidentes muestran que los controles preventivos de seguridad, como las zonas desmilitarizadas tradicionales, la fuerte segregación de red y los múltiples firewalls, no siempre son suficientes para proteger los equipos en los sistemas de control industrial. Por lo que los esfuerzos no solo deben ponerse en la prevención de ataques sino también en la detección y corrección de ataques.

En la Fig. 12, se muestra una línea temporal de los principales incidentes que han ocurrido en estos sistemas, relacionados con ciberataques [21].

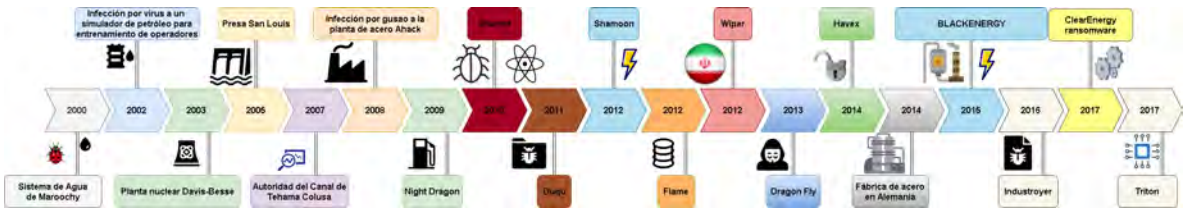


Fig. 12. Principales ataques a los sistemas de control industrial.

En el año 2000 Maroochy Shire en Queensland experimentó problemas con su nuevo sistema de aguas residuales. Una persona llamada Boden pudo controlar 150 estaciones de bombeo de aguas residuales mediante una computadora portátil y un transmisor de radio. Durante un período de tres meses, liberó un millón de litros de aguas residuales sin tratar en un desagüe de aguas pluviales desde donde fluía hacia las vías fluviales locales [83].

En el 2002 un simulador de entrenamiento para operarios en la industria del petróleo fue infectado por un virus común [84]. En enero del 2003 un gusano llamado Slammer infectó la planta de energía nuclear de Ohio [85]. En diciembre del 2005 un error de medición en un sistema de control de la presa de San Louis, Missouri causó una falla masiva en la presa. Eso causó que la presa se fracturará y la gente que estaba alrededor de un diámetro de 4 millas corrieron un riesgo. Se produjeron muchos daños económicos en las tierras agrícolas luego de este incidente [86].

En el 2007 un ex empleado de un pequeño canal de California después de ser despedido instaló un software no autorizado en el SCADA [87]. En el 2008, en Brasil un ex contratista que accedió al interior de una planta de acero, libero un gusano llamado Ahack en una planta de energía. Este virus se extendió en toda la red de automatización, provocando muchas pérdidas debido a las comunicaciones interrumpidas entre los PLC y las estaciones de monitoreo [88]. En el 2009, se llevó a cabo el ataque conocido como Night Dragon, el cual tenía como objetivo compañías petroquímicas, de gas y petróleo, donde se robaba información sensible que posteriormente se vendía .

Un año más tarde, el famoso malware conocido como Stuxnet fue detectado por Kaspersky, el cual había infectado una central nuclear en Irán. Este, modificaba diferentes parámetros de la programación del PLC, del fabricante Siemens, con la finalidad de parar los servicios de turbinas de la planta nuclear. En el 2011, surgiría Duqu, cuyo objetivo era infectar y conseguir información sensible de empresas de medio Oriente.

En el año 2012 Shamoon, Flame y Wiper serían descubiertos. El primero de ellos buscaba causar daño en el sector energético, afectando a la compañía Saudi Aramco. Es perteneciente de una familia de malware muy agresivos y altamente destructivos. El segundo, buscaba afectar las compañías de petróleo y gas de medio Oriente y Europa. El último de ellos, atacaba las empresas de energía, petróleo y gas y entidades gubernamentales en Irán [89].

En el 2013, DragonFly haría su entrada con ciber espionaje en empresas del sector eléctrico en Estados Unidos y Europa [89]. En el 2014, Havex utilizó el estándar de comunicaciones de protocolo abierto (OPC), reuniendo información. Ese mismo año, una fábrica de acero en Alemania sería atacada mediante correos electrónicos lo cual ayudo a los atacantes para obtener acceso a la red y a los sistemas de producción, su finalidad era causar daños físicos [90].

En el 2015 aparecería BLACKENERGY, el cual era un malware destructivo con una capacidad de apagar sistemas críticos, en una planta de energía en Ucrania [89]. En el 2016 Industroyer surgió, y tenía varias capacidades para explotar vulnerabilidades en

plataformas Siemens. En el 2017 ClearEnergy apreció, el cual fue un malware que se dirigió a la infraestructura crítica y sistemas SCADA de plantas nucleares, de energía e instalaciones de agua y residuos. En ese mismo año Triton/Trisis llevaría a cabo su objetivo [21].

Está problemática ha motivado varios esfuerzos que pretenden contribuir desde diferentes enfoques a aumentar la seguridad de los sistemas de control. Un ejemplo de ello, es la detección de anomalías a partir del monitoreo de procesos, en donde se usa técnicas que se basan en poder conocer las condiciones del sistema y poder diagnosticar fallas, detectándolas e identificándolas en etapas tempranas para evitar paradas y en consecuencia planificar acciones de mantenimiento [91]. Estas técnicas pueden ser vistas desde varios enfoques como lo son:

- **Basada en señales** Estos métodos de diagnósticos se basan básicamente en el procesamiento de señales como la vibración, temperatura, ruido acústico, presión, voltajes, corrientes, etc. Realizan análisis temporales y frecuenciales para poder determinar un índice de falla. La falla se detecta una vez que estos índices se cruzan en unos umbrales definidos. Una idea similar a la técnica de FDI.

- **Basada en modelos** Se soportan en un modelo para la predicción de fallas. Requieren un conjunto completo de parámetros del sistema para reducir la incertidumbre y lograr una alta precisión en el diagnóstico de la falla. Este enfoque necesita de precisión en los modelos matemáticos, además los atacantes a menudo se camuflan en estados normales del sistema, dificultando la detección y clasificación de los ataques.

- **Técnicas basadas en datos** Ofrecen ventajas sobre los otros métodos ya que no necesitan establecer ningún modelo físico y se pueden generalizar a una amplia gama de aplicaciones. El proceso consiste en inicialmente extraer características para posteriormente clasificarlas. Sistemas expertos, redes neuronales artificiales, lógica difusa y otros algoritmos se han usado ampliamente en este método.

En [25], determinan que la seguridad en los CPSs puede ser clasificado dentro de dos áreas: seguridad de la información y control de seguridad. La primera de ellas involucra asegurar la información durante el procesamiento de datos, la agregación de datos, así como el uso compartido a gran escala en una red. La seguridad de la información se puede llevar a cabo usando técnicas de encriptación, además de usar mecanismos como control de acceso, certificación, autenticación, monitoreo de entorno, protocolos seguros de enrutamiento, deshabilitar componentes innecesarios, respuesta a incidentes, entre otros. Mientras que la segunda área, abarca la resolución de cualquier problema de control en el entorno de la red, así como la mitigación de los ataques llevados a cabo, usando

algoritmos de estimación para poder reconfigurar los sistemas de control. Aún así, los métodos utilizados para analizar los riesgos de seguridad es diferente dependiendo de las características de los sistemas.

Al mismo tiempo, la detección confiable en tiempo real juega un rol importante en garantizar la fiabilidad y seguridad en este tipo de sistemas, en donde los sensores se encuentran propensos a constantes fallas o ataques maliciosos. Propuestas como la detección de anomalías permiten suponer relaciones espacio temporales en la red de sensores, las cuales permiten detectar anomalías generales a gran escala [77]. Se han presentado algunas soluciones en tiempo real asociados a la seguridad en sistemas embebidos, principalmente orientadas al desarrollo de nuevas arquitecturas, algunas asociadas al acceso a la memoria y metodologías de desarrollo.

Desde el punto de vista del control se han generado propuestas en donde se separa la estimación y control del estado; en ese caso se parte del supuesto de que si existe una ley de retroalimentación que estabilice la planta, a pesar de cualquier ataque a un conjunto determinado de sensores, entonces existe un algoritmo de estimación que permita estimar el estado de la planta a pesar de cualquier ataque a un conjunto de sensores [4].

Teniendo esto en cuenta, uno de los aspectos más relevantes al abordar la seguridad de los sistemas ciberfísicos es estudiar cómo se controla el proceso físico usando principios de control. Dentro de los principales desafíos está el de adquirir la suficiente comprensión de los requisitos de seguridad del proceso bajo control. Es deseable determinar la mayoría de los parámetros de control que son vulnerables en estos sistemas y poder llegar a analizar la sensibilidad en los lazos de control. Para lograr este objetivo en [13] se propuso un método donde se captura el comportamiento dinámico del sistema ciberfísico con y sin ataques, y se modela la propagación del impacto de estos ataques.

Como se ha visto, estas situaciones no deseadas se han presentado en diferentes sectores e independientemente de la naturaleza de los ataques, como pueden ser ataques no invasivos o ataques donde se inyectan falsos datos a las redes de control, eventualmente resulta en daños físicos, debido a que las mediciones de los sensores se ven involucradas por modificaciones en la información que se transmite en las redes de comunicación [3]. Investigadores han tenido en cuenta el impacto de tipos de ataques, como el de Denegación del Servicio, en los procesos físicos, en donde los experimentos muestran que para interrumpir un proceso físico los atacantes deben realizar los ataques en momentos adecuados, de lo contrario las consecuencias serán limitadas [91].

Otros factores, tales como la pérdida de paquetes, retardos en la comunicación, la lógica

de control, el tiempo de programación y el tráfico de fondo de la red, pueden afectar las consecuencias de los ataques. Así, han podido determinar los parámetros de red más importantes que pueden afectar a la resiliencia de los procesos físicos [92].

En las Tablas II, III y IV se describen varios aspectos de sistemas ciberfísicos que han sido vulnerables frente a algunos ataques que se han realizado en sistemas de control industrial, Smartgrids y dispositivos médicos. Esta descripción incluye el elemento atacado, los cambios resultantes en el objeto atacado (Influencia), los componentes afectados indirectamente, los cambios o impactos ocasionados, así como el método por el cual se llevó el ataque y las condiciones previas necesarias para que el ataque fuera exitoso [1].

TABLA II.
Ataques cibernéticos en sistemas de control industrial.

Nombre	Elemento atacado	Influencia	Elemento afectado	Impacto	Método	Condiciones previas
Maroochy	Bombas	Bombas trabajan incorrectamente.	Ajustes correctos en el bombeo de las estaciones manipuladas.	Aguas negras, inundaciones en calles, pérdidas financieras, afectaciones al medio ambiente.	Se usa un PC y un transmisor para manipular las bombas.	Conocimiento privilegiado.
Stuxnet	PLCs de centrifugadoras.	Rotación acelerada en las centrifugadoras.	Rotores de las centrifugadoras.	Reducción del tiempo de vida y daños físicos.	Comandos ilegítimos enviados a los PLCs	PLC infectado por Stuxnet.
Modbus worm	Red de trabajo del sistema de control industrial.	Red de trabajo infectada del sistema de control industrial.	Dispositivos conectados a la red.	Reinicio en servidores.	Inyección de código malware dentro del tráfico de la red de trabajo.	Acceso al tráfico del sistema de control industrial.

TABLA III.
Ataques cibernéticos en Smart Grids.

Nombre	Elemento atacado	Influencia	Elemento afectado	Impacto	Método	Condiciones previas
Extorsión cibernética	Entrega de energía.	Utilidades pierden control sobre su sistema de red.	Consumidores.	Pérdida del servicio y afectaciones financieras.	Explotan elementos de la red conectados a la Internet.	Conocimiento privilegiado.
Experimento aurora	Breaker del circuito.	Cambios en el comportamiento los relays.	Generadores de energía y y subestaciones.	Daños físicos en los generadores e inhabilitamiento del servicio de energía.	Apertura y cierre inesperado de de los interruptores.	Acceso y conocimiento privilegiado.

3.3.1 Estado actual de los métodos para la detección de ataques en sistemas de control

El sector eléctrico también ha incrementado su interés en abordar la problemática de la integridad y confiabilidad de la información en relación a los enfoques de generación

TABLA IV.
Ataques cibernéticos en en dispositivos médicos.

Nombre	Elemento atacado	Influencia	Elemento afectado	Impacto	Método	Condiciones previas
DoS	Dispositivo médico.	El dispositivo es desconectado.	Pacientes.	El paciente no recibe la terapia esperada.	Retransmitir comandos de apagado.	Capturar el comando de apagado previamente al envió realizado por el programador.
Inyección de datos falsos	Bomba de insulina.	Medidas falsas enviadas a la bomba de insulina.	Terapias del paciente.	Decisiones incorrectas y condiciones de salud peligrosa.	Personificando paquetes de datos falsos para envíos.	Intersección de los canales de comunicación de la bomba de insulina.

y distribución de energía en smartgrids [93–96]. Una vez más los CPSs juegan un rol importante, especialmente en la generación de nuevas metodologías en donde se monitoreen, transmitan, procesan y controlen variables, además de garantías de seguridad en los diferentes niveles de la red, en donde los tiempos de cumplimiento de actividades son críticos [97]. Para abordar estos desafíos se han realizado propuestas a partir de métodos como la teoría de juegos [98] y la teoría de grafos [99]. La revisión realizada ha permitido identificar que los fabricantes de los equipos que se usan en las smartgrids eligen su propio entorno de desarrollo, lo que conduce a una diversidad de sistemas operativos y arquitecturas de CPUs, por lo que se dificulta aún más abordar problemáticas de vulnerabilidad para diferentes entornos. Por otra parte, la mayoría de metodologías anteriores fueron desarrolladas en arquitectura x86 sin considerar la arquitectura ARM, cuando muchas de las redes inteligentes integran la ejecución del software sobre estas arquitecturas [100]. Dentro de este sector se han realizado procedimientos para mitigar las consecuencias de los ataques cibernéticos, en donde de manera cíclica se realizan procesos relacionados con la prevención, detección, restricción, restauración y adaptación del sistema [101].

La detección de ataques en las smartgrids, se centran en la percepción de ataques maliciosos. Los atacantes a menudo se caracterizan por comprometer las mediciones y/o controlar los datos buscando una estrategia de ataque eficaz para dañar o afectar un sistema sin poder ser detectado. Debido a las perturbaciones y errores en los sistemas de control, las estrategias de ataques se esconden dentro del margen de error normal para evitar que se activen las alarmas. De este modo, se debe dedicar al diseño de una estrategia de detección válida que permita distinguir los comportamientos de ataque de las perturbaciones y errores dentro del sistema, de forma inteligente [102].

Para asegurar estos sistemas se deben tener en cuenta los siguientes aspectos:

• Detección usando métodos avanzados tales como estimadores de estado, métodos de optimización combinatoria, y sistemas artificiales para detectar datos maliciosos y posibles ciberataques [38].

• Modelado de ataques con condiciones más prácticas, algunas especificaciones asumidas son a menudo introducidas, sin embargo, estas restricciones violan el hecho de que los ataques suelen ser arbitrarios, y los resultados son poco prácticos.

• Detección y estimación de ataques de manera distribuida, debido a la complejidad y distribución espacial de estos sistemas, aumenta las dificultades de detección y estimación. Adicionalmente, puede haber varios ataques al mismo tiempo. Entonces, cómo ubicar y estimar diferentes ataques de forma distribuida es primordial en estos estudios.

• Estrategias de control resilientes, como complemento del método de protección de los sistemas de información, el diseño de las estrategias de control de seguridad juega un papel muy importante en la protección de las smartgrids. Cuando tradicionalmente las protecciones de TI no son válidas, las implementaciones de control conducirán a una mejora significativa en la garantía del desempeño de estos sistemas. Por un lado, el diseño del control debe satisfacer los requisitos generales cuando no hay ataques. Por otro lado, deben seguir siendo válidos para ataques en lugar de rediseñar o cambiar el controlador y por lo tanto, cómo diseñar un controlador de seguridad de este tipo, se requieren.

Adicionalmente, en los últimos años se ha incrementado el uso de máquinas de aprendizaje para poder detectar comportamiento anómalo y detección de intrusos en la redes de comunicación que estos sistemas manejan [103, 104]. De estas se destacan métodos que combinan aprendizaje supervisado, como lo son las SVMs (Support Vector Machine por sus siglas en inglés), como no supervisado. Aunque, las máquinas de aprendizaje y técnicas basadas en inteligencia artificial son cada vez más usadas para la detección de ciberataques dentro de este tipo de sistemas con resultados obtenidos prometedores, se requiere investigación para el desarrollo en ambientes operacionales y entornos más prácticos [38, 105–107].

En [27] se busca mitigar el efecto de estos ciberataques dentro de un sistema de energía, proponiendo una nueva técnica analítica que se basa en la teoría de la cadena de Markov y una distancia Euclidiana. Usando datos históricos de un conjunto de buses confiables, un modelo de Markov del sistema en comportamiento normal se formula. Los estados estimados se analizan calculando la distancia euclidiana desde este modelo

con el proceso real. Los estados que coinciden con la probabilidad más baja se consideran estados atacados. Este método es capaz de detectar un ataque malicioso, que son indetectables por otros métodos. Para trabajos futuros, se pretende realizar análisis que permitan identificar otro tipo de ataques.

Igualmente en [47], se usan cadenas de Markov de primer orden, para detectar el tráfico relacionado con ataques del tipo DoS, en una red de sensores inalámbricos. El principio de detección de anomalías es dividir el tiempo en intervalos de observación de igual longitud. Durante la primera fase, denominada fase de aprendizaje, los nodos recopilan información sobre el tráfico normal esperado. Esto se logra observando un conjunto de características y trafico predefinido y potencialmente multidimensional. Al finalizar la etapa de aprendizaje, cada nodo posee una serie de tiempo, que contiene puntos de observación del tráfico. Esto se denomina perfil del tráfico que describe dicha serie de tiempo. Este perfil es una cadena de Markov. Esta cadena es usada para detectar la baja probabilidad de eventos. Un evento de baja probabilidad es una secuencia de transiciones de baja probabilidad en la cadena. Si las transiciones improbables se acumulan en algún periodo de tiempo, el tráfico se identifica como anómalo. Se busca que este trabajo, se adapte a redes más dinámicas.

Otra forma de usar estos modelos, se puede evidenciar en [108] donde se considera un problema de control para un proceso estocástico, el cual es sensible a ataques tipo DoS. En este trabajo se usa un modelo de Markov, para modelar la estrategia del ataque, el cual bloquea estocásticamente los paquetes de control en el sistema.

Recientemente se han usado algoritmos basados en Bosques aleatorios (Random Forest), para detectar comportamientos maliciosos a través de bases de datos, donde la clasificación binaria es usada para clasificar si el contenido de un paquete es malicioso o no. Por una parte, estos reducen el costo computacional, sin embargo no garantizan una alta precisión [109].Por otra parte, solo es restringido a la transmisión de paquetes y no se conoce de manera exacta en que parte del paquete ocurre la anomalía. Agregándole a esto, que no permite especificar el tipo de ataque [36, 37].

De igual manera se han presentado trabajos que presentan varias técnicas inteligentes, tales como SVMs, algoritmos genéticos [38], redes auto organizadas de colonias de hormigas y máquinas de aprendizaje extremo, los cuales proveen modelos con muy altas precisiones, que son aplicadas en el contexto de la seguridad en redes informáticas, y especialmente en la detección de intrusos y/o ataques. Tales técnicas buscan lograr mejores tasas de reconocimiento en la detección de intrusos, pero aún se percibe que la tasa de falsos positivos sigue siendo el problema para abordar en todos los estudios. Aunque algunas técnicas pueden reducir la tasa de falsos positivos, por el contrario, au-

menta el tiempo de entrenamiento y la clasificación. Este tema es muy importante para la detección de intrusos, donde la detección en tiempo real es un factor relevante [110]. Además, aplicar técnicas de aprendizaje automático y otras técnicas inteligentes, es un desafío debido a que se requiere mayor memoria y potencia de procesamiento que pueden afectar el rendimiento del funcionamiento y la prestación de servicios de dentro del sistema. Además, estas técnicas no son adecuadas para detectar mutaciones de varios ataques. Técnicas avanzadas como Deep Belief Networks (DBN por sus siglas en inglés) y redes neuronales convolucionales profundas (Deep CNN) son posibles soluciones [105, 111].

En [44] se hace uso de este método para clasificar el comportamiento normal y anormal, de un tráfico de datos que puede estar sometido a ataques tipo DoS, con la ayuda de un conjuntos de datos dados para el entrenamiento. Esta máquina hace predicciones de datos y además proporciona resultados relativamente buenos con menos tiempo de entrenamiento. El algoritmo basado en SVM que se realiza, consta de una etapa de entrenamiento y una etapa de evaluación. En la etapa de testeo se desarrolla en primer lugar unos agentes detectores de intrusos, en donde se agrupan por clusters con la ayuda de vectores de datos llamados vectores de soporte. Cada uno de ellos envía estos datos a un nodo adyacente. De este modo se va actualizando el vector de soporte en cada nodo para poder calcular el hiperplano clasificador. El proceso continúa hasta que todos los agentes de identificación en el mismo cluster alcanzan la misma SVM entrenada.

Asimismo en [45] desarrollan un método basado en PCA y SVM para detectar ataques DoS. El articulo analiza el fenómeno de los ataques DoS en una red bajo el protocolo TCP. Para filtrar las interferencias del ambiente y extraer las principales características de manera efectiva y reducir el dimensionamiento de información, se usa el algoritmo de PCA, extrayendo así las principales componentes del flujo de datos, seleccionando ciertos vectores de acuerdos a las necesidades, de este modo se reducen las dimensiones sin perder información de los datos originales. Es así como a partir de estas características se usa una SVM para resolver el modelo de un óptimo hiperplano, para evaluar los datos y proceder a clasificar (dato corrupto o no) y predecir, para lograr la detección. En la Fig. 13 se muestra el diagrama de flujo del método propuesto. Los resultados muestran que el algoritmo tiene una alta precisión y baja tasa de falsos positivos y falsos negativos.

En el sector automovilístico se tiene una creciente en el uso de vehículos en red. Los vehículos en red, son sistemas emergentes que no están ajenos a estos problemas de seguridad dada la creciente cantidad de investigaciones que demuestran la capacidad que tienen los atacantes para implementar ataques a automóviles. Los atacantes han podido controlar con éxito una amplia gama de funciones automotrices (por ejemplo,

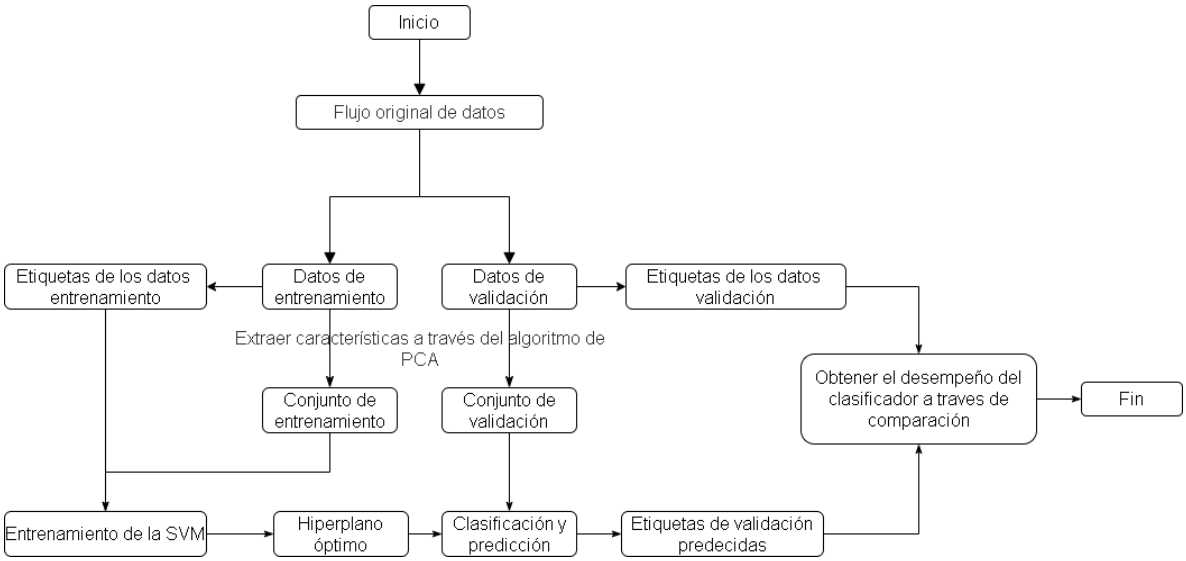


Fig. 13. Diagrama de flujo del método de detección basado en PCA-SVM.

desactivación de frenos y parada de motores), provocando un retiro de 1.4 millones de vehículos. Estos ejemplos de los ataques automotrices han estimulado enormemente a los investigadores para generar sistema de detección de intrusos con la capacidad de prevenir daños graves causados por este tipo de situaciones [109, 112].

Las siguientes categorías han sido propuestas para contra restar ataques maliciosos en este tipo de CPSs [113]:

- Asegurar la confidencialidad e integridad de los mensajes a través de métodos de encriptación y tecnologías de autenticación. Estos entornos requieren confiabilidad en tiempo real y debido a costos computacionales altos, el ancho de banda y los recursos de almacenamiento son restringidos. Por lo tanto, esta categoría es en la gran mayoría de aplicaciones muy poco práctica.

- El uso de interfaces tales como políticas de seguridad y firewall para separar ataques potenciales. Esta estrategia también es poco útil debido al largo ciclo útil que tienen estos sistemas y a la conversión y actualización de los sistemas electrónicos automotrices existentes para mejorar la seguridad, requieren un largo período de tiempo, y estas interfaces no lo proveen.

- Desarrollo de sistemas de detección de intrusos (IDS), los cuales tienen la capacidad de identificar situaciones adversas. Es un método eficaz y compatible con versiones anteriores para la protección de estos sistemas de los ataques y se puede aplicar con recursos limitados de ancho de banda en consideración de las deficiencias de los métodos

antes mencionados.

Estos últimos, permiten ser abordados desde diferentes aristas, cómo se ha mencionado anteriormente. Por ejemplo, se encuentran métodos basados en firmas, donde a partir de datos conocidos tales como medidas de tensión, permiten detectar rápidamente accesos ilegales. Este debe ser tratado como una estrategia integral, y la tecnología utilizada puede combinar métodos de aprendizaje automático, para etapas de extracción y clasificación de características.

Los IDS son usados ampliamente para detectar ciberataques o comportamientos anormales, analizando el tráfico de la red. Sin embargo, los atacantes hábiles pueden ocultar la manipulación de la información del sistema y evitar la detección de intrusiones, siguiendo de cerca el comportamiento esperado del sistema, pero aun inyectando de manera suficiente, información falsa en el sistema que después de un largo período de tiempo, logran sus objetivos del ataque. Los ataques de este tipo se denominan ataques furtivos, que no pueden ser identificados por IDS tradicionales, en las que solo las magnitudes de los residuos son rastreadas y evaluadas.

También se tienen métodos basados en el monitoreo de parámetros, los cuales suelen ser ineficientes para amenazas de seguridad desconocidas y los parámetros pueden variar en diferentes redes de vehículos. Los investigadores de seguridad de estos sistemas han propuesto muchos IDS cuyo diseño se basa en la teoría de la información y el aprendizaje automático para abordar estos problemas.

Por otro lado, si se tiene en cuenta diseños basados en la teoría de la información, se suelen usar las medidas para lograr detecciones de anomalías, siguiendo métodos no supervisados. Específicamente en la comunicación interna de los elementos eléctricos, se usa la medida de entropía, la cual puede reflejar la anomalía. Sin embargo, se ha mostrado que es ineficaz en ataques que modifican el contenido del campo de los datos transmitidos por CAN, protocolo que se utiliza usualmente en estos sistemas. Mientras que los sistemas basados en aprendizaje automático pueden usar métodos de clasificación, que permiten aprender el comportamiento normal del tráfico de la red y si existe alguna desviación podría identificar situaciones anómalas.

Además, se han usado métodos de aprendizaje profundo que están capacitados para extraer características de baja dimensión y se utilizan para discriminar paquetes normales y de piratería, y de esta manera detectar intrusos. En [114] la técnica propuesta puede proporcionar una respuesta al ataque con una tasa de detección considerablemente alta ($99{,}8\,\%$). En [115] se sugirió un detector de anomalías basado en la red neuronal recurrente LSTM para detectar ataques con baja tasa de falsas alarmas. Este tipo de

métodos son los que mejor respuesta han tenido en estos entornos, aunque en ocasiones los costos computacionales son altos [46, 105], especialmente en el proceso de aprendizaje.

En ese mismo contexto, en [46] se propone una SVM usando una función de kernel de base radial, para detectar ataques en sistemas automovilísticos en red. Con esto se busca evitar dos inconvenientes asociados al enfoque de clasificación tradicional de anomalías, donde a menudo no se tiene un conjunto de datos anormales y si existen es muy probable que no sean lo suficientemente representativo, debido a que se desconoce muchas de las fallas en un sistema. De este modo si se usa un conjunto de datos de entrenamiento no representativo de anomalías, se obtiene un modelo de decisión incorrecto.

En [116] se plantea la permutación basada en entropía que puede distinguir los residuos producidos de una serie aleatoria efectivamente. Como resultado, los ataques sigilosos pueden ser identificados de acuerdo con el cambio de esta entropía. Sin embargo, el trabajo todavía tiene algunas limitaciones. Este, requiere de la selección manual de los valores de los parámetros del algoritmo (por ejemplo, el umbral de detección, el tamaño del rango de mapeo residual, la longitud de la serie de prueba y el orden de entropía de permutación), que conduce a una dependencia excesiva sobre la experiencia humana. Además, una subjetividad e inapropiada elección puede tener un impacto negativo en el rendimiento de la detección. En el futuro, planean estudiar y construir las relaciones formales entre los parámetros del algoritmo y el rendimiento de detección y luego diseñar una actualización de parámetros automática y en tiempo real, basado en la teoría de control de retroalimentación, con el fin de realizar la actualización automática de los valores de los parámetros según el cambio del rendimiento de la detección.

3.3.2 Tecnologías de uso reciente en automatización industrial

En relación con este tema, con el objeto de cumplir en un amplio rango los diversos requisitos de seguridad, la industria aeroespacial ha permitido abrir nuevas posibilidades en estos campos adaptando conceptos y técnicas como la virtualización, en donde la seguridad se basa en las propiedades de aislamiento temporal y espacial proporcionadas por un hipervisor [71], aun así, la disponibilidad de nuevos procesadores genera necesidades mayores e inquietudes en cuestión de seguridad. También se han realizado experimentos para explorar la posibilidad de poder usar la tecnología de virtualización en los sistemas de control industrial para migrar aplicaciones de tiempo real [65], aun así es necesario realizar investigaciones más profundas que permitan cubrir aspectos tales como la sobrecarga de memoria en soluciones donde se usen contenedores, las

gestión de los datos que se comparten entre contenedores y verificar su aplicabilidad en el mundo real.

En este mismo sector se han explorado enfoques basados en la diversidad, lo cual permite mejorar la seguridad y la integridad del software, y es recomendado para realizar diseños de software tolerante a fallos. El uso de este enfoque, permite diversificar copias de los mensajes que se van a transmitir, en donde se envían duplicados del mensaje que comprende los datos, expresados de manera diferente y se detecta un error si los datos restaurados de diferentes copias recibidas no coinciden [117].

Otra forma de aprovechar la tecnología de virtualización en procesos de control industrial, es para los PLC donde se busca integrarla en el software que se ejecuta en el dispositivo integrado en el campo. En [74] examinan los desafíos al aplicar técnicas de virtualización para servidores en el dominio de los sistemas integrados en tiempo real. La sobrecarga inducida por una solución de virtualización basada en hipervisor dificulta su uso para sistemas heredados en la automatización industrial, aunque el uso de la emulación de conjuntos de instrucciones combinado con el uso de generaciones de procesadores más potentes puede ayudar a superar este problema [118].

El principal obstáculo para el uso de la virtualización en PLCs y controladores similares es garantizar la puntualidad con las soluciones de virtualización existentes. La granularidad limitada en la que la virtualización basada en hipervisores puede encapsular la funcionalidad [119] aún da como resultado que la funcionalidad heredada se integre en el nivel de la aplicación [120] en lugar de utilizar técnicas de virtualización. Sin embargo, la literatura existente sobre virtualización para PLCs se centra en técnicas de virtualización basadas en hipervisores.

En [66] se presenta una arquitectura para un controlador industrial multipropósito que forma un sistema ciberfísico en el área de la automatización industrial. La arquitectura se basa en gran medida en conceptos de contenedores creados principalmente a partir de sistemas en la nube. Los resultados obtenidos son prometedores en el sentido de que la sobrecarga de la contenedorización es muy baja y bastante constante, mientras que parece mejorar la estabilidad de la aplicación en tiempo real cuando se ejecutan cargas de trabajo adicionales en tiempo no real en paralelo en el sistema y también dentro de los contenedores. Aunque las latencias del peor de los casos observadas durante la evaluación en un sistema Intel aún no son completamente adecuadas para aplicaciones exigentes de control en tiempo real. El dominio de aplicación al que se dirige este trabajo son los PLC y los controladores de automatización con ciclos de control que se ejecutan en el rango de $100ms$ a $1s$. El plazo límite para ejecutar las tareas de control es equivalente al tiempo del ciclo, mientras que la latencia y la fluctuación aceptables

para las tareas de control es típicamente el $10\,\%$ del tiempo del ciclo. Dependiendo de la planta de producción controlada o del proceso industrial, el incumplimiento del plazo también es aceptable siempre que ocurra raramente y no repetidamente. Los resultados presentados demuestran que una ejecución en contenedores de aplicaciones de control puede cumplir con estos requisitos. Por lo tanto, la emulación en contenedor de hardware heredado podría usarse para migrar la aplicación de control existente desarrollada para sistemas de control heredados a la arquitectura de contenedor propuesta y, por lo tanto, eliminar la dependencia de los componentes de hardware heredados.

La filosofía de utilizar las dos plataformas de contenedores más utilizadas, LXC y Docker, es diferente [74]. Los contenedores Docker están basados en microservicios (cada contenedor debe contener una sola aplicación), mientras que LXC, al igual que las máquinas virtuales, permite ejecutar un ecosistema complejo de aplicaciones que es beneficioso para la emulación de sistemas heredados.

Para garantizar el comportamiento predecible en el tiempo de los contenedores, los sistemas operativos deben proporcionar dicha capacidad. Por defecto (Vanilla) Linux no ofrece garantías de tiempo sobre la ejecución de las tareas y, por lo tanto, la previsibilidad es baja. Sin embargo, existen varios enfoques para mejorar la previsibilidad: el parche en tiempo real que mejora la preferencia del kernel de Linux y los enfoques del co-kernel que ejecutan un micro kernel en tiempo real en paralelo al kernel de Linux. Las aplicaciones en contenedores se programan de la misma manera que las aplicaciones nativas utilizando el programador del host, el kernel predeterminado de Linux proporciona tres planificadores: (i) CFS: tiene como objetivo maximizar la utilización de la CPU al mismo tiempo que maximiza el rendimiento interactivo. No da garantías de tiempo. (ii) Programador en tiempo real (RT): El programador permite programar tareas con una prioridad fija utilizando las políticas primero en entrar-primero en salir(FIFO) o Round Robin. Las tareas se ejecutan hasta que ceden o son reemplazadas por tareas de mayor prioridad. La extensión de programación de grupo en tiempo real permite dividir y asignar el tiempo de la CPU entre tareas en tiempo real y no en tiempo real. (iii) EDF Utiliza un servidor de ancho de banda constante y permite asociar a cada tarea un plazo y un período.

El parche en tiempo real (PREEMPT_RT) mejora las primitivas de bloqueo del kernel para maximizar las secciones interrumpibles. La ventaja del parche es que los desarrolladores de aplicaciones no necesitan bibliotecas especiales o API.

En [121] consideran contenedores en tiempo real en el contexto de sistemas de automatización industrial que funcionan con datos en tiempo real y tienen plazos en tiempo real para la detección y respuesta a eventos. El documento enfatiza la necesidad de virtua-

lización a nivel de sistema operativo en una automatización industrial y da ejemplos de requisitos de temporización de aplicaciones industriales (por ejemplo, el accionamiento de motor generalmente requiere un tiempo de ciclo entre $1ms$ y $250\mu s$) y la necesidad de sincronización entre los contenedores. La evaluación de los efectos de los contenedores en el rendimiento de los sistemas de automatización industrial se proporciona en dos casos: (i) comportamiento cíclico de una aplicación en contenedores, (ii) rendimiento de la red virtual para comunicaciones entre contenedores. La prueba de comportamiento cíclico evalúa la capacidad de ejecutar la lógica de la aplicación a intervalos predefinidos, mide la precisión y la fluctuación. La prueba de red virtual evalúa la capacidad de comunicarse entre contenedores coubicados de manera limitada en el tiempo. Las investigaciones ven la computación de contenedores en tiempo real como una tecnología prometedora, sin embargo, los mecanismos de comunicación entre contenedores no están claros.

En [66] y [122] se aborda una arquitectura basada en contenedores para controladores en tiempo real que permiten una implementación de funciones flexible y un soporte de aplicaciones de control heredadas. Dicha arquitectura es necesaria para preservar la funcionalidad de los programas de control heredados y para reducir el costo de mantenimiento de los sistemas heredados (en los que el software a menudo está limitado a un ecosistema de hardware y software específico). Los investigadores analizan la viabilidad de construir un sistema con capacidad en tiempo real (para sistemas heredados) basado en contenedores en tiempo real, apuntan a PLC y controladores de automatización con un tiempo de ciclo entre $100ms$ y $1s$. Realizan un conjunto de pruebas en varios escenarios de carga (i) utilizando aplicaciones en contenedores dentro de Docker y (ii) ejecutando el sistema operativo completo (PowerPC) dentro de LXC. Llegan a la conclusión de que una ejecución en contenedores de aplicaciones de control puede cumplir con los requisitos de los PLC y los controladores de automatización.

En los métodos basados en co-Kernel en tiempo real un micro-kernel en tiempo real se ejecuta en paralelo al kernel de Linux. El co-kernel en tiempo real maneja actividades críticas en el tiempo (por ejemplo, manejar interrupciones y programar subprocesos en tiempo real), el kernel estándar de Linux se ejecuta solo cuando el co-kernel está inactivo. En comparación con el parche en tiempo real, el enfoque de co-kernel ofrece latencias más bajas y menor jitter. Por otro lado, requiere API, herramientas y bibliotecas especiales para el desarrollo de la aplicación. Además, existen impedimentos para escalar soluciones de co-kernel en plataformas grandes (por ejemplo, plataformas de muchos núcleos). Hay dos alternativas de co-kernel: Interfaz de aplicación en tiempo real (RTAI, Real-Time Application Interface por sus siglas en inglés) y Xenomai.

RTAI tiene como objetivo minimizar las latencias a los valores técnicamente más bajos

posibles. Las tareas en tiempo real se compilan como módulos del kernel y se ejecutan en el espacio del kernel. Xenomai es una bifurcación de RTAI. Su misión es permitir tareas en tiempo real en el espacio del usuario. Consiste en una capa de emulación que es capaz de reutilizar código de otros RTOS. En [73] se presenta una arquitectura sobre la modularización de aplicaciones de control en tiempo real en contenedores en tiempo real. Dicha arquitectura modular necesita dos partes esenciales: (i) parte computacional, habilitada por un sistema operativo en tiempo real (combinación de Xenomai y parche en tiempo real), y (ii) parte de mensajería que permite pasar mensajes entre contenedores en tiempo real. Las arquitecturas monolíticas tradicionales se comunican a través de llamadas a funciones y memoria compartida, los contenedores no asumen si se ejecutan en el mismo host o en un entorno distribuido (se comunican a través de la pila de red del sistema operativo estándar), por lo tanto, el paso directo de mensajes a través de la memoria compartida no es apoyado directamente. Por lo tanto, los investigadores proporcionan un diseño e implementación de un sistema de mensajería en tiempo real personalizado para contenedores basado en ZeroMQ.

En [123] se propuso un método que consiste en contenedores en tiempo real y en tiempo no real. El programador garantiza la capacidad de CPU asignada a los contenedores en tiempo real y distribuye dinámicamente la capacidad no utilizada a los contenedores que no son de tiempo real. El trabajo complementa Completely Fair Scheduler con un módulo de ajuste de carga de trabajo que recopila la utilización de CPU por contenedores y un módulo de ajuste dinámico que asigna CPU al contenedor.

En [124] y [125] se implementaron contenedores en tiempo real usando Linux parcheado con co-kernel en tiempo real (RTAI) y utilizando módulos personalizados de monitoreo y aplicación de políticas. Su solución permite convivir contenedores con diferentes niveles de criticidad y evitar que tareas periódicas duras en tiempo real de prioridad fija dentro de los contenedores afecten las garantías temporales de otros contenedores. Las garantías temporales se dan a través de dos mecanismos: (i) asignación de prioridad de tareas propias a tareas dentro de los contenedores y (ii) seguimiento y ejecución de políticas de protección temporal. El primero asegura que a las tareas dentro de los contenedores de alta criticidad se les asigne una prioridad más alta que a las tareas en los contenedores de baja criticidad y, por lo tanto, nunca sean reemplazadas por tareas de contenedores de menor criticidad. El último monitorea las tareas y, en caso de sobrecostos o tiempo extra, aplica una de las políticas de protección temporal (es decir, mata o suspende la tarea defectuosa, suspende la tarea hasta el próximo período).

En [126] se propuso una arquitectura que utiliza contenedores para modularizar aplicaciones de control en tiempo real. Analizaron las oportunidades y desafíos que conlleva una aplicación de control en tiempo real basada en contenedores y propusieron una

arquitectura de referencia que permite la reutilización, portabilidad y flexibilidad. La arquitectura incorpora soluciones para la comunicación entre contenedores y entre contenedores y hardware. Demostraron que el parche PREEMPT_RT se puede combinar con el kernel de Cobalt y que las aplicaciones basadas en Cobalt se pueden ejecutar dentro de contenedores. Realizaron evaluaciones comparativas que prueban el tiempo de ida y vuelta de los mensajes con diferentes métodos de transporte. Los resultados sugieren que los tiempos de ida y vuelta entre 50 y $150\mu s$ en el peor de los casos son factibles y, por lo tanto, es posible implementar una aplicación con un intervalo periódico de $500\mu s$ que se puede dividir en varios módulos dependientes.

En [122] realizan pruebas comparativas en aplicaciones de PLC industriales modularizados. Este análisis examina el impacto de la virtualización basada en contenedores en las limitaciones en tiempo real. Como no existe una solución para la migración de código heredado de PLC, la migración a contenedores de aplicaciones podría extender la vida útil de un sistema más allá de los límites del dispositivo físico. A pesar de que las pruebas mostraron una latencia en el peor de los casos del orden de $15ms$ en hosts basados en Intel, los autores argumentan que los motores de contenedor pueden ser reducidos y optimizados para la ejecución en tiempo real. En [66] se describió y probó una posible arquitectura multipropósito en un caso de uso del mundo real. Los resultados muestran el peor caso de latencia de aproximadamente $1ms$ para una computadora de placa única Raspberry PI, lo que hace que la solución sea viable para tiempos de ciclo de aproximadamente $100ms$ a $1s$. Los autores afirman que aún deben investigarse temas como la sobrecarga de memoria, el acceso restringido de los contenedores y los problemas debidos a la inmadurez de la tecnología.

En [73] aborda cómo se puede lograr la comunicación determinista de contenedores y dispositivos de campo en una nueva arquitectura basada en contenedores. Propusieron una solución basada en Linux como sistema operativo host, que incluía ambos sistemas parche PREEMPT-RT centrado en la apropiación del kernel y Xenomai orientado al co-kernel. Con este parche, el enfoque exhibe mejor previsibilidad, aunque adolece de problemas de seguridad introducidos por archivos de sistema expuestos requeridos por Xenomai. Por esta razón, sugirieron limitar su aplicación para la ejecución de código crítico para la seguridad. Analizaron y discutieron la mensajería entre procesos en detalle, centrándose en las propiedades específicas necesarias en las aplicaciones en tiempo real. Finalmente, implementaron un tiempo de ejecución de organización que administra la comunicación dentro del contenedor y demostraron que son posibles tiempos de tarea tan bajos como $500\mu s$.

En [127] se exploran los límites y la viabilidad de migrar aplicaciones en tiempo real de servidores bare-metal a configuraciones IAAS virtualizadas. Demostraron que la conte-

nerización ofrece un paradigma novedoso para las aplicaciones de control. Sin embargo, las tareas de cálculo previamente aisladas pueden operar simultáneamente e interactuar entre sí, lo que podría influir en el rendimiento del tiempo. Las técnicas de virtualización modernas funcionan lo suficientemente bien como para adaptarse a entornos duros en tiempo real. Tanto la prueba de latencia como la de rendimiento mostraron resultados satisfactorios que confirman la viabilidad de la migración de una aplicación. Concluyen que este nuevo paradigma requiere una investigación sobre temas como la seguridad de los contenedores, el acceso restringido y el intercambio de datos dentro de los contenedores y sugieren una arquitectura para ayudar a la migración y ubicación de estas nuevas aplicaciones en un enfoque de Industria $4{,}0$.

En [128] se centraron en identificar el rendimiento de las instancias basadas en la virtualización de hardware a través de máquinas virtuales basadas en kernel (KVM) y la virtualización de sistemas operativos de contenedores utilizando Docker compatible con varias plataformas. Los puntos de referencia confirman que Docker da como resultado un rendimiento igual o mejor que los KVM en casi todos los casos. En [84] analizaron tres técnicas de contenedorización para su uso en la computación en nube. El artículo compara Linux Containers (LXC) de Canonical, Docker y Singularity, con una aplicación completa. En muchos aspectos, los contenedores Singularity funcionaron mejor, a veces incluso mejor que la implementación completa, pero esto se debe en gran parte al enfoque combinado del motor; Singularity es una solución de virtualización incompleta, ya que otorga acceso a operaciones de E/S sin cambios de contexto.

Entre los desafíos de la virtualización basada en contenedores en tiempo real se identifican los siguientes [74]:

- Comunicación entre contenedores en tiempo real. La comunicación en tiempo real entre un contenedor y su entorno debe investigarse más a fondo. Al igual que la comunicación en tiempo real entre contenedores y la gestión de datos compartidos entre contenedores.

- Seguridad de los contenedores. Restringiendo el acceso a los contenedores y la comunicación entre contenedores. Análisis de seguridad de contenedores en tiempo real y gestión de vulnerabilidades para la aceptación en la industria.

- Problemas genéricos que pueden dañar el comportamiento en tiempo real (por ejemplo, cachés compartidos, memoria y E/S). Falta de pruebas de latencia y rendimiento de versiones recientes de un kernel de Linux parcheado. Así como un análisis adecuado de la configuración de los parámetros del kernel de Linux que pueden mejorar el determinismo general de la tarea.

• Las mediciones de la sobrecarga de memoria de la solución de contenedor y evaluar en aplicaciones reales.

Las aplicaciones de la cuarta revolución industrial exigen gran flexibilidad de los sistemas de control de procesos para lograr puntos operativos óptimos que integren toda la cadena de valor de un proceso. Esto requiere la captura de nuevos datos que posteriormente se procesan en diferentes niveles de la jerarquía de los procesos de automatización, con requisitos y tecnologías acordes a cada nivel.

Lo anterior presenta desafíos relacionados con la incorporación de nuevas funcionalidades en los procesos y la interoperabilidad entre ellos. Abordar estas soluciones con base en criterios convencionales, agregando nuevos nodos de hardware cuando se requieren nuevas funciones, conlleva mayores costos de implementación, la reconfiguración de sistemas para soportar los nuevos datos transmitidos a través de la red sin afectar el cumplimiento de plazos en tiempo real y dificultades en el intercambio de información entre plataformas de diferentes fabricantes.

Este contexto requiere un enfoque más flexible que el de agregar nuevos nodos, y es la reconfiguración de los nodos existentes de acuerdo con los requisitos de las nuevas funcionalidades en el sistema. Luego, como consecuencia del aumento de recursos en los nodos hardware, tecnologías como la virtualización hacen que la inclusión de nuevos componentes sea más flexible a muy bajos costos, lo cual es muy conveniente en términos de flexibilidad y el intercambio de información entre componentes. Obviamente, como en los enfoques tradicionales, es necesario verificar el cumplimiento de los plazos de tiempo real.

Por otro lado, la centralización de sistemas de información heterogéneos en un conjunto reducido de servidores y plataformas de SO contribuye a la reducción en la cantidad de hardware y sistemas operativos a administrar, a menudo también trae algún nivel de estandarización, ofreciendo ventajas como administrar menos máquinas individuales, lo que significa menos máquinas para parchear y menos hardware para proteger de riesgos. Mediante el uso de tecnologías de virtualización y la gestión de plataformas de hardware y software heredadas como máquinas virtuales, el diagnóstico de problemas o incluso la re-implementación se pueden gestionar de forma remota y reducir el tiempo de inactividad, que es fundamental en el control de procesos durante todo el día.

3.3.3 Arquitecturas de CPSs para automatización industrial

En lo que respecta a procedimientos de diseño de CPSs es necesario modelar arquitecturas y procedimientos de diseño que permitan proporcionar una descripción técnica y

estándares para la integración e implementación de estos sistemas [129].

Entre las propuestas de arquitectura existentes para el diseño de CPS, la Arquitectura 5C se puede destacar ya que es un modelo de referencia muy conocido con uso generalizado durante el desarrollo de este tipo de sistemas. Está arquitectura consta de 5 niveles que se distribuyen de la siguiente manera [130]:

- El nivel de conexión inteligente, que busca integrar los dispositivos físicos conectados en una red de comunicación.

- El nivel de conversión de datos a información, que busca la conversión de datos de dispositivos monitoreados a información, para comprenderlos y aplicarlos al mundo físico.

- El nivel cibernético, que busca usar la información para la virtualización de los dispositivos, es el nivel responsable de comunicación entre activos.

- El nivel de cognición, que tiene como función monitorear y diagnosticar predicción de fallas y optimización en las tareas de mantenimiento.

- El nivel de configuración, que busca transmitir desde lo virtual hacia el mundo físico, realizando tareas que se ajusten y se adapten automáticamente.

Con la cuarta revolución industrial y el aumento de datos y dispositivos industriales, Se crearon otras arquitecturas de referencia con propuestas relacionadas a los CPS, como RAMI 4.0 en el sector de fabricación para la virtualización de dispositivos en la cadena de valor, e IIRA para la integración y cooperación entre las industrias con un enfoque en IIoT.

La arquitectura RAMI 4.0 es un modelo referente para arquitecturas en las Industrias 4.0 teniendo como finalidad definir las estructuras de comunicación y un lenguaje común dentro de la fábrica con su propio vocabulario, semántica y sintaxis. De esta forma se facilita la integración de IoT y servicios que se requieran. Está arquitectura es representada por un mapa 3d compuesto de 3 ejes, denominados niveles de jerarquía, ciclo de vida del producto y arquitectura por capas. El Eje 1 busca difundir la idea de máquina y sistemas flexibles, funciones distribuidas a través de la red que garantizan la interacción y comunicación entre todos los participantes y productos involucrados. El eje 2 describe activos en la cadena de valor desde su idea, desarrollo y mantenimiento hasta su producción y uso. Finalmente, el eje 3 son las diversas capas de la arquitectura [131].

Por otra parte, IIRA es una arquitectura abierta. El modelo está organizado desde cuatro puntos de vista (empresarial, usabilidad, implementación y funcionabilidad) para identifi-

car y clasificar las preocupaciones de una arquitectura IIoT. Por tanto, las preocupaciones sobre los sistemas IIoT se analizan y abordan sistemáticamente, y luego, sus resultados se documentan como modelos y otra información en los respectivos puntos de vista. El punto de vista empresarial identifica a los participantes y su negocio, visiones, valores y objetivos en los sistemas IIoT. El punto de vista de uso describe la expectativa del sistema IIoT de proporcionar el negocio previsto. El punto de vista de implementación identifica las tecnologías requeridas para implementar los componentes funcionales, su comunicación, esquemas y sus procedimientos de ciclo de vida, tales como topología, estructura , distribución técnica y descripción de componentes. Finalmente, el punto de vista funcional se enfoca en los componentes funcionales y la interrelación e interacción entre ellos y con elementos externos en el medio ambiente. Sin embargo, debe explorarse de una manera más clara una solución de estandarización unificada capaz de garantizar la interoperabilidad entre diferentes sistemas industriales [132].

Como se puede notar existen similitudes entre las arquitecturas, cuyos dominios de IIRA implementan funciones similares con los niveles respectivos de la Arquitectura 5C y capas de RAMI 4.0.

La integración de modelos de amenazas y el seguimiento de las soluciones de seguridad aplicadas se pueden mejorar en estos enfoques. El refinamiento de los modelos durante todo el ciclo de vida del desarrollo de los programas puede mejorarse aumentando el grado de automatización. El modelado clásico de amenazas se puede integrar en un enfoque holístico basado en modelos para diseñar y desarrollar CPS que permitan abordar las amenazas.

3.3.4 Estándares de ciberseguridad en sistema de control industrial

La seguridad de la información es un aspecto crítico y juega un rol importante en la protección del negocio de una organización. Las organizaciones deben proteger su información y activos para mantener el valor y la reputación de la organización. Es más, la gestión eficaz de la seguridad de la información requiere del apoyo y compromiso de la gerencia en la implementación de políticas y procedimientos que permitan el planteamiento de pautas y guías en esta área. De este modo, se han definido una variedad de marcos de seguridad de la información, que incluyen diferentes tipologías. En esta sección se exponen tres marcos de referencia asociado a la seguridad de la información en los sistemas de control industrial.

3.3.4.1. ISO/IEC 27001

Es un framework de seguridad de la información publicado en conjunto por la ISO (International Organization for Standardization, por sus siglas en inglés) y la IEC (International Electrotechnical Commission, por sus siglas en inglés). La finalidad de estas normas es proporcionar una serie de requisitos a las organizaciones para un Sistema de Gestión de Seguridad de la Información. De este modo las compañías establece un sistema de gestión que permiten abordar temáticas relacionadas con la confidencialidad, disponibilidad e integridad de la información, las cuales son pilares de la seguridad de la información.

Las empresas que logren la certificación con el estándar ISO 27001 deben cumplir con diferentes requisitos. El primero de ellos es examinar de forma sistemática los riesgos de la compañía, en donde se tengan en cuenta las amenazas, vulnerabilidades e impactos sobre el sistema. El segundo criterio se relaciona con las etapas de diseñar e implementar una serie de controles de ciberseguridad para solucionar riesgos que no son permitidos. El último criterio es adoptar un proceso que permita asegurar los mecanismos de seguridad siguiendo estándares requeridos de forma periódica [133].

3.3.4.2 IEC 62443

Es un conjunto de normas y estándares enfocado en la seguridad de la información en los sistemas de control y automatización industrial. Tiene como objetivo orientar el funcionamiento de forma segura en los sistemas de automatización industrial, desde el diseño hasta la gestión, teniendo en cuenta la implementación.

Esta norma se centra en el diseño de soluciones seguras teniendo en cuenta la alta disponibilidad, la configuración (información de ingeniería), largos ciclos de vida, operación desatendida, operación en tiempo real y comunicación, así como los requisitos de seguridad en los sistemas de control.

La IEC 62443 complementa la norma ISO 27001, que abarca las principales regulaciones para la seguridad de la información. De este modo, ambas normas permiten un método integral para proteger a las empresas frente a incidentes relacionados con ciberataques y amenazas cibernéticas [133].

El estándar se enfoca en evaluar la seguridad en sistemas de control industrial en cuatro categorías, como se observa en la Fig. 14. En la categoría general se abordan los conceptos generales, modelos y terminologías que definen las arquitecturas de referencia en los sistemas. La categoría de políticas y procedimientos establece el programa para la gestión de la ciberseguridad, de parches y de proveedores para mantener los niveles

de protección a medida que trascurre el tiempo y las amenazas y tecnologías cambian. En el nivel de sistema se establecen las guías para la evaluación de riesgos, los requisitos de seguridad y la tecnología que se tiene al alcance para incrementar los niveles de protección. Finalmente en la categoría de componente se describen los requisitos que se deben tener en cuenta en el diseño de los productos y componentes de forma segura.

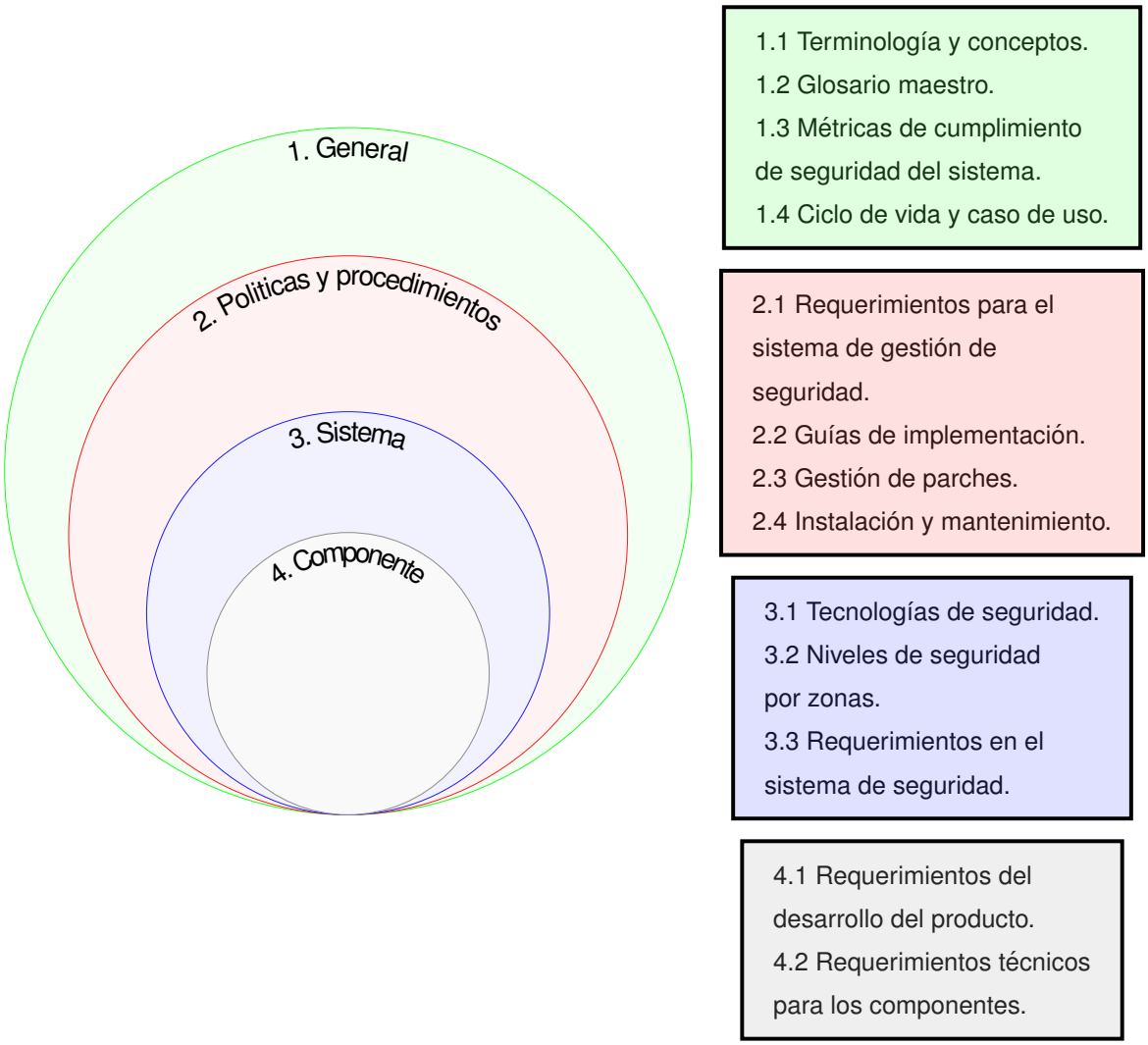


Fig. 14. Categorías en la norma IEC 62443.

3.3.4.3 Arquitectura de ciberseguridad según el NIST

La respuesta a incidentes de seguridad informática se ha convertido en un componente importante de los programas de tecnología de la información (TI). Debido a que realizar una respuesta a incidentes de manera efectiva es una tarea compleja, establecer una capacidad de respuesta a incidentes exitosa requiere una planificación y recursos

sustanciales.

Debido a esto el Instituto Nacional de Estándares y Tecnología ha desarrollado una arquitectura enfocado en la ciberseguridad en donde los controles de ciberseguridad se dividen en cinco categorías, las cuales son identificar, proteger, detectar, responder y recuperar.

Adicionalmente a estas cinco funciones, el marco está conformado por categorías, subcategorías y referencias normativas, como las que se presentaron anteriormente. Todos trabajan en conjunto para gestionar los riesgos asociados a la seguridad de la información. De este modo las propuestas desarrolladas deben estar regidas por este tipo de arquitecturas y normas que permiten generar soluciones con un nivel de

3.4 CONCLUSIONES

Entender la teoría de control y la seguridad asociada a los nuevos esquemas de control, puede conducir a una mejor evaluación de los riesgos y consecuencias de los ciberataques, en donde se planteen diferentes diseños de algoritmos de detección de ataques mediante el monitoreo del sistema físico, además de mejorar el diseño de controladores y arquitecturas resistentes a diversos ciberataques.

Debido a lo que se ha presentado, se ha logrado identificar límites con respecto a la seguridad en estos sistemas frente a algunos ataques, lo cual es importante para identificar el tipo de ataque y de esta manera utilizar el método más apropiado para contrarrestarlo, de tal forma que se garanticen estados seguros en los sistemas. Se cree que comprender las interacciones del control del sistema con el mundo físico conlleve a la capacidad de desarrollar marcos generales y sistemáticos que permitan asegurar los sistemas de control en tres áreas fundamentales:

- Comprender mejor las consecuencias de un ataque para la evaluación de riesgos. Si bien se ha procurado una evaluación de riesgos en estudios previos sobre ciberseguridad para sistemas SCADA, hoy en día hay pocos estudios sobre la identificación de la estrategia de un ataque de un adversario, el cual haya obtenido acceso no autorizado a algunos dispositivos. Se requiere más investigación para comprender el modelo de amenaza con el fin de diseñar defensas apropiadas para asegurar la mayoría de los sensores y actuadores críticos del sistema.

- Diseñar nuevos algoritmos para la detección de los ataques. Al monitorear el comportamiento del sistema bajo control se debería estar en la capacidad de detectar una amplia gama de ataques. Aunque se ha trabajado en el estudio de inyección de falsos datos a

sistemas de control no se han considerado modelos dinámicos del sistema de control. Por tanto, se requiere investigar sobre los modelos usados para describir la dinámica del sistema en conjunto con la teoría de control, para generar estrategias que permitan la detección de intrusos dentro del sistema.

• Diseñar nuevos algoritmos y arquitecturas resistentes a los ataques, lo cual significa diseñar y operar sistemas de control tolerante a ataques cibernéticos sin perder el control de las funciones críticas. La idea general es diseñar sistemas en donde si los atacantes logran evadir algunos mecanismos de seguridad básicos, se enfrenten a varios dispositivos de seguridad específicos en el control que permitan minimizar el daño en el sistema. En particular se requiere investigar en como configurar y adaptar los sistemas de control cuando estos se encuentren bajo ataque, y de esta manera aumentar la resiliencia del sistema.

Tener una visión completa y una comprensión de la capacidad del atacante es un requisito previo para garantizar la seguridad en estos sistemas integrados. Los análisis de seguridad, el diseño y desarrollo de nuevas metodologías, deben tener en cuenta el panorama de amenazas con el fin de identificar los requisitos de seguridad, y de esta manera innovar y aplicar controles de seguridad dentro de un límite de restricciones.

Esta nueva clase de problemas de control requiere de enfoques que incorpore mecanismos de protección en la etapa de diseño para hacer frente a las fallas y, en particular, las vulnerabilidades cibernéticas dentro del sistema y mejorando así la supervisión en red. De este modo, se busca que estos sistemas estén en la capacidad de adaptarse a fallas o eventos, que de un modo u otro pueden comprometer la estabilidad del sistema así como otros aspectos.

El cumplimiento de requisitos de estas aplicaciones plantea un gran reto científico y técnico, en donde se extienden las ciencias computacionales con paradigmas y métodos de teoría de control e ingeniería eléctrica, para lo cual se debe direccionar la investigación en áreas del modelado, programación, compiladores y sistemas operativos, que contribuyan a alcanzar balances entre capacidad de cómputo, consumo de energía y estrategias de protección de información y las estrategias de control.

Se logró identificar que se requieren nuevos métodos de análisis para este tipo de sistemas cuando se encuentran bajo ataque, donde se soporten modelos integrados que detallen el comportamiento de los modelos físicos y computacionales, y su interacción. Además se observa la ausencia de métricas que permitan identificar niveles de seguridad frente a diversos tipos de ataques en sistemas ciberfísicos y sistemas de control en general, así como la ausencia de estrategias que permitan tolerar diversos tipos de

ciberataques en sistemas ciberfísicos, soportadas en nuevas arquitecturas que integren de manera modular estrategias de detección, aislamiento y reconfiguración del sistema.

4. ANÁLISIS DE EFECTOS DE ATAQUES EN REDES DE CONTROL

En esta capítulo se presenta el desarrollo de un modelo que integra los subniveles computacional, energético y de automatización de un caso de un CPS, particularmente una microgrid aislada, lo cual permite el análisis de los efectos de ataques cibernéticos sobre este tipo de sistemas.

El capítulo se organiza de la siguiente manera. En la primera parte se realiza una descripción de las amenazas de seguridad que están presentes en un CPS. En la segunda parte se realiza una descripción y un modelamiento formal de los tipos de ciberataques más comunes que se encuentran en estos sistemas. En la tercera Sección se presenta el desarrollo del CPS para poder realizar el análisis de los efectos que tienen estos ataques en el funcionamiento de estos sistemas. Por último se presentan las conclusiones del capítulo.

4.1 AMENAZAS DE SEGURIDAD SOBRE UN CPS

Estos sistemas como se ha mencionado son objetivo de ataques cibernéticos [8–10, 134–136] los cuales pueden causar daños irreparables al sistema físico que se controla, así como a las personas que dependen de él. La Tabla V permite observar por capa los principales amenazas conocidas sobre los CPSs [137, 138].

TABLA V.
Amenazas sobre CPSs.

Capa Física	Capa de Comunicaciones	Capa Cibernética
Ataques de Denegación de Servicio		
Ataques Físicos	Ataques de enrutamiento	Virus, troyanos
Falla de equipos	Black Hole	Olvidar comandos de control
Interferencia electromagnética	Flooding	Código malicioso
Modificación de datos	Sinkhole	Infiltramiento de datos
Intercepción de datos	Reenvío selectivo	

Por una parte en la capa física, los dispositivos que la componen pueden afectar de manera física el sistema. A los ataques que se llevan a cabo en esta zona se le conocen con el nombre de ataques físicos. También se pueden llevar a cabo amenazas relacionadas con la interferencia electromagnética, que es cuando otro dispositivo compromete la calidad de los datos. Además de esto pueden ocurrir ataques netamente cibernéticos, tales como es el DoS (Deniel of Service por sus siglas en inglés), el cual puede estar

presente en cualquier nodo de la red, implicando que el dispositivo objetivo puede parar de proveer servicios de confianza debido a un sobreconsumo del ancho de banda de la red. La intercepción de datos puede tener lugar en esta capa, donde los canales de comunicación llegan a ser interceptados y se accede a los datos transmitidos, por ejemplo, de los sensores y/o actuadores. Por último los ataques de modificación de datos o ataques de integridad, los atacantes modifican la información que viaja en los canales de comunicación. Como tal, la seguridad en la capa física debe implicar la seguridad física de las infraestructuras, la protección de datos recopilados y de la ejecución de comandos.

En la capa de comunicaciones el principal problema radica en la interrupción o el reenvío de los datos. Un ataque de enrutamiento por ejemplo interfiere en el proceso de enrutamiento, enviando información de enrutamiento errónea. El ataque de agujero negro (Black Hole), el dispositivo comprometido establece el enrutamiento con otros dispositivos, lo que conduce a un evento de perdida de paquetes. En un ataque de inundamiento (flooding), el atacante tiene como objetivo agotar los recursos de los servidores de red a través de un DDOS (Distributed Deniel of Service por sus siglas en inglés). Un ataque de sumidero (Sinkhole), el dispositivo comprometido intentará que todo el posible tráfico llegué hacia él mismo. Finalmente, un ataque de reenvío selectivo, un dispositivo malicioso deliberadamente pierde algunos o todos los paquetes recibidos. Debido a esto, es fundamental garantizar la seguridad en esta capa.

Mientras que en la capa de aplicación o cibernética, puede verse afectada por los virus, troyanos, y otros malware, los ataques realizados en esta capa pueden ocasionar impactos catastróficos en la capa física. La privacidad de los datos transmitidos y almacenados son requisitos fundamentales en esta capa. Esto exige una política estricta de control de acceso y mecanismos de autenticación, para garantizar la protección del sistema.

4.2 TIPOS DE CIBERATAQUES

Como se mencionó anteriormente, los ataques físicos son ataques que pueden estar presentes en la capa física. Dentro de ellos se pueden encontrar las siguientes categorías [139]:

1. **Ataques invasivos:** Tienen como objetivo alterar físicamente la correcta operación en un integrado. Este tipo de ataques requieren de personas con gran experticia.
2. **Ataques semi-invasivos:**Este tipo de ataques tratan de observar el comportamiento del chip del sistema integrado después de que un atacante haya disparado un evento. Un ejemplo de ello pueden ser los denominados ataques de falla, los

cuales inducen una falla en el flujo de cómputo del procesador durante una operación de criptografía y observan el resultado criptográfico a medida que la falla se propaga. Este resultado permite que un chip incrustado no protegido pueda ser usado para deducir información sensible. Este tipo de fallas se pueden inducir mediante fallas de energía o cambios en los tiempos de reloj, variaciones extremas de temperatura, etc.

3. **Ataques no invasivos:**Usan la explotación de las características del integrado, tales como la disipación de potencia, tiempo de cálculo, entre otras, para poder extraer información sobre los datos procesados.

Por otro lado se encuentran los ataques cibernéticos, dentro de los cuales los más frecuentes son los ataques de DoS, ataques de repetición y ataques de engaño o integridad [140]. Los ataques tipo DoS son también conocidos como ataques Jamming, y son estrategias que ha menudo son usadas por los atacantes para afectar la transmisión de las medidas o señales de control, al ocupar el recurso de la red de manera que el rendimiento del sistema puede deteriorarse tanto como sea posible. Los ataques de repetición son ataques muy comunes y naturales por los atacantes que desconocen la dinámica de los sistemas. Consiste en registrar lecturas de los sensores y actuadores comprometidos por una cierta cantidad de tiempo para repetir este set de información en posteriores intervalos. Mientras que los ataques de engaño o integridad, también conocidos como ataques de inyección de falsos datos, son los ataques más generales y son considerados como los más peligrosos en este tipo de sistemas, porque los atacantes pueden ingresar datos maliciosos que degradan el rendimiento general de los sistemas.

El esquema de un CPS bajo ataque se observa en la Fig. 15. Se observa la interacción que existe entra las tres capas, con cada uno de los elementos que se ven involucrados en el control del proceso así como las afectaciones que puede llegar a generar el atacante $\mathcal{A} = \{a_k^u, a_k^y\}$ que afecta tanto la acciones de control como las medidas de las variables del proceso.

Las mediciones de las variables del proceso y los valores de las acciones de control son fundamentales para el buen funcionamiento de un sistema de control, y su modificación mediante ciberataques puede producir inestabilidad en el sistema de control [140, 141]. Un ciberataque que resulta de la manipulación de las variables del proceso es referido como un ataque de integridad y un ataque que resulta en una pérdida prolongada de estas señales es llamado como un ataque tipos DoS, los cuales pueden ser modelados por las Ecuaciones (30) y (31), respectivamente.

$$\overline{y}_i(k) = y_i(k) + y_i(k)^a \tag{30}$$

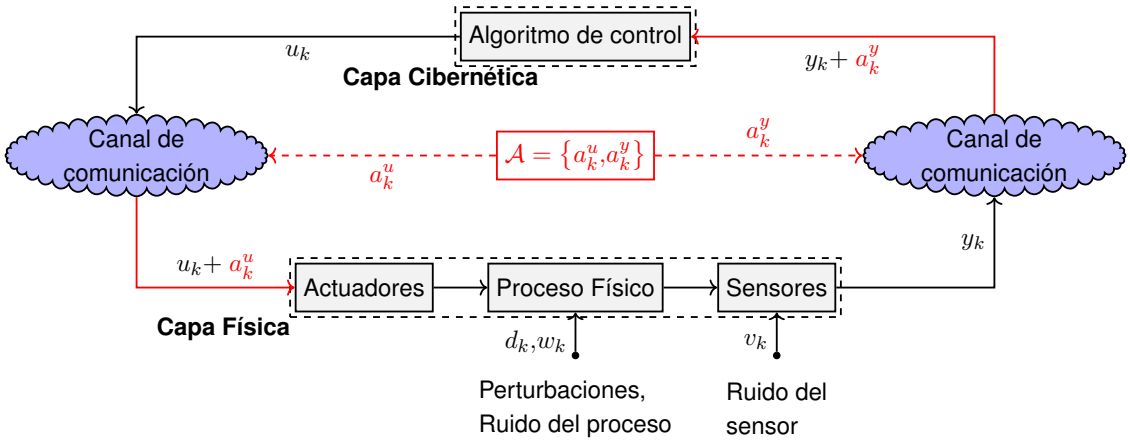


Fig. 15. Sistema de control en un CPS bajo ataque.

$$\overline{y}_i(k) = y_i(k)_{t_{s-1}} \tag{31}$$

Donde $\overline{y}_i(k)$ corresponde a la medición del i-ésimo sensor que llega al controlador en el tiempo k, $y_i(k)$ corresponde a la medición del i-ésimo sensor antes de ser transmitida al controlador en el tiempo k, y $y_i(k)^a$ es un vector inyectado por los atacantes que cambia la medida $y_i(k)$ en el tiempo k. Para el ataque DoS $y_i(k)_{t_{s-1}}$ corresponde a la medición antes del inicio del ataque DoS. El intervalo de tiempo para la ocurrencia del ataque se define por $\tau_a = [t_s \ t_e]$. Del mismo modo esta idea se puede extender a las acciones de control.

Todas estas amenazas mencionadas han ocurrido en diferentes tipos de CPSs. A nivel de sistemas de control industrial una gran número de ataques han explotado las vulnerabilidades que se presentan en los protocolos de comunicación [142, 143], en el sector energético las smart grids han sufrido ataques de Denegación de Servicio [144] así como ataques donde se inyecta información falsa [145], en dispositivos médicos se han llevado a cabo experimentos donde se llevan a cabo ataques de replica [146], en el sector de los carros inteligentes se han evaluado el impacto que tiene ataques del tipo de Denegación de Servicio [147] así como manipulación de la información registrada por los sensores [147, 148], entre otras situaciones que han sido reportadas.

Se pueden llegar a plantear diferentes escenarios de ataques [149] dependiendo del tipo de sistema que este involucrado, así mismo el impacto puede trascender a diferentes niveles, desde no garantizar la estabilidad del sistema hasta tener consecuencias que afecten el entorno económico y ambiental del proceso [6].

Para el desarrollo de la propuesta, se asume que cualquier sensor puede ser afectado por cualquier tipo de ataque, de integridad o de DoS, dado que son los ataques con ma-

yor frecuencia así como los que generan un gran impacto negativo en el funcionamiento de los sistemas. Además, los ataques pueden ocurrir en cualquier momento en varias partes del sistema. Esta última premisa es significativa porque los ataques simultáneos son menos discutidos en trabajos anteriores; así, dependiendo del tipo de ataque realizado al sistema, la salida (3) puede tomar la forma de (30) o (31).

4.3 EFECTOS DE ATAQUES EN SISTEMAS DE CONTROL – CASO MICROGRID

En esta sección se analiza el efecto de ataques de integridad y los ataques de denegación del servicio en una microgrid. En general, una microgrid es un sistema de potencia eléctrica que integra fuentes de generación distribuida conectadas a las redes de distribución de forma independiente. Se caracteriza por trabajar en baja y media tensión, actúa como una sola entidad controlable respecto a la red, puede conectarse y desconectarse de la red de distribución regional para permitirle operar en modo conectada a ésta o en modo aislada, y genera energía en AC, DC o en forma mixta. Estos sistemas integran los desarrollos en ingeniería eléctrica, almacenamiento energético y los avances de las tecnologías de la información y comunicación (TIC), para la generación, transmisión, distribución, almacenamiento y comercialización de energía, incluyendo las energías alternativas; posibilitando la integración de las actividades de coordinación de protecciones, control, instrumentación, medida, calidad y administración de energía, bajo un sistema de gestión que busca el uso eficiente y racional de la energía.

La energía se genera a partir de pequeñas fuentes que se encuentran lo más cerca posible a las cargas, y generalmente se soportan en fuentes renovables o no convencionales. Uno de los mayores inconvenientes de estas fuentes de generación es la disponibilidad del recurso, pues no es posible regular variables como la radiación solar, flujos de agua en ríos o la energía eólica, debido a ello el problema es mayor cuando se opera la microgrid en modo aislado y se tienen requisitos de abastecimiento continuo, por lo que asegurar la estabilidad del sistema y la cobertura de la demanda se convierten en desafíos para el diseño de los sistemas de control, y de allí la necesidad de integrar generadores cuyo abastecimiento se pueda controlar, como es el caso de los generadores diésel.

La integración de fuentes de generación distribuida presenta la necesidad de contar con sistemas de control que contemplen la optimización del sistema y su correcto funcionamiento, para lo que es indispensable contar con sistemas de comunicación que permitan tener la información en tiempo real, de forma que se pueda asegurar la estabilidad del sistema y la confiabilidad del servicio [150]

4.3.1 Modelado de la microgrid

El modelo de microgrid analizada se presenta en la Fig. 16. Está integrada por tres fuentes de generación, biodiesel, fotovoltaico y diésel. Cada fuente de generación tiene conectado un inversor monofásico DC/AC, la señal generada por cada inversor es conectada al punto de acople común (PCC, point of common coupling por sus siglas en inglés) a través de una línea de transmisión. Existe un controlador predictivo, del tipo DMC, que recibe la medición de la energía inyectada en la carga y establece los niveles de referencia de los inversores de tal forma que se garantice el abastecimiento establecido en la carga. Finalmente, un algoritmo de programación lineal define las referencias al algoritmo DMC de tal forma que se minimicen una función de costo que incluye índices relacionados con afectaciones medioambientales y económicas sobre la comunidad que se abastece de energía por parte de la microgrid. El modelo también contempla el intercambio de información entre los controladores a través de redes de comunicación, y el tiempo de cómputo de los algoritmos implementados en cada procesador.

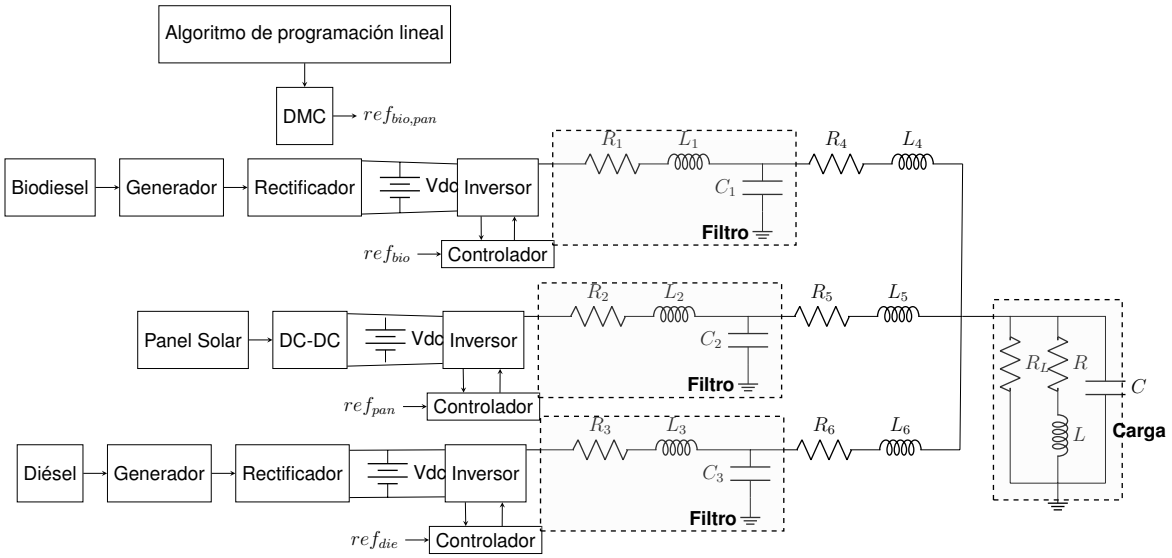


Fig. 16. Esquemático de microgrid aislada.

Dos de los tres inversores DC/AC se comportan como fuentes de corriente (CSI), mientras que el tercero se comporta como una fuente de tensión (VSI). El inversor que actúa como fuente de tensión recibe una referencia arbitraria ya que se trata de un sistema aislado, mientras que los inversores de corriente reciben una referencia de potencia activa y reactiva, la cual relacionan internamente a una referencia de corriente. Se implementaron lazos de control resonante, los cuales corrigen los errores de tal forma que cada inversor suministre a la red la cantidad de energía requerida. Se consideró una carga

lineal tipo RLC.

El inversor que actúa como fuente de tensión se soporta en Diésel, debido a que este recurso puede ser controlado con mayor facilitad que los asociados a fuentes alternativas de energía, y de esta forma se puede garantizar una operación continua de este inversor en ausencia de fallos en el sistema.

En las siguientes secciones se presentan los diferentes niveles que constituyen la microgrid.

4.3.1.1. Modelo de las fuentes de generación

En este trabajo se consideraron tres fuentes de generación, biodiesel, fotovoltaico y diésel. A continuación, se presentan los modelos considerados para cada una.

- **Modelo del panel fotovoltaico**

El modelo del panel fotovoltaico utilizado se detalla en [151]. Para modelar el componente eléctrico del módulo de módulo solar híbrido (PV/T), se analizó el circuito equivalente de una celda fotovoltaica, Fig. 17.

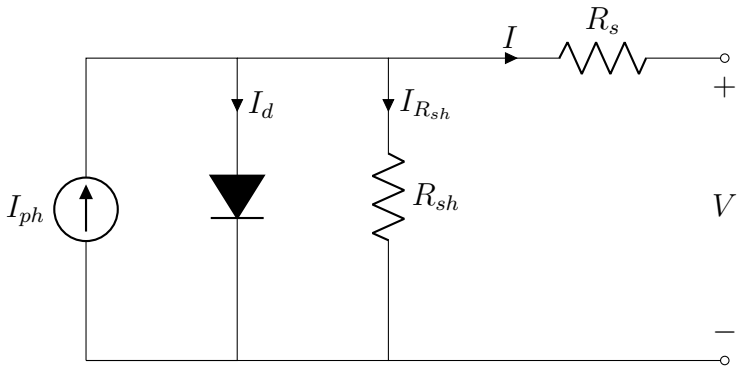


Fig. 17. Circuito eléctrico de una celda fotovoltaica.

La resistencia en serie R_s, representa las resistencias al paso de corriente en el material semiconductor, el metal de la red, los contactos y el bus colector de corriente; la resistencia paralela R_{sh}, conocida como resistencia shunt, representa el comportamiento no lineal del semiconductor. La ecuación (32) describe el comportamiento de la corriente de salida de la celda.

$$I = I_{ph}N_p - I_d - I_{R_{sh}} \tag{32}$$

Donde I es la corriente de salida de la celda, I_{ph} es corriente fotogenerada, I_d es la co-

rriente del diodo, I_{sh} es la corriente de fuga de la resistencia en paralelo, N_p es el número de módulos fotovoltaicos conectados en paralelo, $I_{R_{sh}}$ es la corriente por la resistencia en paralelo. R_{sh} and R_s son parámetros del módulo fotovoltaico que generalmente no son proporcionados por los fabricantes. Para aproximar estos parámetros se usan las ecuaciones propuestas en [152].

Para simular el comportamiento de la celda solar se utilizaron los parámetros de un módulo PVT-M PREMIUM [153].

- **Modelo del motor diésel**

La propuesta presentada en [154] se utilizó para representar la dinámica del motor diésel. Desde la perspectiva de un sistema de control, un motor diésel puede describirse como un sistema de retroalimentación de velocidad. Dada una referencia del operador, el regulador del motor, que también funciona como sensor, reconoce la diferencia entre la velocidad real y la deseada, y regula la alimentación de combustible para mantener la velocidad del motor en el valor especificado.

El sistema actuador de combustible a menudo se representa como una red de desfase de primer orden, con una ganancia K_2 y una constante de tiempo τ_2. La Ecuación (33) muestra la dinámica de este actuador y considera la constante de control K_3. La salida del actuador es el flujo de combustible $\phi(s)$ y su entrada es la corriente $I(s)$.

$$\phi(s) = \frac{K_3 K_2}{1 + \tau_2 s} I(s) \tag{33}$$

El flujo $\phi(s)$ es transformado a un torque mecánico $T(s)$ con un tiempo de retardo τ_1 y el par motor constante K_1, detallado en la Ecuación (34).

$$T(s) = \phi(s) K_1 e^{-\tau_1 s} \tag{34}$$

Un volante muestra los efectos dinámicos de la inercia del motor, la velocidad angular del volante es ω_w, el coeficiente de fricción viscosa es ρ. Este modelo incluye un integrador con la constante de aceleración del volante J que se utiliza para filtrar una parte importante de los efectos de las perturbaciones y el ruido. Se agregó un integrador entre la señal de referencia ref y el actuador del motor, necesario para eliminar el error de estado estable. El esquema del sistema se muestra en la Fig. 18 [154].

Los valores de los parámetros completos para el sistema de motor diésel se muestran en la Tabla VI [155].

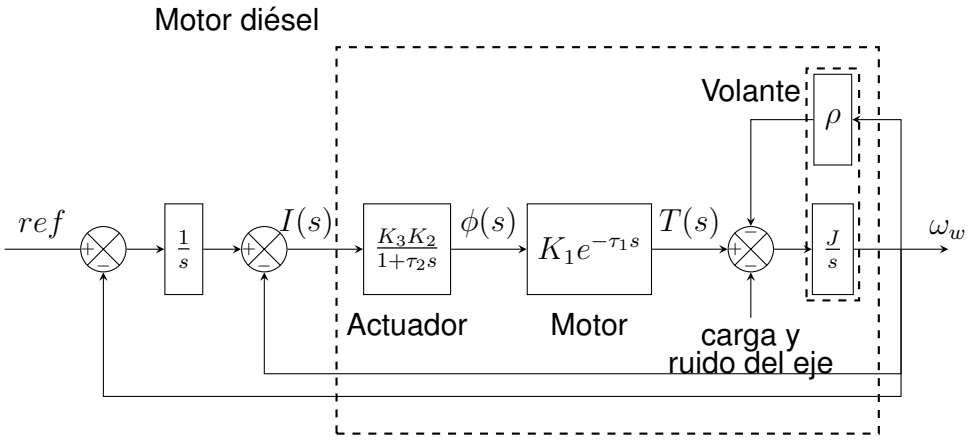


Fig. 18. Diagrama de bloques del sistema de motor diésel.

TABLA VI.
Parámetros del sistema de un motor diésel.

Parámetros	Valor nóminal
K_1 (pu)	$1{,}15$
K_2 (pu)	1
K_3 (pu)	1
τ_1 (s)	$0{,}5$
τ_2 (s)	$0{,}125$
$J(s^{-1})$	$0{,}3$
ρ (pu)	$0{,}1$

- **Modelo de generador de biodiesel**

El modelo de generador de biodiesel se centró en representar la dinámica de generación. En este caso, la propuesta descrita en [156] se utilizó. Este modelo comprende una microturbina acoplada a un generador síncrono de imanes permanentes.

4.3.2 Niveles de control de la microgrid

En este trabajo se implementaron dos niveles de control. En el primer nivel, denominado control primario, se implementaron algoritmos de control proporcional-resonante en cada uno de los inversores asociados a las fuentes de generación [157]. En el segundo nivel, conocido como control secundario, se implementó un controlador MPC del tipo DMC, el

cual establece las referencias a los reguladores de primer nivel con el fin de alcanzar las referencias deseadas en la carga. Adicionalmente se implementó un algoritmo de optimización de programación lineal, el cual establece las referencias al control secundario para minimizar una función de costo que considera aspectos de influencia de la microgrid a la comunidad donde esté instalada. A continuación se detalla cada uno de ellos.

4.3.2.1. Control de los inversores

El modelo del inversor se presenta en la Fig. 19. El inversor tiene como única entrada la señal D, la cual es usada por el sistema de modulación para controlar cada uno de los cuatro transistores de un puente completo. El inversor por sí mismo solo puede generar señales cuadradas a su salida, por lo que para obtener una sinusoidal suave es necesario conectar un filtro pasa-bajo que elimine las componentes de alta frecuencia de la señal cuadrada, lo cual se implementó por medio de una inductancia y una capacitancia.

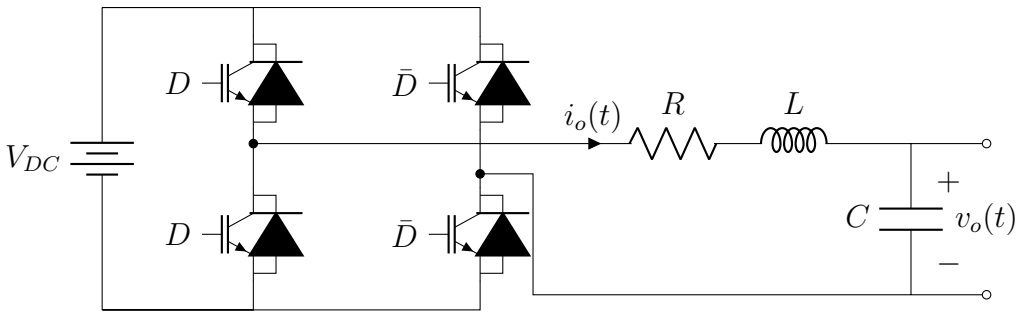


Fig. 19. Inversor de puente completo conectado a un filtro LC y una carga resistiva.

Debido a que los inversores deben estar sincronizados para controlar la corriente del inversor, es necesario no solo medir la corriente de salida sino también medir la tensión de red, que es común para todos los inversores. Dado que la corriente se puede determinar conociendo la potencia y el voltaje, es posible exigir cantidades de potencia más fácilmente para un inversor CC/CA.

Se seleccionó un controlador resonante para la implementación del lazo de control de corriente debido a las limitaciones asociadas a los controladores PID para eliminar por completo el error de estado estable en los inversores.

Para determinar cuál debe ser la corriente de referencia se empleó la teoría de potencia instantánea. La teoría de potencia instantánea fue desarrollada originalmente para sistemas trifásicos, para los cuales se lleva a cabo una transformación llamada $\alpha - \beta - 0$.

Esta transformación permite representar las tres tensiones trifásicas por medio de dos tensiones alternas con lo que se simplifica la representación del sistema. Considerando lo anterior, la ecuación proveniente de la teoría de potencia instantánea que relaciona las corrientes, las tensiones y las potencias se muestra a continuación, Ecuación (35).

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{1}{v_\alpha^2 + v_\beta^2} \begin{bmatrix} v_\alpha & v_\beta \\ v_\beta & -v_\alpha \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} \tag{35}$$

Debido a que el sistema implementado es monofásico, para implementar el cálculo de potencia instantánea es necesario llevar a cabo una simplificación. En un sistema trifásico simétrico y balanceado las componentes en α y en β son dos sinusoidales cuyo ángulo es $90°$ entre sí, de esta forma se establece la tensión medida como la componente en α, y la componente en beta se genera de forma artificial desfasando $90°$ la tensión medida [158]. Teniendo en cuenta lo anterior se utiliza la ecuación presentada, en la cual se toma una referencia de potencia activa y reactiva que se desea inyectar a la red eléctrica, y también se ingresa la tensión medida, a partir de lo cual se calcula la corriente. De esta forma la fase de la corriente se sincroniza con la tensión de la red de acuerdo a la cantidad de activos y reactivos deseados.

De este modo la Ecuación (35) se utiliza para establecer la referencia de potencia activa y reactiva a inyectar en la red eléctrica y la tensión medida también se utiliza para calcular la corriente. De esta forma, la fase de la corriente se sincroniza con la tensión de la red según la cantidad de potencia activa y reactiva deseada, como se observa en la Fig. 20.

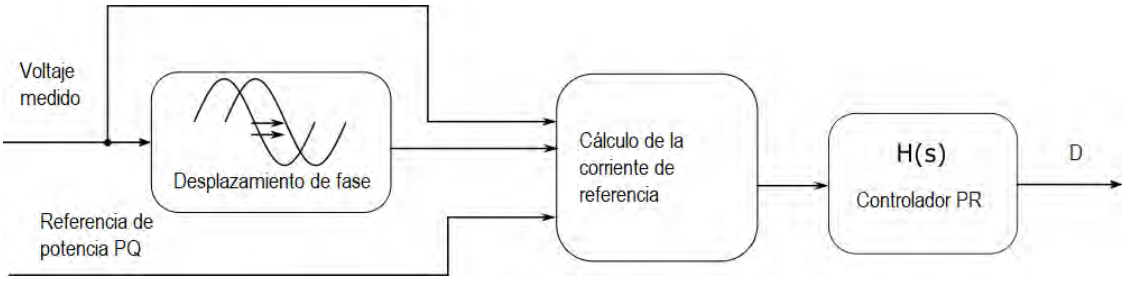


Fig. 20. Lazo de control de corriente implementado en cada CSI.

En la microgrid eléctrica es necesario que uno de los convertidores establezca la tensión de la red (VSI), de forma que los convertidores que operen como fuente controlada de corriente puedan hacer su respectivo cálculo de la corriente de referencia. Sin un convertidor que opere como fuente controlada de tensión la operación del sistema tal y como se tiene planteado no sería viable.

Debido a que se desea implementar un sistema aislado, la referencia de tensión puede ser arbitraria en el sentido que no debe sincronizarse con ninguna otra señal. En este caso se estableció una sinusoidal de $60Hz$. En la Fig. 21 se muestra el esquema de control empleado para el inversor de tensión.

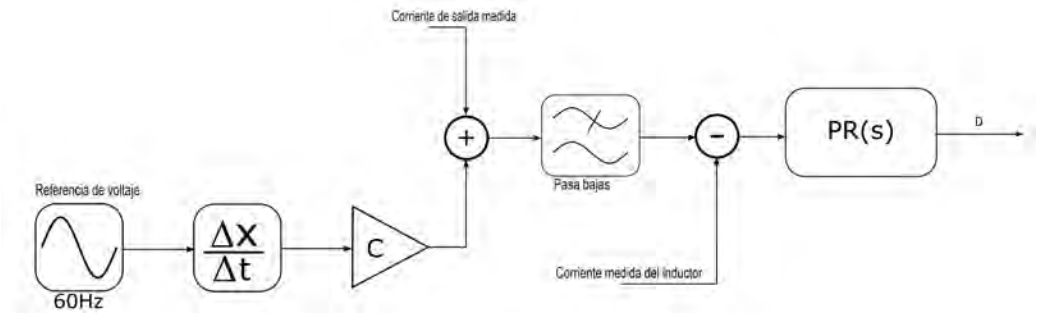


Fig. 21. Lazo de control de tensión en el VSI.

El control de tensión se implementa usando el mismo esquema de control de corriente presentado anteriormente. La tensión de salida del inversor se puede controlar a partir de la corriente del capacitor del filtro LC. La derivada de la tensión multiplicada por la capacitancia es igual a la corriente del capacitor. Al sumar la corriente de referencia del capacitor con la corriente de salida, se obtiene una corriente de referencia para el inductor. Esta corriente es filtrada a través de un filtro pasa-bajo para eliminar ruido. Una vez se obtiene la corriente de referencia se resta con la corriente medida del inductor para obtener el error de corriente. De esta forma al hacer cero el error en la corriente del inductor de manera indirecta se está controlando la corriente del capacitor, la cual a su vez permite controlar la tensión de salida del inversor. El error de corriente en el inductor se pasa a un controlador proporcional resonante para obtener la acción de control $D(t)$ para el inversor DC/AC. De esta forma se logra implementar un lazo de control de tensión mediante un lazo de control de corriente.

4.3.2.2 Estrategia de control de segundo nivel

En la Fig. 22 se establece el modelo de control de segundo nivel para la microgrid propuesta. En este nivel se analiza el sistema compuesto por dos unidades de generación distribuidas (DG), una a partir de paneles fotovoltaicos y la otra proveniente del biodiesel. Cada unidad de generación se conecta a través de un punto de acople común, llamado PCC, a través de un inversor y un filtro tipo LCL, cuya función es reducir los efectos producidos por las diferencias de potencial en los voltajes de salida de cada inversor [159]. Los bloques llamados “Control DG1 ” y “Control DG2”, son los sistemas de control para cada uno de los inversores conectados a su respectiva fuente de generación, cuya

función es regular la tensión y la frecuencia en el PCC.

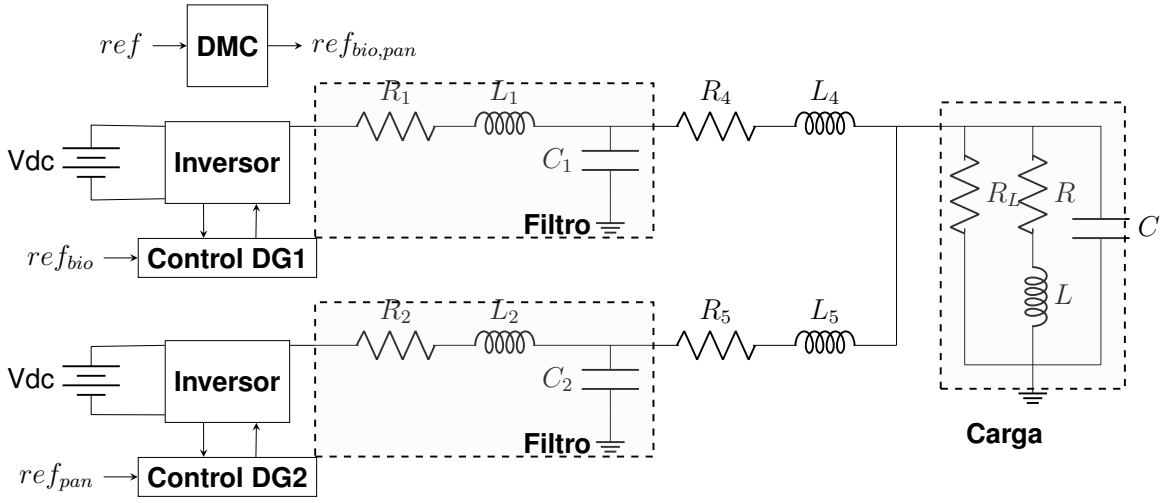


Fig. 22. Modelo de control de segundo nivel.

Un algoritmo de control predictivo basado en modelo (MPC), del tipo DMC, recibe las referencias de potencia deseadas y ajusta los valores PQ de corriente para cada uno de los inversores, los cuales se establecen como referencia para los lazos de control primario de cada CSI (Current Source Inverter). Un algoritmo de optimización por programación lineal establece la mejor combinación de potencias activa y reactiva para minimizar una función de costo en la que se relacionan costos de generación de cada fuente, impactos ambientales, entre otros; y estos son los valores que se establecen como referencias en el MPC.

La representación en espacio de estado continuo del sistema es:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x}(\mathbf{t}) + B\mathbf{u}(\mathbf{t}) \\ \mathbf{y} &= C\mathbf{x}(\mathbf{t}) + D\mathbf{u}(\mathbf{t}) \end{aligned} \tag{36}$$

Usando leyes fundamentales de sistemas eléctricos, se determina un modelo lineal del sistema:

$$\begin{array}{lll} \dot{x}_1 = -\frac{R_1}{L_1}x_1 - \frac{x_8}{L_1} + \frac{u_1}{L_1} & \dot{x}_2 = -\frac{R_2}{L_2}x_2 - \frac{x_9}{L_2} + \frac{u_2}{L_2} & \dot{x}_3 = -\frac{R_3}{L_3}x_3 - \frac{x_{10}}{L_3} + \frac{u_3}{L_3} \\ \dot{x}_4 = -\frac{R_4}{L_4}x_4 + \frac{x_8}{L_4} - \frac{x_{11}}{L_4} & \dot{x}_5 = -\frac{R_5}{L_5}x_5 + \frac{x_9}{L_5} - \frac{x_{11}}{L_5} & \dot{x}_6 = -\frac{R_6}{L_6}x_6 + \frac{x_{10}}{L_6} - \frac{x_{11}}{L_6} \\ \dot{x}_7 = -\frac{R}{L}x_7 + \frac{x_{11}}{L} & \dot{x}_8 = -\frac{x_1}{C_1} - \frac{x_4}{C_1} & \dot{x}_9 = -\frac{x_2}{C_2} - \frac{x_5}{C_2} \\ \dot{x}_{10} = -\frac{x_3}{C_3} - \frac{x_6}{C_3} & \dot{x}_{11} = \frac{x_4+x_5+x_6-x_7}{C} - \frac{x_{11}}{R_L C} & \end{array} \tag{37}$$

El vector de estados es $x = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}]^T$, el vector de entradas es $u = [u_1\ u_2\ u_3]^T$, y el vector de salidas es $y = [x_4\ x_5\ x_6\ x_{11}]^T$. Los tres primeros

estados se refieren a las corrientes que circulan por cada una de las bobinas en el filtro de salida del inversor. Los estados $4, 5$ y 6, se refieren a las corrientes que circulan por un elemento inductor usado para el modelado de la línea de transmisión en cada uno de los inversores. El séptimo estado se refiere a la corriente que circula por un elemento inductor ubicado en el sitio de la carga. Los estados $8, 9$ y 10, se refieren la tensión en el elemento capacitor del filtro de salida de cada inversor, respectivamente. El estado 11 se refiere a la tensión en la carga. Las señales de entrada son las señales provenientes de la salida del puente H, de cada inversor respectivamente. Las salidas se transmiten a través de una red comunicaciones hacia los inversores.

Los parámetros del sistema son los elementos resistivos, inductivos y capacitivos usados en los filtros de salida de cada generador $(R_{1,2,3}, L_{1,2,3}$ y $C_{1,2,3})$, respectivamente. Los elementos resistivos e inductivos que son usados para el modelo de la línea de transmisión de cada inversor hacia la carga $(R_{4,5,6}, L_{4,5,6})$. Finalmente, los elementos en la carga son R,L,C y R_L. Los parámetros usados para la simulación se presentan en la Tabla XIII.

TABLA VII.
Parámetros del sistema.

Parámetro	**Símbolo**	**Valor**
Resistencia del filtro de salida del inversor del generador de Biomasa, Renovables y Diesel (respectivamente).	R_1, R_2, R_3	$0{,}1\Omega, 0{,}2\Omega$,$0{,}001\Omega$
Inductancia del filtro de salida del inversor del generador de Biomasa, Renovables y Diesel (respectivamente).	L_1,L_2, L_3	$40mH, 30mH, 5mH$
Capacitancia del filtro de salida del inversor del generador de Biomasa, Renovables y Diesel (respectivamente).	C_1, C_2, C_3	$22\mu F, 22\mu F, 610\mu F$
Resistencia de línea de transmisión del generador de Biomasa, Renovables y Diesel (respectivamente).	R_4, R_5, R_6	$0{,}005\Omega, 0{,}0012\Omega, 0{,}001\Omega$
Inductancia de línea de transmisión del generador de Biomasa, Renovables y Diesel (respectivamente).	L_4, L_5, L_6	$35\mu H$,$22\mu H$,$10\mu H$
Elementos en la carga.	R, R_L, L, C	$10\Omega, 10\Omega, 5{,}2H, 20{,}000\mu F$

La acción de control se obtiene resolviendo el siguiente problema de optimización, Ecuación (38).

$$\begin{aligned} \text{mín} \quad & \sum_{k=1}^{Hp} ||r - x_{DG1}(k)||_Q + ||U_1||_R \\ \text{mín} \quad & \sum_{k=1}^{Hp} ||r - x_{DG2}(k)||_Q + ||U_2||_R \\ \text{s.t.} \quad & -I_1 < U_1 < I_1 \\ & -I_2 < U_2 < I_2 \end{aligned} \tag{38}$$

La función objetivo se utiliza para hallar los valores óptimos de las acciones de control $U_{1,2}$ que permiten reducir el error en estado estacionario. Estas a su vez son las referencias para los controladores de corriente de los inversores de cada generador, asegurando que la predicción de los estados futuros del sistema sea cercana al modelo del sistema, y restringiendo estas referencias a $\pm I_{1,2}$, las cuales representan los valores máximos de corriente de salida que pueden suministrar cada generador.

En el problema de optimización, la señal r es la referencia y el parámetro Hp indica la ventana de predicción, para este caso se seleccionó igual 3.

4.3.2.3 Optimización económica

Como estrategia para mitigar los impactos de la microgrid se implementó un nivel de optimización económica, el cual se encarga de establecer las referencias al MPC. El planteamiento es el siguiente:

- $P_{carga} = P_{renovables} + P_{no\,renovables}$, donde P es la potencia.

- Se desprecia los costos económicos y ambientales de los generadores de las energías alternativas después de estar instalados. .

- Función de costo (generadores con fuentes no alternativas) :

$$\begin{aligned} F_{cA} &= A_1 P_{Gen_1} + A_2 P_{Gen_2} + \cdots + A_n P_{Gen_n} \\ F_{cB} &= B_1 P_{Gen_1} + B_2 P_{Gen_2} + \cdots + B_n P_{Gen_n} \\ &\vdots \\ F_{cX} &= X_1 P_{Gen_1} + X_2 P_{Gen_2} + \cdots + X_n P_{Gen_n} \\ F_{cT} &= F_{cA} + F_{cB} + \cdots + F_{cX} = \sum_{k=1}^{n} K_{kp} P_{Gen_k} \end{aligned} \tag{39}$$

- Suponiendo dos generadores: Diésel y Biodiesel:

$$\begin{aligned} 0 \leq P_{Gen_D} \leq P_{Gen_{D_{max}}} \\ 0 \leq P_{Gen_{Bio}} \leq P_{Gen_{Bio_{max}}} \end{aligned} \tag{40}$$

- $P_{Gen_D} + P_{Gen_Bio} = P_{no\,renovables} \geq 0$
- $F_C = K_1 P_{Gen_D} + K_2 P_{Gen_{Bio}}$
- $K1 \geq 0\ K2 \geq 0 \Rightarrow Fc$ es definida positiva.
- Reemplazando $P_{Gen_Bio} \Rightarrow$

$$\begin{aligned} F_C &= K_1 P_{Gen_D} + K_2 (P_{no\,renovables} - P_{Gen_D}) \\ F_C &= K_2 P_{no\,renovables} + (K_1 - K_2) P_{Gen_D} \end{aligned} \tag{41}$$

Se tienen los siguientes casos:

- $K_1 = K_2 \Rightarrow F_C = K_2 P_{no\,renovables}$
- $K_1 > K_2$
  - $\Rightarrow F_C\ creciente\ \forall\ P_{Gen_D}$
  - $\Rightarrow F_C$ mínimo cuando $P_{Gen_D} = 0$
- $K_1 < K_2$
  - $\Rightarrow F_C\ decreciente\ \forall\ P_{Gen_D}$
  - $\Rightarrow F_C$ mínimo cuando $P_{Gen_D} = P_{Gen_{D_{max}}}$

4.3.3 Resultados de simulación

El sistema propuesto también considera el intercambio de datos entre controladores mediante redes de comunicación y el tiempo de cálculo de los algoritmos implementados en cada procesador de acuerdo con un kernel con política de prioridad de tareas. La simulación de esta microgrid con perfiles de demanda se realiza mediante matlab y simulink. El esquema implementado se muestra en la Fig. 23.

El comportamiento del sistema en condiciones normales de funcionamiento se presenta en la Fig. 24.

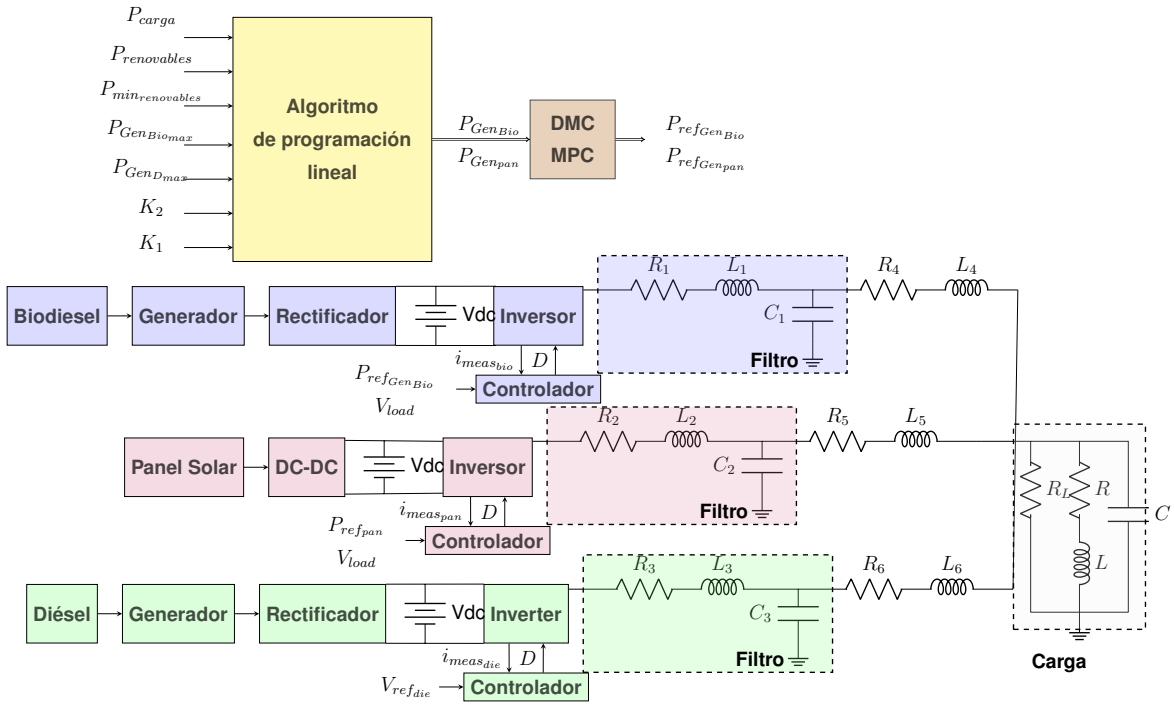


Fig. 23. Diagrama eléctrico de la microgrid.

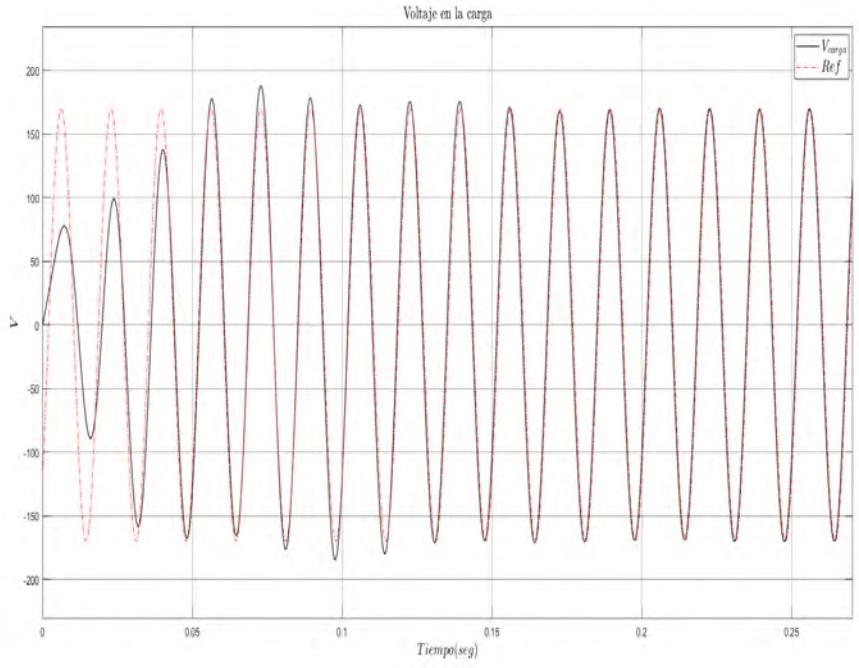


Fig. 24. Respuesta temporal de la tensión de carga.

Los resultados de simulación del modelo obtenido no solo permiten evaluar situaciones relacionadas directamente con la generación de las señales de energía, sino que también posibilitan el análisis de situaciones que se pueden presentar en el comportamiento de la arquitectura computacional que soporta la aplicación, como por ejemplo la ocurrencia de ciberataques que afecten las estrategias de control, para lo cual se implementaron dos casos. Se analizaron dos casos, en el primer caso se agregaron nodos a la red de comunicaciones que la saturaron, ocasionando retrasos significativos en el tiempo de transmisión de las señales simulando un ataque tipo DoS, Fig. 25; en el segundo caso se implementaron tareas en los nodos que modificaron los valores de las señales simulando un ataque de integridad, Fig. 26.

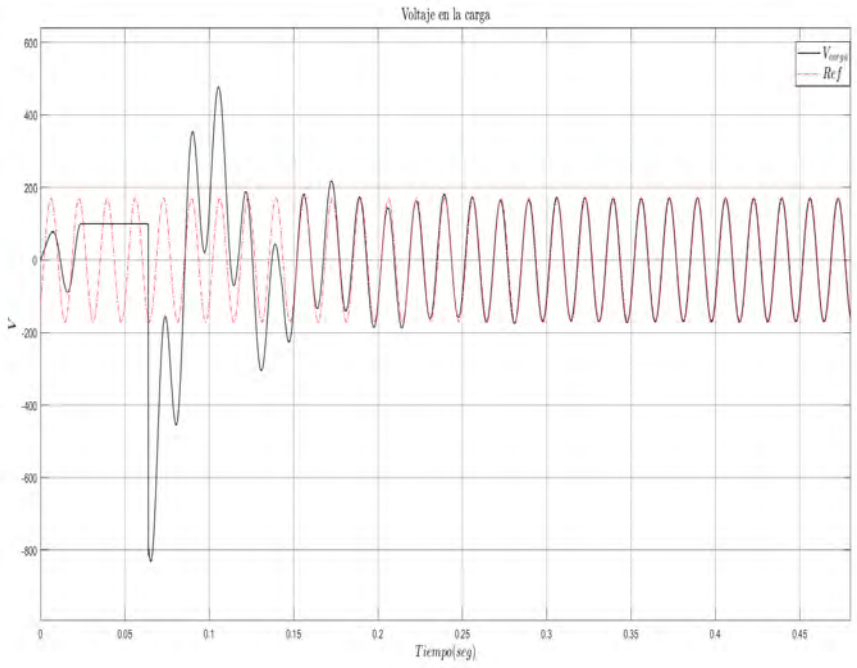


Fig. 25. Respuesta temporal de la tensión de carga bajo ataque DoS.

Los resultados en la respuesta temporal del sistema al generarle ataques de integridad y el de denegación del servicio permitieron observar que hay error permanente en estado estacionario en las variables a controlar durante la duración de los ataques, debido a que el controlador calcula acciones de control indebidas como consecuencia de la corrupción de las variables del proceso medidas que fueron atacadas, en este caso en particular el ciberataque se lleva en el nodo que mide la tensión en la carga. Adicionalmente, se observaron transitorios diferentes dependiendo del tipo de ataque que se efectúa sobre el sistema. Siendo el más agresivo el que ocurre durante el ataque del tipo DoS. Tener en cuenta que este ataque tiene impacto cuando hay cambios en la variable de medida,

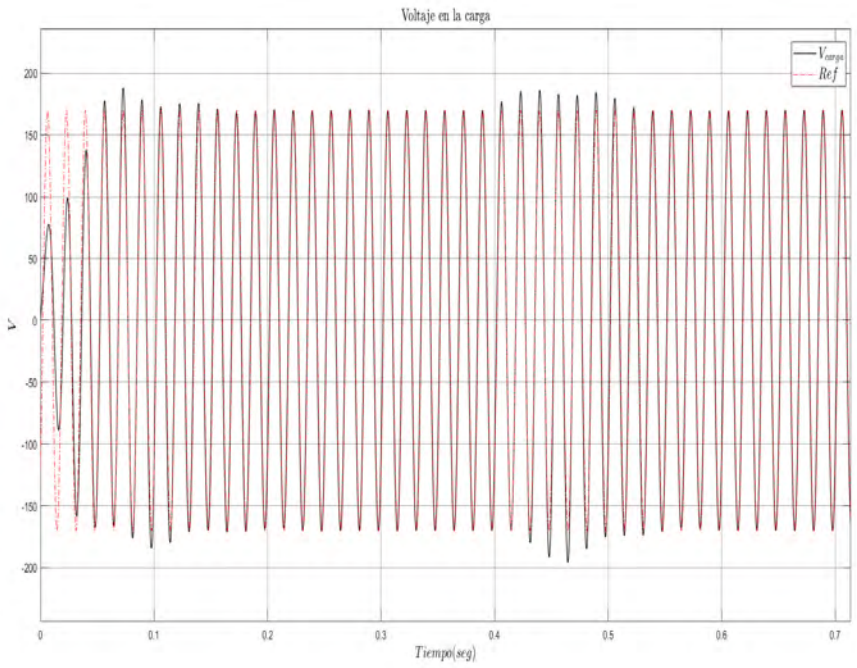


Fig. 26. Respuesta temporal de la tensión de carga bajo ataque de integridad.

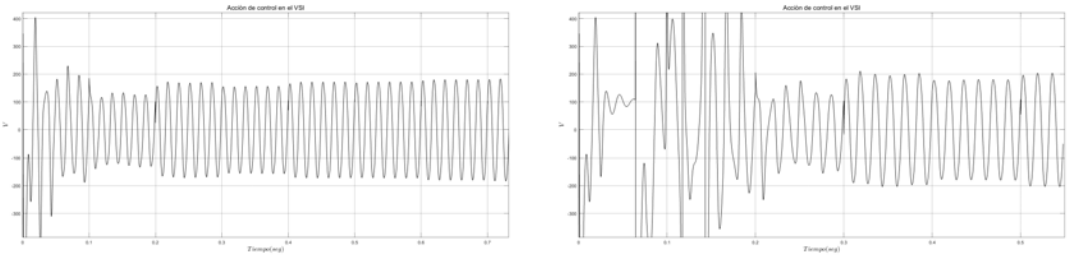
si la respectiva variable atacada no tiene cambios, el efecto de este ataque en el proceso de control será nulo.

Del mismo modo es interesante ver el efecto en las acciones de control cuando se llevan a cabo este tipo de ataques en la microgrid. Los diferentes efectos se observan en la Fig. 27.

Se puede observar que la influencia del ataque DoS genera una acción de control que requiere más energía que en el caso del ataque de integridad. Estos diversos efectos que se observaron tanto en las acciones de control como en las variables medidas del proceso, es información interesante que se usará para el diseño de los métodos de detección que se plantearon en el desarrollo de este trabajo, dado que se observó que el impacto, dependiendo del tipo de ataque es diferente.

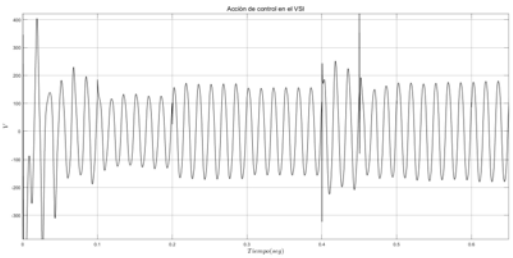
4.4 CONCLUSIONES

En este capítulo se abordó el primer objetivo de la propuesta de tesis. Mediante una revisión del estado actual de las contribuciones se analizaron los tipos de amenazas que se presentan actualmente en los sistemas ciberfísicos. Se logró evidenciar que los ataques más frecuentes y con mayor impacto en el ámbito de los sistemas de control soportados



(a) Acción de control en el VSI bajo operaciones normales.

(b) Acción de control en el VSI bajo ataque DoS.



(c) Acción de control en el VSI bajo ataque de integridad.

Fig. 27. Acción de control en el VSI.

en los sistemas ciberfísicos son los ataques de integridad y los ataques de denegación del servicio. Estos ataques pueden llegar a afectar las variables medidas del proceso y las acciones de control. El esquema mostrado en la Fig. 15 permite visualizar un modelo esquemático, en donde se observan las diferentes partes que se ven afectadas por los ataques que se consideraron en el desarrollo de esta propuesta.

Mediante un caso de estudio, microgrid, se logró observar mediante simulación los efectos de estos ataques en la funcionalidad de un sistema ciberfísico. El modelo simulado integró los modelos dinámicos de un sistema de control distribuido, los respectivos algoritmos de control y los aspectos computacionales y de comunicación que integran estos sistemas. Se observó que la consecuencia generada por los ataques difiere de la naturaleza del mismo. De este modo los efectos generados por un tipo de ataque afecta el comportamiento dinámico del sistema de una forma diferente. Las características observadas dan indicios de que a partir de la información proveniente de las medidas y de las acciones de control es posible desarrollar métodos que permitan detectar la ocurrencia de un ciberataque.

Los resultados expuestos han sido publicados en [157, 160, 161].

5. PROCEDIMIENTO DE DISEÑO DE SISTEMAS CIBERFÍSICOS PARA TOLERAR LOS ATAQUES DE INTEGRIDAD Y DoS

En este capítulo se presenta el planteamiento del procedimiento para diseñar sistemas ciberfísicos que permitan tolerar los ciberataques de integridad y DoS.

El capítulo se organiza de la siguiente manera. En la primera sección se realiza la propuesta del procedimiento de diseño. En la segunda sección se aborda la propuesta de estrategia para detectar ataques cibernéticos en sistemas de control soportados en CPSs. La tercera sección presenta la arquitectura propuesta para el desarrollo de CPSs y la verificación de los requisitos temporales. Finalmente, las conclusiones del capítulo son presentadas.

5.1 PROPUESTA DEL PROCEDIMIENTO DE DISEÑO

Considerando la diversidad de arquitecturas para el diseño de este tipo de sistemas, durante el planteamiento del procedimiento de diseño se tuvo en cuenta los siguientes aspectos:

- La aplicación debe descomponerse en componentes que permitan ofrecer o requerir micro servicios de otros componentes.

- Asociar cada microservicio descompuesto anteriormente a un contenedor y agruparlos estratégicamente según las necesidades que se requieran.

- Asociar en cada contenedor si el servicio es ofrecido o requerido con una publicación o suscripción.

- Disponer de un sistema de detección y aislamiento de ciberataques que permita ubicar qué componente del sistema está siendo afectado.

- Evaluar el sistema de detección mediante métricas que permitan establecer el rendimiento del mismo.

- Disponer de réplicas de contenedores y/o microservicio, de este modo si alguno se ve afectado puede ser reemplazado por otro que esté funcionando de manera correcta.

- Identificar y validar las limitaciones temporales de las aplicaciones.

El procedimiento de diseño propuesto se presenta en la Fig. 28.

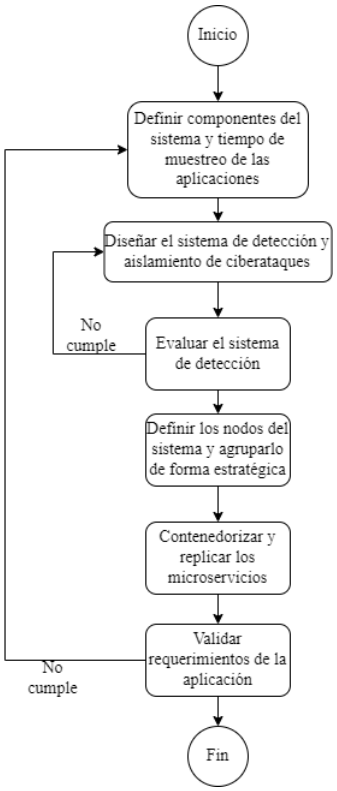


Fig. 28. Procedimiento de diseño de sistemas ciberfísicos para tolerar ciberataques.

Este planteamiento involucra los pasos mencionados anteriormente. En la primera etapa se tiene que establecer cuales son los componentes del sistema y que aplicaciones se requieren para su correcto funcionamiento. Al momento de definir estas aplicaciones se requiere que se definan las restricciones temporales que estas deben cumplir para tener el rendimiento adecuado. Se procede con una etapa, en donde se plantea el diseño de un sistema de detección y aislamiento de ciberataques, cuya finalidad es manifestar, a partir de una información de entrada, la ocurrencia de un ciberataque y qué componente es el que está siendo afectado. Este sistema requiere de un proceso de evaluación a partir de unas métricas que permitan definir el rendimiento del mismo. Si no cumple con un rendimiento adecuado, se debe replantear el diseño del sistema de detección hasta obtener un buen desempeño. El algoritmo de las etapas del procedimiento correspondientes al diseño del sistema de detección y evaluación de este se presentan en la Fig. 29.

En la etapa siguiente se definen los nodos que integraran al sistema. Estos nodos serán desarrollados en una arquitectura que dispondrá de microservicios soportados en la tecnología de contenedores los cuales a través de la filosofía de publicar/subscribir permitirán la comunicación entre los servicios. Este diseño adopta el desacoplamiento como

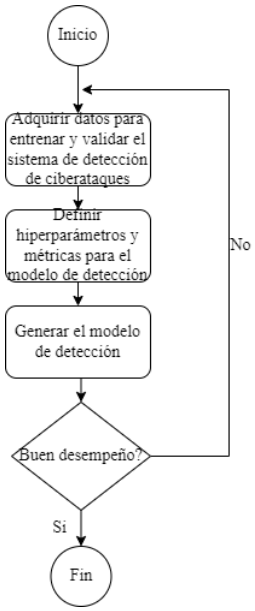


Fig. 29. Diagrama de flujo para diseñar el sistema de detección de ciberataques.

paradigma fundamental al interconectar entidades que interactúan entre sí. Esta función de desacoplamiento permite la interconexión de entidades en función de los datos que desean intercambiar en lugar de la interconexión directa punto a punto de esas entidades. Debido a esto, se logra la integración de una gran cantidad de tecnologías diferentes y se brindan buenas características de escalabilidad. Bajo el paradigma de publicar/subscribir, si el sistema de detección da como resultado que algún microservicio o componente del sistema esta siendo afectado por un ciberataque, se puede detener la ejecución del mismo y ejecutar una replica del microservicio que en su momento se encuentre funcionando correctamente. Esta es la idea básica para el diseño de sistemas ciberfísicos tolerantes a ciberataques.

En la última etapa se verifica los requerimientos de la aplicación. Si en el proceso de validación el sistema no cumple con las restricciones y condiciones impuestas, se debe devolver a las etapas previas y realizar un rediseño.

En las siguientes secciones se define la estrategia para la detección y aislamientos de ciberataques así como el diseño de la arquitectura de los nodos que integraran los elementos que componen al sistema ciberfísico.

5.2 ESTRATEGIA PARA DETECTAR CIBER ATAQUES EN SISTEMAS CIBERFÍSICOS

La detección de ataques cibernéticos en CPS se aborda típicamente mediante una comparación entre un sistema ideal y el sistema real. Lo primero a realizar es generar una señal residual $res(k)$ a través de la Ecuación (6). A partir de este resultado se usa la Ecuación (7) para evaluar el residual a través de umbrales. Finalmente, se realiza un proceso de toma de decisiones a través de indicadores. Este procedimiento se muestra en el algoritmo 1, donde N es el número de muestras, $\hat{y}_d(k)$ son salidas desacopladas, a_d es la señal de detección y a_i es la señal de aislamiento. Estos pasos fueron presentados previamente en la sección 3.2.1.

Algorithm 1: Algoritmo de detección y aislamiento de ciberataques

Input: Entradas y salidas del proceso

Output: Señal de detección y aislamiento

Inicialización : Condiciones iniciales del proceso

1: **for** $i = 1$ to N **do**
2: Medir $y(i)$
3: Estimar $\hat{y}(i)$ y $\hat{y}_d(i)$
4: Calcular $res(i) = |y(i) - \hat{y}(i)|$
5: Calcular $res_d(i) = |y(i) - \hat{y}_d(i)|$
6: **if** ($res(i) >= \tau_{det}$) **then**
7: $a_d(i) = 1$
8: **else**
9: $a_d(i) = 0$
10: **end if**
11: **if** ($res_d(i) >= \tau_{iso_{on}}$ & $res_d(i) <= \tau_{iso_{off}}$) **then**
12: $a_i(i) = 1$
13: **else**
14: $a_i(i) = 0$
15: **end if**
16: **end for**
17: **return** a_d,a_i

Estos pasos implican el uso de residuos que deben tomar valores cercanos a 0 en situaciones en las que el sistema no está siendo atacado. Por otro lado, cuando hay un ataque, las señales residuales deben tener valores distintos de 0.

Aunque una sola señal residual puede alertar o detectar un ciberataque, se requiere un

conjunto de residuales para aislarlo. Entonces, para localizar el origen del ciberataque, es necesario que algunos residuos sean sensibles sólo para una parte concreta del sistema. Lo anterior implica que el conjunto de residuos debe ser independiente de otros ciberataques definidos. De esta manera, para aislar un ciberataque, se considera un conjunto estructurado de residuos, donde cada vector residual puede ser utilizado para detectar un ciberataque en un lugar específico del sistema. En este caso, se utiliza el umbral ($\tau_{iso_{off}}$), definido para situaciones normales. En el caso opuesto, en el que el sistema está siendo atacado, las señales residuales deben tener valores desviados de 0. En este caso, se utilizan los umbrales $\tau_{iso_{on}}$ y τ_{det} para referirse al aislamiento y a la detección. Aunque una sola señal residual ($res(k)$) puede detectar la ocurrencia de un ciberataque, se requiere un conjunto de estas señales para aislar ($res_d(k)$), lo que permite evaluar de forma desacoplada las diferentes partes del sistema, para localizar el origen del ciberataque.

La arquitectura propuesta para la detección y aislamiento de ciberataques se presenta en la Fig. 30. Esta arquitectura incluye un modelo de predicción que utiliza un conjunto de datos de entrada x_0,$x_1, \ldots, x_{k-1}$ para estimar las salidas $\hat{y}_1$,$\hat{y}_2, \ldots, \hat{y}_k$ (este conjuntos de datos dependerán específicamente del tipo de datos disponibles del proceso), y estos valores se utilizan para obtener la señal residual $res(k)$, como se muestra en (6). Estas señales son utilizadas por un clasificador para detectar las anomalías presentes en el proceso.

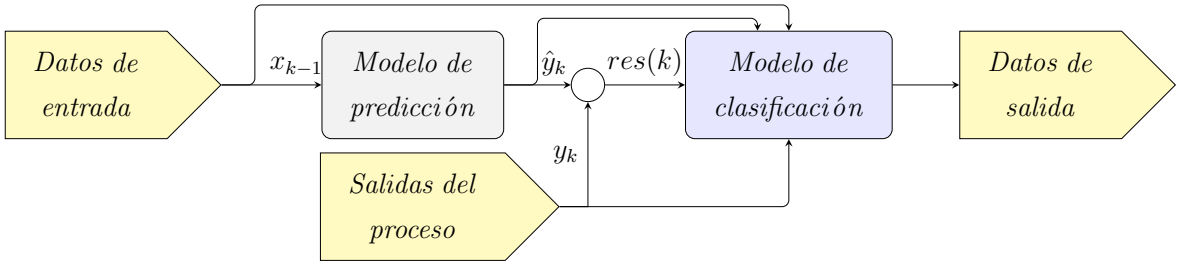


Fig. 30. Modelo de arquitectura general para detectar y aislar el ciberataque en un CPS.

A continuación se presentan casos en los que se desarrolla el sistema de detección a partir del esquema mostrado en la Fig. 30.

5.2.1 Detección y aislamiento Caso: Microgrid

Este primer caso permitió observar el desempeño de un sistema detección utilizando el enfoque basado en el diseño LO y UIOs, que permitieran aislar y detectar la ocurrencia de ciberataques dentro de la microgrid planteada en la sección 4.3. Los ataques a considerar fueron los analizados en el capítulo 4. En ese sentido se propone la siguiente

arquitectura lógica para la implementación del sistema de detección de ciberataques en la microgrid, Fig. 31.

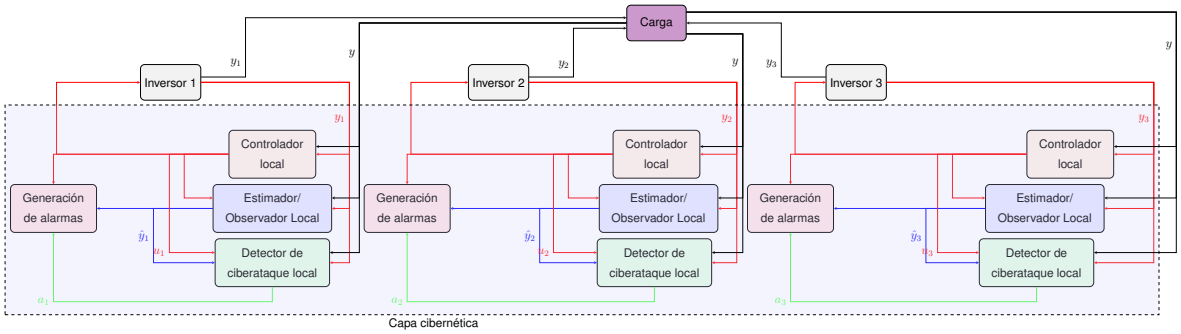


Fig. 31. Arquitectura propuesta para la detección de ataques implementada para la microgrid.

En este caso, se piensa generar ataques que pueden llegar a ocurrir en los sensores de de las salidas de corrientes de los inversores así como la tensión en la carga. Estas variables son las que se denominan y_1,y_2,y_3 y y en el esquema mostrado en la Fig. 31.

El proceso para la detección del ciberataque se muestra en la Fig. 32. Como se observa, cualquier situación anómala, se detecta en el proceso de comparación y evaluación entre la variable del proceso real y la variable estimada.

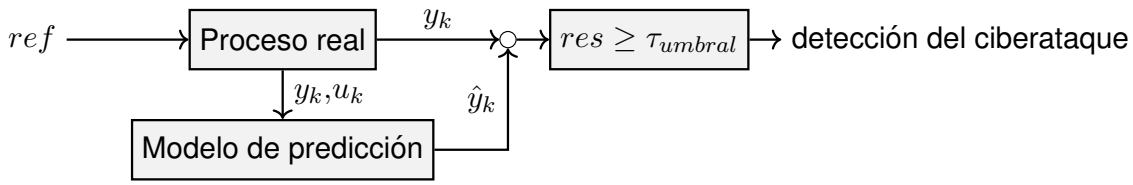


Fig. 32. Sistema de detección.

En este caso en particular es de interés observar el desempeño de metodologías que usan observadores de estado para generar las salidas estimadas. De este modo se obtiene el esquema mostrado en la Fig. 33 para el proceso de detección.

Para poder aislar las diversas situaciones que pueden llegar a presentarse en los diferentes puntos de medición vulnerables a ciberataques, se requiere tener variables desacopladas, lo cual permite tener un grado de sensibilidad para detectar que parte del sistema está siendo afectado, Fig. 34.

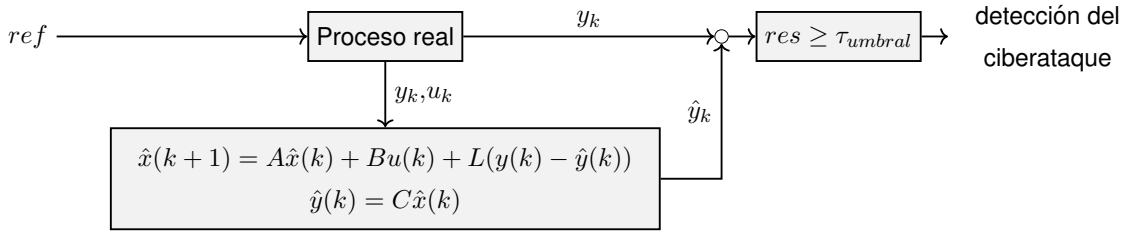


Fig. 33. Detección basada en el observador de Luenberguer.

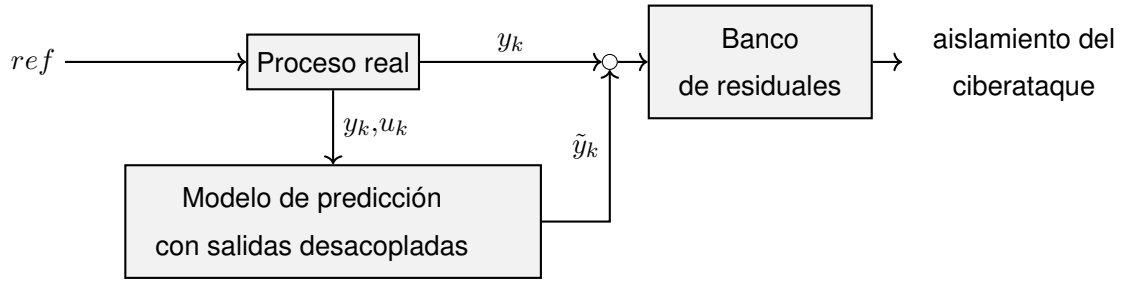


Fig. 34. Sistema de aislamiento.

De este modo se usan los UIOs que permiten tener salidas sensibles a unas situaciones e insensibles a otras. Esto se muestra en la Fig. 35.

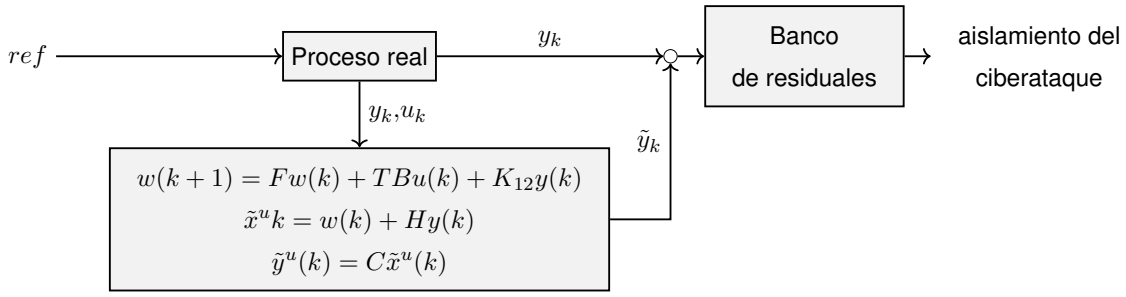


Fig. 35. Aislamiento basado en el observador de entradas desconocidas.

Así con este conjunto de esquemas, se puede llegar a indicar cuál es la variable afectada. Para observar el desempeño de estos esquemas se llevaron a cabo las situaciones mostradas en la Tabla VIII.

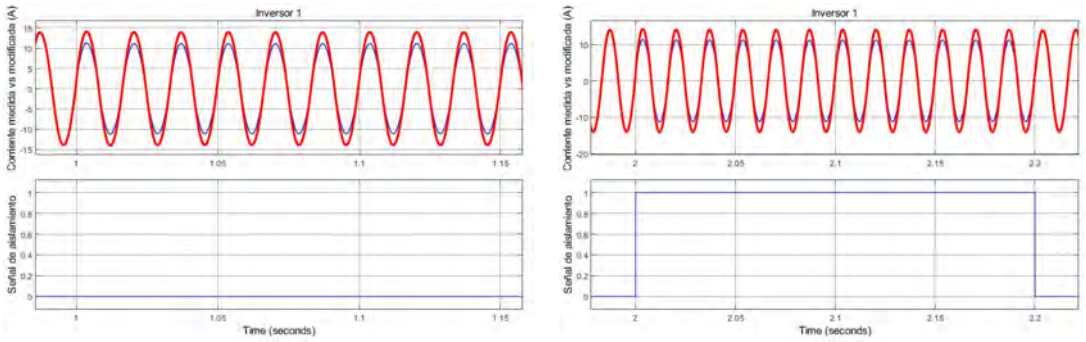
Se generaran cuatro ataques en dos sensores que miden la corriente en dos de los inversores que componen al sistema. Se tiene inicialmente un par de ataques simultáneos que ocurren en el intervalo de tiempo entre $[1\ 1{,}3]$ y posteriormente dos ataques indivi-

TABLA VIII.
Ataques generados en la microgrid.

Sensor afectado	Ataque	Tiempo de inicio(s)	Tiempo de fin(s)
Sensor de corriente del inversor 1	$y_1(k) = y_1(k) + 0{,}25y_1(k)$	1	$1{,}3$
		2	$2{,}2$
Sensor de corriente del inversor 2	$y_2(k) = y_2(k) + 0{,}25y_2(k)$	1	$1{,}3$
		$2{,}5$	$2{,}8$

duales llevados a cabo en los intervalos definidos en la Tabla VIII. Se estudia el fenómeno de los ataques simultáneos dado que rara vez son abordados en el diseño de los sistemas de detección de ciberataques, lo cual es preocupante porque estas situaciones pueden darse con mucha frecuencia en el mundo real.

Los resultados del sistema de detección, así como la respuesta temporal de la variable afectada, se observan en la Fig. 36 y Fig. 37. Se puede observar que la señal sinusoidal

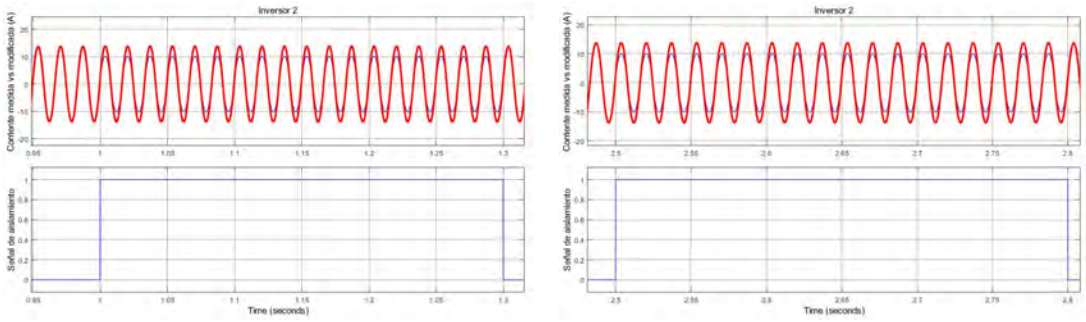


(a) Primer ataque sobre el inversor 1. (b) Segundo ataque sobre el inversor 1.

Fig. 36. Respuesta temporal y señal de alarma de la corriente del inversor 1.

azul, la cuál es la señal corrompida se atenúa, esto es debido a que el controlador trata de mantener el error en estado estacionario disminuyendo las corrientes generadas por estos inversores. El resultado del sistema de detección se puede observar en las señales de alarma, las cuales toman el valor de uno cuando se detecta una anomalía y toma el valor de cero cuando se presentan situaciones normales. Se puede observar que el algoritmo solo detectó el cambio en el segundo intervalo en el primer inversor, de $2s$ a $2{,}2s$, mientras que el primer ataque, que fue un ataque simultaneo, no pudo ser detectado.

Como se puede observar, el método utilizado presentó dificultades en la detección de los



(a) Primer ataque sobre el inversor 2. (b) Segundo ataque sobre el inversor 2.

Fig. 37. Respuesta temporal y señal de alarma de la corriente del inversor 2.

diferentes casos propuestos, presentando menor desempeño en presencia de ataques simultáneos. Dichos resultados se justifican en lo siguiente. Por un lado, está el depender de un modelo preciso del proceso, algo que en aplicaciones prácticas es muy difícil de alcanzar. Adicionalmente, el identificar un umbral apropiado es un arte, y depende en gran medida del conocimiento que se tenga del proceso.

Particularmente, en los casos analizados la principal dificultad se asocia con el umbral, dado que se tiene pleno conocimiento del modelo simulado. Aunque algunos autores han desarrollado observadores de estado para la detección y aislamiento de ciberataques [157, 162–164], estos métodos presentan inconvenientes porque la detección de anomalías se realiza mediante la comparación de un umbral fijo definido por un dato histórico de comportamiento normal, con la diferencia entre las variables del proceso real y los valores generados por un modelo estimado. Entonces, puede conducir a una tasa considerable de falsos positivos y falsos negativos. Lo anterior se debe a que, para el diseño de los bancos de observadores, se utiliza el conocimiento de los parámetros y la dinámica del sistema, que en ocasiones puede ser significativamente diferente del desempeño real del sistema. Entonces, ambas propuestas están limitadas por el conocimiento del proceso, como la definición del umbral, que, en situaciones reales, puede no ser fácil de modelar con precisión.

Como alternativa para abordar las situaciones descritas en los casos anteriores, se evaluó el uso de técnicas basadas en inteligencia artificial para realizar la detección y el aislamiento, debido a que en los últimos años, se han empleado para detectar ataques cibernéticos [27, 44–50]. Estos métodos han presentado un buen desempeño para encontrar modelos de procesos que incluso presentan dinámicas no lineales bastante pronunciadas. La tecnología de aprendizaje automático es uno de los métodos basados en datos que emerge como método para detectar ataques en estos sistemas [48, 49, 165–174].

Este planteamiento puede tener un mayor acercamiento a la implementación en casos reales en los que se tiene un elevado grado de incertidumbre en los modelos de los procesos. En este sentido, para la estimación de los estados del sistema se utilizan algoritmos que han presentado buen comportamiento en el tratamiento de problemas de regresión; obteniendo de esta manera los modelos con el cual se comparan las variables del proceso, para posteriormente realizar la detección y aislamiento de los ataques utilizando redes de aprendizaje profundo (Deep Learning) apropiadas para el tratamiento de problemas de clasificación.

Para validar la propuesta se utilizaron dos bancos de pruebas. Para la selección de estos conjuntos de datos se realizó una búsqueda que incluyó palabras clave, como seguridad en sistemas de control industrial, detección de fallas, anomalías, ciberataques en sistemas de control y diseño de CPSs seguros. A partir de esta búsqueda, se consideraron las publicaciones que tuvieron un tiempo de publicación de menos de 5 años, así como el número de veces que los conjuntos de datos se usaron para evaluar la seguridad de los CPS. También se consideró el tipo de ataques que se implementaron, ya que el enfoque es abordar diferentes tipos de ataques, incluidos los de mayor frecuencia e impacto en los sistemas de control que se encuentran en los CPSs (ataques de integridad y DoS). El primer banco de pruebas corresponde al SWaT(Secure Water Treatment, por sus siglas en inglés), que son un conjunto de datos, que proporciona datos reales de una versión simplificada de una planta de tratamiento de agua del mundo real. Este conjunto de datos permite a los investigadores diseñar y evaluar los mecanismos de defensa de los CPS y contiene tanto el tráfico de la red como los datos relacionados con las propiedades físicas del sistema [175]. Por otro lado, existe otro banco de pruebas que consta de tres tanques interconectados [42] que ha permitido validar diferentes tipos de métodos de detección de ciberataques a CPS. Estos dos banco de pruebas han permitido validar diferentes propuestas centradas en técnicas que nos permiten, de una forma u otra, analizar la detección de ciberataques [162, 165, 176–186] y han permitido orientar esta investigación para mejorar la propuestas realizadas.

A partir del esquema mostrado en la Fig. 30 y debido a que las tareas de predicción y clasificación serán abordadas a partir de inteligencia artificial es muy probable que haya que realizar una etapa de preprocesamiento, debido a que frecuentemente las características de las señales en un proceso específico son diferentes, los valores con diferente magnitud podrían afectar el procedimiento de entrenamiento del regresor y del clasificador, por lo tanto, todos los datos de entrada al clasificador se normalizan usando su media y desviación estándar para obtener el z-score para cada uno como se muestra en (42).

$$z = |x - \mu|/\sigma \tag{42}$$

Donde x son los datos de entrada, μ es la media y σ es la desviación estándar.

Aunque la arquitectura presentada es general, es una base para seleccionar diferentes tipos de técnicas de aprendizaje automático para las etapas de predicción y clasificación. La idea es utilizar redes neuronales profundas para extraer patrones que permitan la detección de ciberataques (como LSTM o CNN 1-dimensional). Como no se incluyó un método para encontrar correlaciones espacio-temporales para detectar ciberataques, se espera que las redes neuronales puedan realizar esta tarea implícitamente.

La arquitectura basada en redes neuronales para la detección de ciberataques en un CPS, se puede detallar en la Fig.38. Un modelo que representa la dinámica del proceso genera las señales de salida $x(k)_s$ que corresponden a la reconstrucción de todos los estados (se asume que las salidas son los estados del proceso o alguna combinación lineal de ellos, aunque puede ser extendido a casos no lineales). Para aislar el ataque, existe un conjunto de modelos de redes neuronales que relacionan los estados del proceso con sus respectivas acciones de control para generar estados que se desacoplan entre sí ($x(k)_{d_{1,2,...x}}$). De esta forma, es posible aislar el ataque de forma equivalente al uso de UIOs, pero con la ventaja de que las redes neuronales permiten abordar la incertidumbre en las representaciones. Con este conjunto de redes neuronales, se genera la señal residual $res(k)$.

Las funciones de detección y aislamiento se implementan mediante redes neuronales artificiales, que utilizan los estados de proceso $x(k)$, las acciones de control $u(k)$, las señales de referencia $r(k)$, las señales residuales $res(k)$, y las señales generadas por el modelo de predicción, para poder alertas y localizar el ciberataque que se esta llevando a cabo.

El error cuadrático medio (MSE) [187] se adopta como función de pérdida del modelo para entrenar el modelo de predicción.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2, \tag{43}$$

Donde n es la cantidad de datos, x_i es el estado real y $\hat{x}_i$ es el estado estimado. Para el clasificador, se usa la función de costo entropía cruzada categórica (CCE) [188] debido a que el problema a resolver es un problema de clasificación de clases múltiples de una sola etiqueta.

$$J_{CCE} = -\sum_{q=1}^{l}\sum_{k=1}^{p} d_{qk}\log(y_{qk}). \tag{44}$$

Con p clases, tamaño de datos de entrenamiento de l, la entrada de x_q, donde $q =$

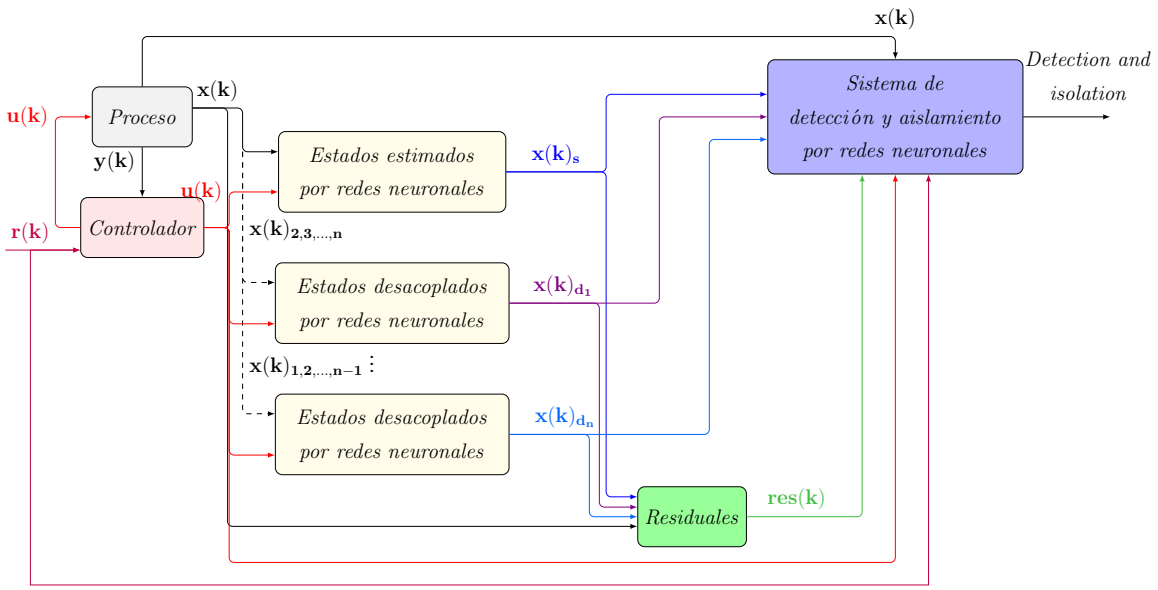


Fig. 38. Arquitectura basada en redes neuronales para detectar y aislar ciberataques.

$1,2,\ldots,l$ y y_{qk} $(0 \leq y_{qk} \leq 1)$, $k = 1,2,\ldots,p$ es la probabilidad estimada que pertenece a la clase k, y d_{qk} $(0$ o $1)$ se convierte en la etiqueta dada (44).

5.2.2 Sistema de detección y aislamiento de ciberataques basado en redes neuronales, Caso: Secure Water Treatment Dataset-SWaT

Este conjunto de datos fue completado por la Universidad de Tecnología y Diseño de Singapur para proporcionar a los investigadores datos recopilados de un entorno ICS complejo y realista. El banco de pruebas es una planta de tratamiento de agua a escala totalmente operativa que produce agua purificada. SWaT se compone de seis procesos principales correspondientes a los componentes físicos y de control de la planta de tratamiento de agua; cada etapa (de P1 a P6) está equipada con un cierto número de sensores y actuadores. Los sensores incluyen medidores de flujo, medidores de nivel de agua, conductividad y analizadores de pH, entre otros, mientras que los actuadores consisten en bombas que transfieren agua de una etapa a otra, bombas que dosifican químicos y válvulas que controlan el flujo de entrada. El proceso no es circular y se elimina el agua P6. Los sensores y actuadores de cada etapa están conectados al PLC correspondiente [189, 190] . Este proceso se muestra en la Fig. 39 [175].

La etapa P1 controla el flujo de agua cruda abriendo o cerrando una válvula motorizada que está conectada a la entrada del tanque T101. Por medio de la bomba P101, el agua comienza a fluir desde T101 a través de la estación de dosificación de químicos en la

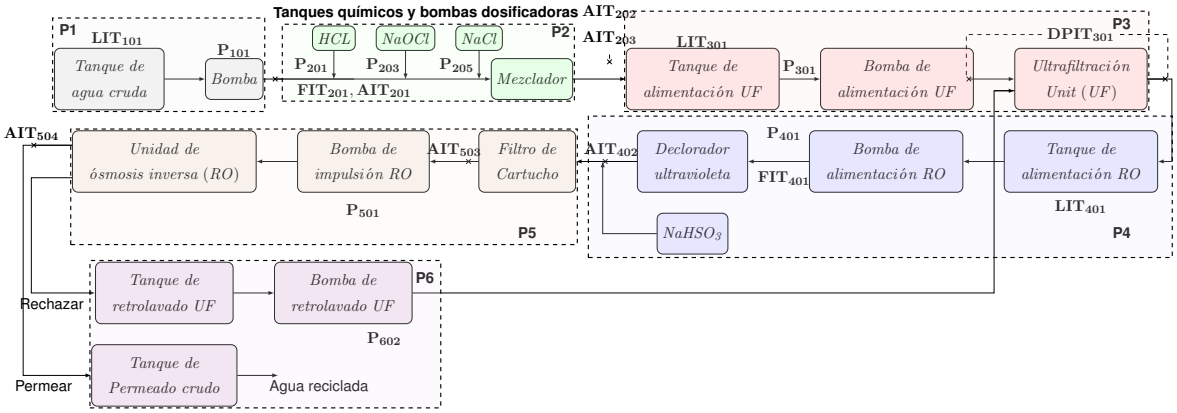


Fig. 39. Descripción general de los procesos del banco de pruebas SWaT.

etapa P2 y es seguida por el proceso de ultrafiltración (UF) ubicado en la etapa P3, que elimina los materiales no deseados. Asimismo, la bomba de alimentación de la etapa P3 es la encargada de suministrar el agua a la unidad de ultrafiltración. En la etapa P5, las impurezas inorgánicas se separan mediante un proceso de ósmosis inversa. La salida del proceso de ósmosis inversa se almacena en el tanque de permeado de la etapa P6 para su distribución. La etapa P6 también controla la limpieza de las membranas de ultrafiltración en P3 mediante el proceso de retrolavado. Cada cierto período de tiempo, el proceso de retrolavado se activa al encender la bomba de retrolavado y se realiza en menos de un minuto. El proceso de retrolavado puede ser iniciado alternativamente por un PLC cuando el valor del sensor de presión diferencial aumenta por encima de $0{,}4$, lo que significa que las membranas de UF están obstruidas [175, 190].

Para este caso se utilizaron el conjunto de datos de entrenamiento 1 y el conjunto de datos de entrenamiento 2. El primero corresponde a los datos recopilados en condiciones normales de funcionamiento. Este conjunto de datos se publicó el 20 de noviembre de 2016 y se generó a partir de una simulación de un año. El segundo conjunto de datos corresponde a situaciones en las que se generan escenarios de ataque. Este conjunto de datos con datos parcialmente etiquetados se publicó el 28 de noviembre de 2016. El conjunto de datos tiene una duración aproximada de seis meses y contiene varios ataques [175, 190].

Los datos del primer conjunto de datos se utilizan para generar un modelo correspondiente al bloque ”Modelo de predicción”que se muestra en la Fig. 30. La arquitectura propuesta en este caso se basa en un modelo CNN 1D, como se muestra en la Fig. 40.

Los datos de entrada se componen de 43 características compuestas principalmente por mediciones de sensores, estados de las bombas y posiciones de las válvulas. La

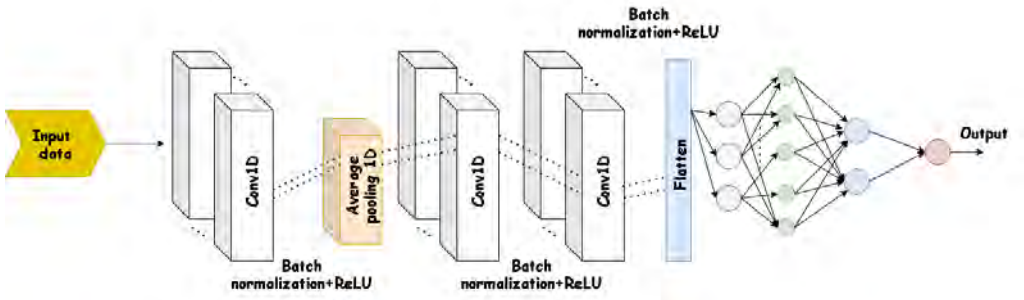


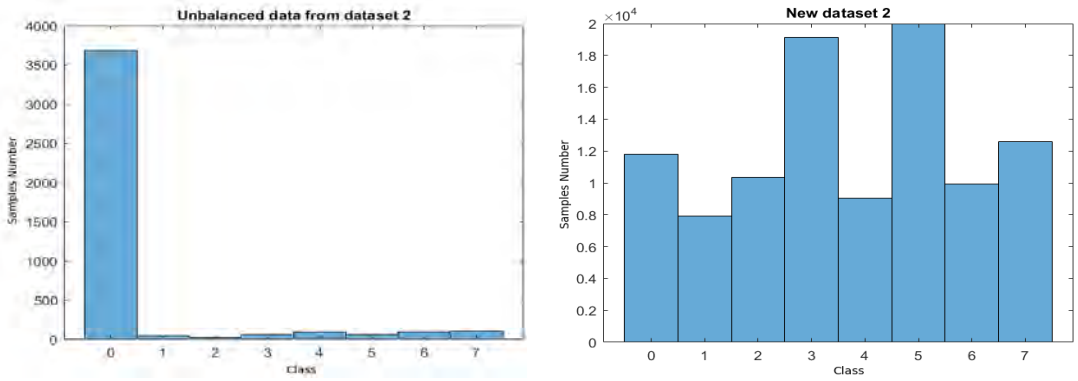
Fig. 40. Modelo de predicción para el dataset SWaT.

primera capa de convolución consta de 2 filtros y el tamaño del núcleo es 3. La capa de agrupación promedio 1D tiene un paso de 2 y el mismo relleno; la segunda capa de convolución tiene 20 filtros y un tamaño del núcleo de 20; la última capa de convolución está compuesta por 10 filtros y un tamaño de núcleo de 5. Finalmente, se utiliza una capa completamente conectada con una capa de 43 neuronas y una neurona en la capa de salida, todas con funciones de activación lineal. Además, la capa de normalización por lotes se agrega con la activación de ReLU en varias partes de la red. La función de pérdida utilizada fue MSE y el optimizador fue el gradiente descendiente estocástico con impulso. Para el entrenamiento, se dispuso de un máximo de 40 épocas con una tasa de aprendizaje inicial de 0,001. En este caso, el 30 % de los datos se utilizó para validar y el 70 % de los datos para entrenar.

Los parámetros de las capas para esta red se encontraron de tal manera que se logre el MSE más bajo posible. El aumento del número de capas, neuronas, tamaño de filtro o número de filtros no corresponde con una mejora significativa del rendimiento.

El segundo conjunto de datos se utilizó para el proceso de clasificación; está compuesto por 4177 datos, de los cuales 3685 corresponden a condiciones normales de operación, 50 pertenecen al primer escenario de ataque, 24 corresponden al segundo escenario de ataque, 60 al tercer y quinto ataque, 94 al cuarto y sexto ataque, y 110 al el séptimo escenario. Como se puede ver en la Fig. 41(a), este conjunto de datos está desequilibrado y luego generaría problemas al clasificador. La barra centrada en 0 corresponde a las condiciones normales de funcionamiento, mientras que las restantes corresponden a los diferentes escenarios de ataque del dataset. Podría afectar a los algoritmos en relación con las clases minoritarias. Para abordar esta situación, inicialmente, se utilizaron métodos, como el sobremuestreo y el submuestreo aleatorio, para una clasificación desbalanceada sin obtener resultados satisfactorios. Por esta razón, se siguió el enfoque que se muestra en [191]. Esta propuesta es una modificación de datos temporales determinados por secuencias óptimas que se alinean con los datos originales, generando así

nuevos datos sintetizados en el tiempo para el conjunto de datos de entrenamiento. La distribución de las diferentes clases para el nuevo conjunto de datos que se utilizará se muestra en la Fig. 41(b). Si bien se observa que es un conjunto de datos desequilibrado, se incrementó la cantidad de datos generados a partir de los escenarios de ataque y se mejoró el rendimiento.



(a) Distribución Original del dataset SWaT. (b) Nueva distribución del dataset SWaT.

Fig. 41. Distribución del dataset SWAT.

Para estimar las salidas se utilizó el nuevo conjunto de datos, los cuales se compararon con las variables de proceso habituales para obtener la señal residual. Las salidas estimadas, las variables de proceso y los residuos correspondientes, constituyen los datos de entrada para el clasificador cuya arquitectura se muestra en la Fig. 40.

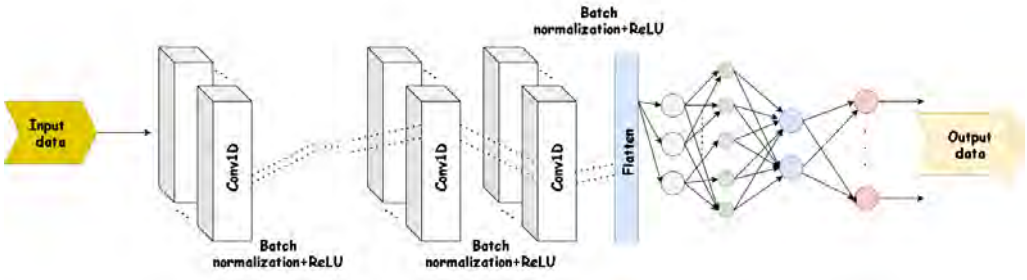


Fig. 42. Modelo de clasificación para el dataset SWaT.

El clasificador fue implementado por un grupo de capas convolucionales en cascada con una capa de normalización por lotes con la función de activación ReLU entre ellas. El número de capas convolucionales seleccionadas fue de cinco obteniendo una precisión superior al 90 %. El número de filtros implementados desde la entrada hasta la capa completamente conectada fue 128, 64, 32, 16 y 8 respectivamente. El tamaño de

kernel en cada uno fue de 10. La capa totalmente conectada está compuesta por ocho neuronas en su capa de entrada con función de activación lineal, mientras que la última capa tiene ocho neuronas con funciones de activación softmax correspondientes a los 7 ataques y los escenarios de operación habituales.

La función de pérdida utilizada fue CCE y el optimizador utilizado fue descenso de gradiente estocástico con impulso. Para el entrenamiento se dispuso de un máximo de 4 épocas, con una tasa de aprendizaje inicial de 0,0001, el 30 % del conjunto de datos se utilizó para validar y el 70 % para entrenar.

El proceso de validación del diseño propuesto para el sistema de detección y aislamiento de ciberataques que puedan presentarse en sistemas CPSs se presenta a continuación. Para ello se requieren definir las métricas que se usaron.

5.2.2.1 Métricas de evaluación

Las métricas consideradas en este trabajo fueron los verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN). Para evaluar el desempeño de la arquitectura propuesta, se utilizaron las siguientes métricas: precisión, exactitud, recall (sensibilidad o TPR), puntaje F1 y tasa de verdaderos negativos (TNR) o especificidad . Estas métricas se calcularon de la siguiente manera:

$$Precision = \frac{TP}{TP + FP} \tag{45}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{46}$$

$$Recall = \frac{TP}{TP + FN} \tag{47}$$

$$F1\ Score = \frac{2TP}{2TP + FP + FN} = 2\frac{Precision \times Recall}{Precision + Recall} \tag{48}$$

$$TNR = \frac{TN}{FP + TN}. \tag{49}$$

Además, se consideraron las curvas de característica operativa del receptor (Receiver Operating Characteristics, ROC por sus siglas en inglés) y Precision-Recall. La curva ROC permite observar la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos(FPR) a medida que cambia el umbral de decisión en un proceso de clasificación. Mientras que la curva Precision-Recall traza la relación entre la precisión, la cual se define como la capacidad que tiene un modelo de evitar la incorrecta clasificación de muestras negativas, y la capacidad del modelo de detectar todas las muestras positivas (Recall).

5.2.2.2 Resultados y discusión

Los resultados obtenidos para este conjunto de datos se muestran a continuación. El entrenamiento y los resultados se llevan a cabo en el software MATLAB. La Fig. 43 muestra la matriz de confusión para cada una de las clases disponibles. A partir de estos resultados, se obtienen las métricas definidas y se presentan en la Tabla IX.

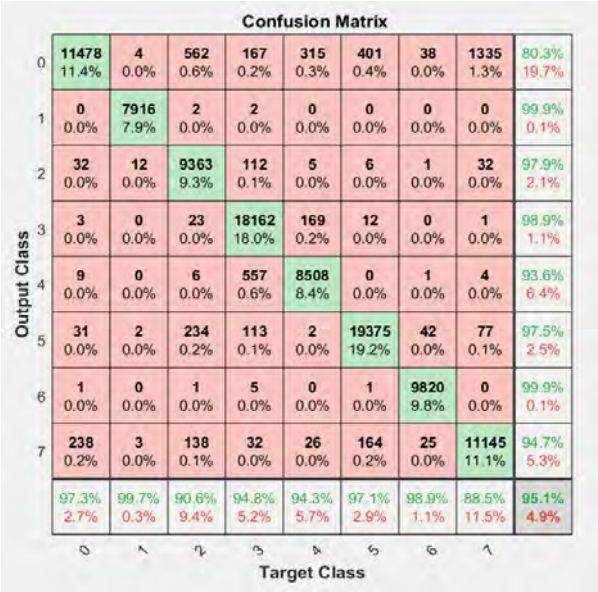


Fig. 43. Matriz de confusión para el dataset SWaT.

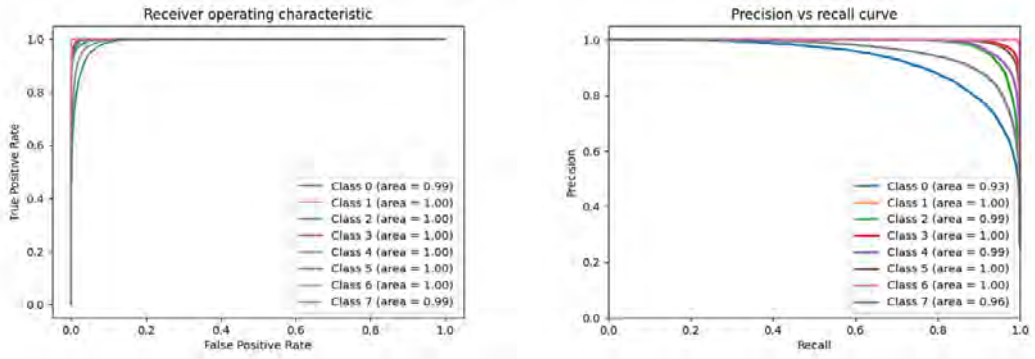
TABLA IX.
Resumen de métricas para el dataset SWaT.

	Exactitud	Precisión	Recall	F1 Score	TNR
Clase 0	0.97	0.81	0.97	0.88	0.98
Clase 1	0.99	0.99	0.99	0.99	0.99
Clase 2	0.99	0.98	0.91	0.94	0.99
Clase 3	0.99	0.99	0.95	0.97	0.98
Clase 4	0.99	0.94	0.94	0.94	0.99
Clase 5	0.99	0.98	0.97	0.97	0.98
Clase 6	0.99	0.99	0.99	0.99	0.99
Clase 7	0.98	0.95	0.89	0.92	0.98

La clase 0 corresponde a la operación habitual, mientras que las clases de 1 a 7 son los diferentes escenarios de ataques presentados en [175,190]. Se observa que la exactitud

es alta en todos los casos. Lo anterior muestra un alto porcentaje de muestras correctamente clasificadas por el modelo propuesto. Teniendo en cuenta la precisión, se observa que todos los escenarios de ataque presentan una puntuación superior a 0.94, lo que significa que muchos datos fueron correctamente clasificados en los diferentes escenarios de ataque. Del mismo modo, las puntuaciones de recall están por encima de 0,91 en la mayoría de clases, lo que permite minimizar la tasa de falsas alarmas. Finalmente, la puntuación F1 muestra puntuaciones superiores a 0,92. Se destaca la alta tasa de TNR en cada una de las clases, lo que significa que la FPR es baja.

Las curvas ROC y Precision-Recall que se muestran en las Fig. 44(a) y 44(b). Ambas presentan un desempeño apropiado, lo que indica que el modelo tiene una buena capacidad para distinguir las diferentes clases. El área bajo la curva (AUC) muestra que la proporción de muestras clasificadas correctamente es alta en cada una de las clases.



(a) Curva ROC (b) Curva Precisión-recall

Fig. 44. Curvas ROC y de Precisión-Recall para el dataset SWaT.

La Tabla X presenta el resumen y una comparación de la propuesta presentada en este documento con otros métodos que han utilizado esta base de datos. En las métricas de Recall y F1 Score, la propuesta realizada presenta un mejor desempeño en relación con los otros métodos. En cuanto a los valores de precisión y exactitud, la propuesta realizada está por encima en casi todos los casos, excepto en los dos últimos métodos que la superan por un margen de puntuación de 0,04. Sin embargo, el rendimiento de la métrica F1 Score es alto, lo que indica que se obtuvo una detección de clase satisfactoria y confiable.

TABLA X.
Comparación de rendimiento en el dataset SWaT.

Método	Exactitud	Precisión	Recall	F1 Score
Propuesto	0.95	0.95	0.95	0.95
SVM [176]	-	0.93	0.70	0.79
RNN [176]	-	0.94	0.70	0.80
1D CNN [177]	-	0.96	0.80	0.87
TABOR [178]	0.95	0.86	0.79	0.82
STAE-AD [180]	-	0.96	0.82	0.88
AE [181]	-	0.89	0.80	0.84
AE Frequency [181]	-	0.92	0.83	0.87
LSTM [179]	-	0.98	0.68	0.88
One Class SVM [179]	-	0.93	0.70	0.80
SDA+1D CNN+ LSTM [165]	0.99	0.99	0.85	0.91
SDA+1D CNN+ GRU [165]	0.99	0.99	0.85	0.92

5.2.3 Detección y aislamiento Caso: Banco de pruebas de tanques interconectados

Este banco de pruebas se ha utilizado ampliamente para probar propuestas para detectar anomalías [162, 182–186]. El sistema hidráulico consta de tres tanques cilíndricos idénticos con el mismo área de sección transversal S, como se muestra en la Fig. 45. Estos tanques están conectados por dos tubos cilíndricos de la misma área de sección transversal, denominados S_n, y tienen el mismo coeficiente de salida, denominado μ_{13} y μ_{32}. El flujo de salida nominal ubicado en el tanque 2 tiene la misma área de sección transversal que la tubería de acoplamiento entre los cilindros, pero un coeficiente de flujo de salida diferente, denotado μ_{20}. El flujo de entrada de los tanques proviene de dos bombas, con un caudal, q_1 y q_2. Se utiliza un convertidor digital / analógico para controlar cada bomba. Un sensor de presión diferencial piezo resistivo realiza la medición de nivel necesaria. La idea del sistema es mantener los niveles de altura del fluido almacenado en los tanques 1 y 2 en un punto de operación particular.

Los parámetros se muestran en la Tabla XI, y el modelo matemático se presenta en las

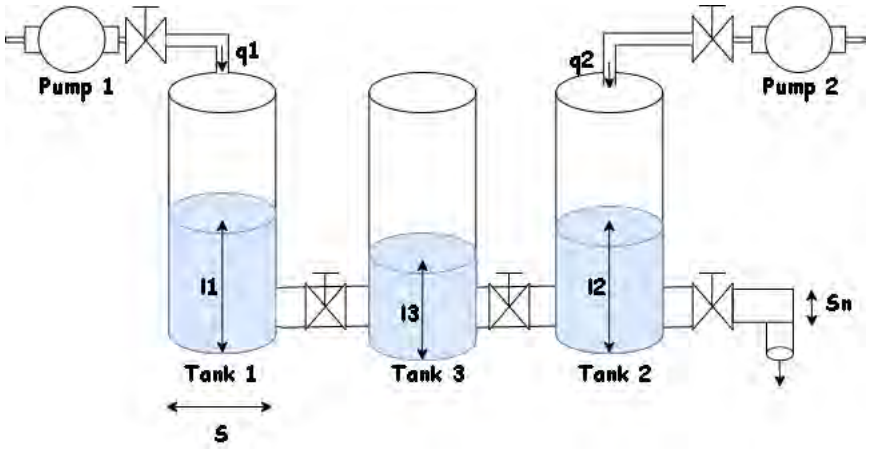


Fig. 45. Diagrama esquemático del sistema de tres tanques.

Ecuaciones (50), (51) y (52) [42].

$$\begin{aligned} \frac{dl_1(t)}{dt} &= (q_1(t) - q_{13}(t))/S \\ \frac{dl_2(t)}{dt} &= (q_2(t) + q_{32}(t) - q_{20}(t))/S, \\ \frac{dl_3(t)}{dt} &= (q_{13}(t) - q_{32}(t))/S \end{aligned} \tag{50}$$

$$q_{mn}(t) = \mu_{mn} S_p sign(l_m(t) - l_n(t)) \sqrt{2g|l_m(t) - l_n(t)|} \; (m, n = 1, 2, 3 \; \forall \; m \neq n), \tag{51}$$

$$q_{20}(t) = \mu_{20} S_p \sqrt{2g l_2(t)}. \tag{52}$$

TABLA XI.
Parámetros del sistema de tres tanques.

Variable	Símbolo	Valor
Área de la sección transversal del tanque	S	$0{,}0154$ m^2
Área de la sección transversal entre tanques	S_n	5×10^{-5} m^2
Coeficiente de salida	$\mu_{13} = \mu_{32}$	$0{,}05$
Coeficiente de salida	μ_{20}	$0{,}675$
Caudal máximo	$q_{imax}(i \in [1 \; 2])$	$1{,}2 \times 10^{-4}$ m^3/s
Nivel máximo	$l_{jmax}(j \in [1 \; 2 \; 3])$	$0{,}62$

Suponiendo que $l_1 > l_2 > l_3$, se estableció una aproximación lineal alrededor de un punto de equilibrio (U_0, Y_0) usando la expansión de la serie de Taylor. El sistema linealizado

se describe mediante una representación de espacio de estado discreto con un período de muestreo de $T_s = 1s$. Esto se muestra en la Ecuación (53).

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned}. \tag{53}$$

Los estados $x(k)$ corresponden al nivel de fluido de los tanques. El propósito de este estudio es controlar el sistema alrededor del punto operativo (U_0, Y_0), como se muestra en Ecuación (54).

$$\begin{aligned} Y_0 &= [0{,}4\ 0{,}2\ 0{,}3]^T(m) \\ U_0 &= [0{,}35 \times 10^{-4}\ 0{,}375 \times 10^{-4}]^T(m^3/s) \end{aligned}. \tag{54}$$

En este caso de estudio se consideró que las salidas $y = [l_1\ l_2]^T$ sigan unas referencias deseadas, estas mediciones se considerarán como objetivo del ataque. Se utilizó la técnica de asignación de polos por retroalimentación de estado. Por lo tanto, se diseñó una matriz de ganancia de retroalimentación K, de modo que los valores propios de bucle cerrado del sistema aumentado sean iguales a $[0{,}92\ 0{,}97\ 0{,}9\ 0{,}95\ 0{,}94]$. Se utilizó el software MATLAB para encontrar las matrices A y B, así como las ganancias del controlador. Los valores se pueden observar en la Ecuaciones (55), (56) y (57).

$$A = \begin{bmatrix} 0{,}9888 & 0{,}0001 & 0{,}0112 \\ 0{,}0001 & 0{,}9781 & 0{,}0111 \\ 0{,}0112 & 0{,}0111 & 0{,}9776 \end{bmatrix} \tag{55}$$

$$B = \begin{bmatrix} 64{,}5687 & 0{,}0014 \\ 0{,}0014 & 64{,}2202 \\ 0{,}3650 & 0{,}3637 \end{bmatrix} \tag{56}$$

$$K = [K_1\ |K_2] = 10^{-4} \left[\begin{pmatrix} 21{,}6 & 3 & -5 \\ 2{,}9 & 19 & -4 \end{pmatrix} \ | \begin{pmatrix} -0{,}95 & -0{,}32 \\ -0{,}3 & -0{,}91 \end{pmatrix} \right] \tag{57}$$

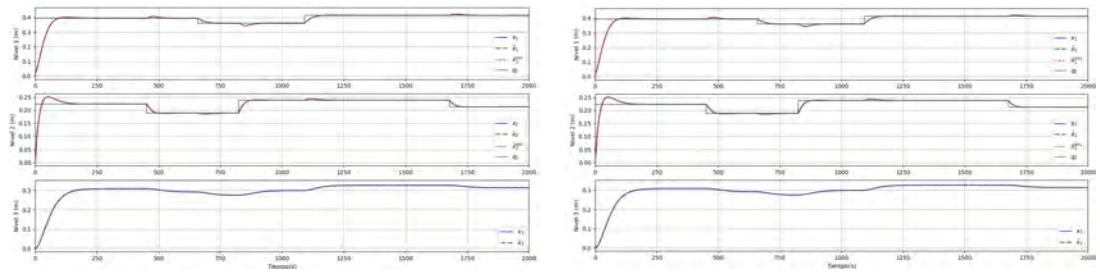
Para este caso se deseó realizar una comparación entre los métodos tradicionales que usan los observadores y la definición de umbrales con la arquitectura planteada por redes neuronales. Se plantearon arquitecturas basadas en LSTM y CNN, las cuales han mostrados buenos resultados en la implementación de sistemas detectores de intrusos en redes de comunicación. A continuación se describe el proceso de diseño de cada una de las etapas que se requieren para el sistema de detección con cada uno de los métodos.

5.2.3.1 Diseño de los observadores de estado

Teniendo en cuenta la discretización del modelo descrito en la Ecuación (50) se diseñan los observadores que se requieren para la detección de ciberataques. Haciendo uso de las expresiones (8), (9), (12), (13) y (14) y las matrices descritas en (55), (56) y (57), se obtienen las respectivas matrices de todos los observadores, Ecuación (58).

$$
\begin{aligned}
A &= \begin{bmatrix} 0{,}9888 & 0{,}0001 & 0{,}0112 \\ 0{,}0001 & 0{,}9781 & 0{,}0111 \\ 0{,}0112 & 0{,}0111 & 0{,}9776 \end{bmatrix} \qquad B = \begin{bmatrix} 64{,}5687 & 0{,}0014 \\ 0{,}0014 & 64{,}2202 \\ 0{,}3650 & 0{,}3637 \end{bmatrix} \\
L &= \begin{bmatrix} 0{,}9899 & 0{,}0005 \\ 0{,}0004 & 0{,}9894 \\ 0{,}010 & 0{,}0107 \end{bmatrix} \\
F_1 &= \begin{bmatrix} 0{,}9888 & -0{,}0010 & 0{,}0112 \\ 0 & 0{,}0010 & 0 \\ 0{,}0112 & 0{,}0120 & 0{,}9776 \end{bmatrix} \quad T_1 = \begin{bmatrix} 1{,}0000 & -0{,}0001 & 0 \\ 0 & 0 & 0 \\ 0 & -0{,}0001 & 1{,}0000 \end{bmatrix} \\
K_{1U} &= \begin{bmatrix} 0{,}0001 \\ 0 \\ 0{,}0111 \end{bmatrix} \qquad H_1 = \begin{bmatrix} 0{,}0001 \\ 1 \\ 0{,}0001 \end{bmatrix} \\
F_2 &= \begin{bmatrix} 0{,}0010 & 0 & 0 \\ 0{,}0010 & 0{,}9781 & 0{,}0111 \\ 0{,}0121 & 0{,}0111 & 0{,}9776 \end{bmatrix} \quad T_2 = \begin{bmatrix} 0 & 0 & 0 \\ -0{,}0001 & 1{,}0000 & 0 \\ -0{,}0001 & 0 & 1{,}0000 \end{bmatrix} \\
K_{2U} &= \begin{bmatrix} 0 \\ 0{,}0001 \\ 0{,}0112 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 1 \\ 0{,}0001 \\ 0{,}0001 \end{bmatrix}
\end{aligned}
\tag{58}
$$

Teniendo esto en cuenta, estos serán los modelos que se usaron para la detección y aislamiento de ciberataques, siguiendo los esquemas que se han mostrado anteriormente. La Fig. 46(a) y la Fig. 46(b) muestran resultados de este sistema, donde se observan tanto las referencias, q_1, q_2, los estados reales, x_1,x_2,x_3, y los estados estimados por el conjunto de observadores diseñados, $\hat{x}_1$,$\hat{x}_2$,$\hat{x}_3$ para el LO y $\tilde{x}_1^{uio_1}$,$\tilde{x}_2^{uio_1}$,$\tilde{x}_1^{uio_2}$,$\tilde{x}_2^{uio_2}$ para los estados estimados por el banco de UIOs.



(a) Usando el UIO_1. (b) Usando el UIO_2.

Fig. 46. Respuesta temporal bajo condiciones normales.

5.2.3.2 Diseño de las redes LSTM y CNN-1D para el sistema de detección de ciberataques

Para construir el conjunto de datos para la detección de ataques se implementó el esquema mostrado en la Fig. 47, el cual cuenta con módulos para obtener medidas de las variables del proceso, así como las acciones de control aplicadas por los actuadores. Se utilizó una Ethernet como red de control. Esta representación es equivalente a los bloques de $Proceso$ y $Controlador$ en la arquitectura presentada en la Fig. 38.

Se generaron dos conjuntos de datos. El primero es un conjunto de datos en operaciones normales para determinar un modelo que estima las salidas del sistema. El segundo incluye ciberataques a los sensores 1 y 2. Estos ciberataques pueden ser ataques de integridad o DoS.

En ambos casos se generaron $499{,}000$ muestras. Las referencias del sistema oscilan entre $0{,}35m$ y $0{,}45m$ para l_1, y entre $0{,}185m$ y $0{,}25m$ para l_2. Los intervalos de tiempo se definieron aleatoriamente con una distribución uniforme y cambios de referencia cada $500s$ a $850s$.

Los casos se muestran en la Tabla XII. El caso 0 corresponde a la operación sin ataques. Los siguientes casos corresponden a situaciones en las que se pueden generar ciberataques de integridad o DoS sobre cualquier sensor, siguiendo los modelos descritos por las Ecuaciones (30) y (31). En los casos de 1 a 4, solo se genera un ciberataque a la vez, mientras que los casos de 5 a 8 corresponden a ataques simultáneos.

Los intervalos de tiempo en los que ocurren los ciberataques se definieron de manera que el conjunto de datos estuviera equilibrado, por lo que se definieron de forma aleatoria y distribuida de manera uniforme. Los ataques de integridad se implementaron cambian-

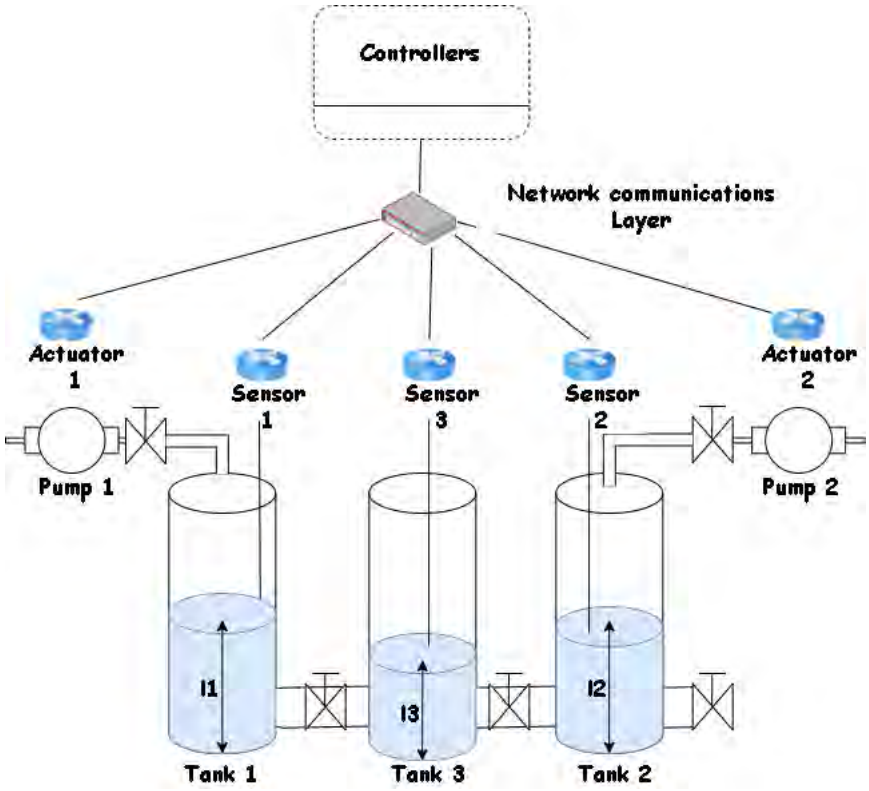


Fig. 47. Banco de pruebas de tanques interconectados.

TABLA XII.
Casos de ciberataques en el sistema de tanques.

Caso	Descripción
Caso 0	Operación normal
Caso 1	Ataque de integridad al sensor 1
Caso 2	Ataque de integridad al sensor 2
Caso 3	Ataque de DoS al sensor 1
Caso 4	Ataque de DoS al sensor 2
Caso 5	Ataque de integridad al sensor 1 y de DoS al sensor 2
Caso 6	Ataque de integridad al sensor 2 y de DoS al sensor 1
Caso 7	Ataque de integridad al sensor1 y 2
Caso 8	Ataque de DoS al sensor 1 y 2

do la variable modificada en un rango de $5\,\%$ a $8\,\%$ de su valor medido. Este rango de valores depende de la sensibilidad del sistema ya que habrá procesos particulares donde el efecto de la variación de las medidas en un rango dado no tenga tanto impacto como en otros. Todos los casos presentados corresponden a las clases que identificará el clasificador. La distribución de estos datos se muestra en la Fig. 48.

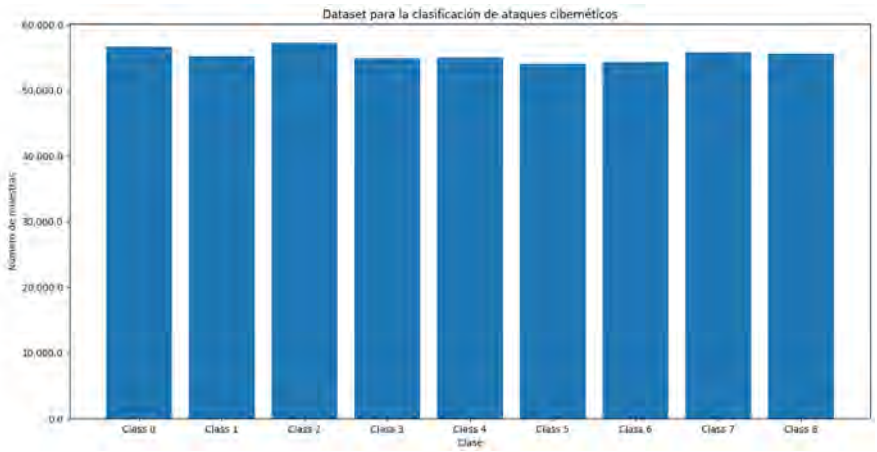


Fig. 48. Conjunto de datos para la clasificación de ataques cibernéticos.

❖Modelo de predicción

La Fig. 38 presenta la arquitectura implementada. El primer modelo genera la estimación de estados del proceso, mientras que se obtuvieron dos modelos más para reconstruir estados independientes x_1 y x_2, los cuales son los estados susceptibles de sufrir ciberataques.

Los datos de entrada del primer modelo están compuestos por cinco características, las cuales están compuestas por las medidas de los sensores y las acciones de control correspondientes al vector descrito en la Ecuación (59):

$$[x_1(k-1),x_2(k-1),x_3(k-1),u_1(k-1),u_2(k-1)]^T \tag{59}$$

El modelo tiene tres salidas correspondientes a los estados del proceso. El vector a reconstruir es la Ecuación (60):

$$[\hat{x}_1,\hat{x}_2,\hat{x}_3]^T = [x_1(k),x_2(k),x_3(k)]^T \tag{60}$$

Para estimar las salidas desacopladas, los datos de entrada se componen de cuatro características compuestas por las medidas de los sensores y las acciones de control. Para el primer estado desacoplado, los datos de entrada son definidos como la Ecuación (61).

$$[x_2(k-1),x_3(k-1),u_1(k-1),u_2(k-1)]^T \tag{61}$$

Esto genera una salida desacoplada estimada para x_1, Ecuación (62).

$$\hat{x}_{1d} = x_1(k) \tag{62}$$

Además, para estimar el segundo estado desacoplado del sistema, la estructura de los datos de entrada y la salida de estas redes se muestran en las Ecuaciones (63) y (64).

$$[x_1(k-1),x_3(k-1),u_1(k-1),u_2(k-1)]^T \tag{63}$$

$$\hat{x}_{2d} = x_2(k) \tag{64}$$

Donde k es el número de muestras.

Para estimar las salidas del sistema, se entrenaron arquitecturas basadas en LSTM y CNN 1D, usando Python y la biblioteca de Keras-Tensorflow. Para el entrenamiento del modelo de predicción se utilizó el 30 % y el 70 % de los datos para validar y entrenar. Para los modelos que se van a predecir se utilizó el error cuadrático medio (MSE) como función de pérdida.

A partir de esto, se generaron dos modelos para estimar todos los estados, y se generaron 4 modelos para estimar los estados 1 y 2 de forma desacoplada. La arquitectura que utiliza la red LSTM para estimar todos los estados está compuesta por 15 unidades como las que se muestran en la Fig. 8, la salida de esta red está conectada a una capa totalmente conectada que estima todos los estados. Las arquitecturas basadas en LSTM para estimar los estados desacoplados se componen de 100 unidades. Del mismo modo, se utiliza una capa completamente conectada en la salida, con una función de activación lineal. La función de activación utilizada para el LSTM fue ReLU y sigmoide para el paso recurrente.

Por otro lado, la arquitectura 1D basada en CNN utilizada para estimar todos los estados está compuesta de dos capas convolucionales, una capa 1D de agrupación promedio entre las capas convolucionales y una capa completamente conectada. La primera capa convolucional tiene un tamaño de kernel de cinco y tiene ocho filtros, mientras que la segunda capa tiene un tamaño de kernel de 3 con 16 filtros. Cada una de estas capas tiene una función de activación tangente hiperbólica. Entre las capas anteriores, hay una capa de agrupación promedio 1D con un tamaño de agrupación de 2 y zancadas de 2 con el mismo relleno. Entre las capas convolucionales y la capa completamente conectada, hay una capa de normalización por lotes con función de activación de tipo Leaky ReLU. En la capa completamente conectada, hay una capa de entrada de 48 neuronas y una capa de salida compuesta por 3 neuronas con una función de activación lineal para estimar los estados correspondientes. La función de pérdida utilizada fue MSE y el optimizador utilizado fue Adam. Para el entrenamiento, se dispuso de un máximo de

4 épocas y un tamaño de lote de 10 con una tasa de aprendizaje inicial de $0{,}01$. Para entrenar el modelo, se usaron $30\,\%$ de los datos para validar y $70\,\%$ para entrenar. Los diversos parámetros de las capas de esta red se encontraron de tal manera que se logrará el MSE más bajo posible, fue $0{,}000067$. El aumento del número de capas, neuronas, tamaño de filtro o número de filtros no se corresponde con una mejora significativa de la arquitectura propuesta. La arquitectura se muestra en la Fig. 49.

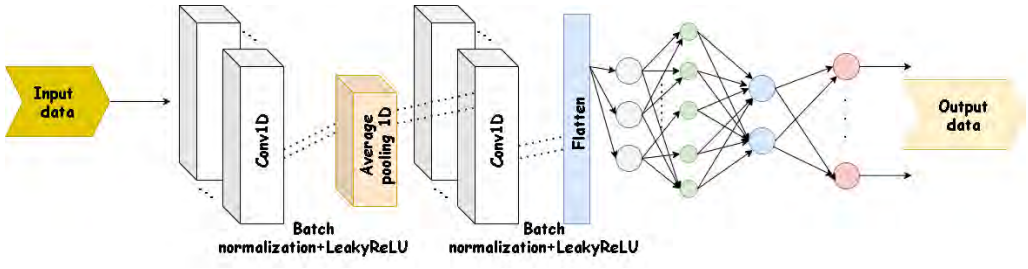


Fig. 49. Arquitectura CNN-1D para estimar todos los estados.

Mientras que las arquitecturas de desacoplamiento de estado basadas en CNN-1D se componen de 2 capas convolucionales y una capa completamente conectada. La primera capa convolucional tiene un tamaño de kernel de 4 y tiene 8 filtros, mientras que la segunda capa tiene un tamaño de kernel de 2 con 16 filtros. Cada una de estas capas tiene la función de activación tangente hiperbólica. Entre estas capas, hay una capa de agrupación promedio con un tamaño de agrupamiento de 2 y una zancada de 2 con el mismo relleno. Entre las capas convolucionales y la capa completamente conectada, hay una capa de normalización por lotes y una función de activación de tipo Leaky ReLU. Antes de la capa completamente conectada, se agregó una capa de exclusión $(0{,}15)$. En la capa completamente conectada, hay una capa de entrada de 32 neuronas y una capa de salida compuesta por 1 neurona con función de activación lineal para estimar el estado correspondiente. La función de pérdida utilizada fue MSE y el optimizador utilizado fue Adam. Para el entrenamiento, estaba disponible un máximo de 4 épocas y un tamaño de lote de 10 con una tasa de aprendizaje inicial de $0{,}01$. El $70\,\%$ de los datos se utilizó para entrenar el modelo y $30\,\%$ para validarlo. Los diversos parámetros de las capas de esta red se encontraron de tal manera que se logrará el MSE más bajo posible. El aumento del número de capas, neuronas, tamaño de filtro o número de filtros no se corresponde con una mejora significativa de la arquitectura propuesta.La arquitectura se muestra en la Fig. 50.

El resumen de todos los MSE se muestra en la Tabla XIII.

En cuanto al modelo de predicción, se observa que quienes utilizan LSTM tienen un MSE menor al momento de reconstruir todos los estados, mientras que en los estados

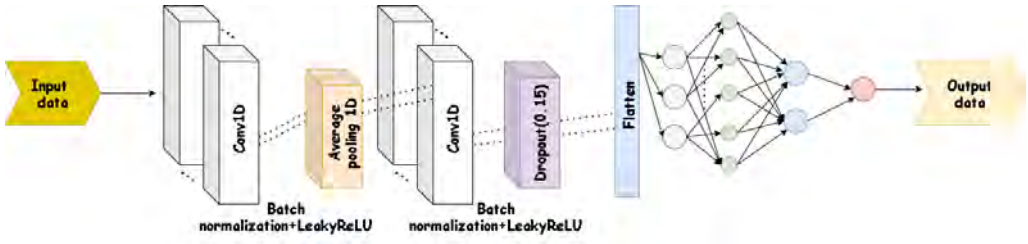


Fig. 50. Arquitectura CNN-1D para estimar los estados desacoplados.

TABLA XIII.
Resumen del MSE en los modelos de predicción.

Modelo de predicción	MSE de los datos de entrenamiento	MSE de los datos de validación
Sistema basado en LSTM (prediciendo todos los estados)	$7{,}995 \times 10^{-7}$	$1{,}113 \times 10^{-6}$
Sistema basado en CNN (prediciendo todos los estados)	$3{,}338 \times 10^{-5}$	$3{,}054 \times 10^{-6}$
Sistema basado en LSTM (estado desacoplado 1).	$3{,}321 \times 10^{-6}$	$5{,}281 \times 10^{-5}$
Sistema basado en CNN (estado desacoplado 1).	$4{,}549 \times 10^{-5}$	$1{,}732 \times 10^{-5}$
Sistema basado en LSTM (estado desacoplado 2).	$6{,}615 \times 10^{-6}$	$4{,}175 \times 10^{-6}$
Sistema basado en CNN (estado desacoplado 2).	$2{,}403 \times 10^{-5}$	$1{,}920 \times 10^{-5}$

desacoplados las arquitecturas basadas en CNN-1D tienen un menor MSE para el primer estado.

❖Modelo de clasificación

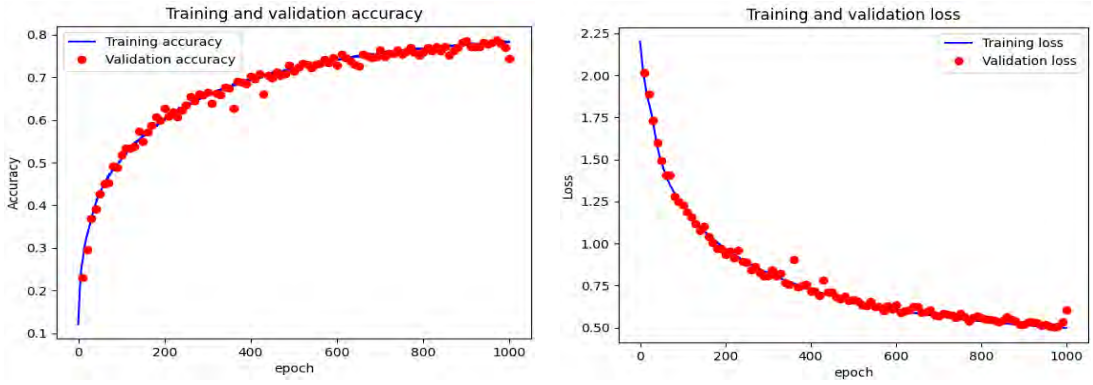
Para el clasificador se generaron 2 modelos, que combinan el uso de los modelos anteriores. El clasificador que utiliza LSTM está compuesto por 100 unidades con función de activación ReLU y función de activación sigmoidea para el paso recurrente mientras que la arquitectura propuesta para el clasificador basada en CNN-1D es similar a la que se muestra en la Figura 42. Está compuesta por tres capas convolucionales cuya función de activación es tangente hiperbólica. La primera capa convolucional tiene 80 con tamaño de núcleo de 15. La segunda y tercera capas convolucionales tienen el mismo tamaño de kernel, pero el número de filtros es 60 y 30, respectivamente. También hay una capa de normalización por lotes con la función de activación de Leaky ReLU. Finalmente, se utiliza una capa completamente conectada con una capa de entrada de 25 neuronas y una capa de salida con nueve neuronas correspondientes a las clases establecidas anteriormente. La última capa usa la función softmax. La función de pérdida utilizada fue CCE y el optimizador utilizado fue descenso de gradiente estocástico con impulso. Para la capacitación, se estableció un máximo de 1000 épocas, con un tamaño de lote de 10 y una tasa de aprendizaje inicial de $0{,}0001$. Para el entrenamiento del modelo, se usaron $30\,\%$ de los datos para validar y $70\,\%$ para entrenar. Los datos de entrada son

represnetados en la Ecuación (65).

$$[x_1, x_2, x_3, \hat{x}_1\, \hat{x}_2, \hat{x}_3, \hat{x}_{1d}, \hat{x}_{2d}, q_1, q_2, res, res_1, res_2]^T \tag{65}$$

Donde x_1, x_2, x_3 corresponden a las variables reales del proceso; $\hat{x}_1, \hat{x}_2, \hat{x}_3$ son las salidas estimadas por la arquitectura que se muestra en la Fig. 49; $\hat{x}_{1d}$ y $\hat{x}_{2d}$ corresponden a los estados desacoplados estimados por la arquitectura de la Fig. 50, q_1 y q_2 son los referencias de proceso y res, res_1 y res_2 son las señales residuales obtenidas al comparar los estados reales del proceso con los estados estimados, y la comparación individual entre los dos primeros estados reales del proceso y los estados desacoplados estimados, respectivamente.

Las Fig. 51(a) y 51(b) presentan la evolución de la función de costo y la métrica de precisión obtenida durante el procedimiento de entrenamiento de uno de los clasificadores.



(a) Exactitud (b) Función de pérdida

Fig. 51. Exactitud y función de pérdida durante el entrenamiento.

5.2.3.3 Resultados y discusión

En este caso se utilizó el lenguaje de programación Python y la biblioteca Keras para entrenar y obtener los resultados. Con el fin de evaluar el desempeño de esta arquitectura, se utilizaron las mismas métricas del caso anterior.

Los resultados obtenidos de los modelos basados en el aprendizaje profundo y los observadores de estado se muestran en la Tabla XIV. Como se puede apreciar, el mejor desempeño en base a las métricas establecidas se obtuvo con la arquitectura basada en CNN-1D completamente. De igual forma, estas arquitecturas fueron capaces tanto

de detectar los ciberataques propuestos, que corresponden a las clases 1 a 8, como de aislar cada uno de los ataques llevados a cabo en los dos sensores del sistema, identificando tanto el sensor que está siendo atacado, así como el tipo de ataque que se está llevando a cabo.

TABLA XIV.
Desempeño de los diferentes métodos.

Clase	Exactitud					Precisión					Recall					F1 Score				
	LO/UIOs	CNN/CNN	CNN/LSTM	LSTM/CNN	LSTM/LSTM	LO/UIOs	CNN/CNN	CNN/LSTM	LSTM/CNN	LSTM/LSTM	LO/UIOs	CNN/CNN	CNN/LSTM	LSTM/CNN	LSTM/LSTM	LO/UIOs	CNN/CNN	CNN/LSTM	LSTM/CNN	LSTM/LSTM
Clase 0	0.92	0.97	0.93	0.93	0.90	0.81	0.83	0.89	0.79	0.66	0.72	0.96	0.55	0.70	0.65	0.74	0.89	0.68	0.75	0.66
Clase 1	0.71	0.96	0.95	0.93	0.89	0.78	0.89	0.84	0.78	0.88	0.50	0.83	0.80	0.82	0.87	0.42	0.86	0.80	0.81	0.87
Clase 2	0.72	0.97	0.95	0.95	0.95	0.75	0.89	0.81	0.85	0.82	0.52	0.84	0.81	0.89	0.83	0.43	0.87	0.88	0.89	0.82
Clase 3	0.65	0.98	0.90	0.82	0.82	0.58	0.86	0.50	0.56	0.62	0.51	0.97	0.98	0.87	0.67	0.46	0.91	0.67	0.68	0.64
Clase 4	0.42	0.98	0.92	0.93	0.92	0.41	0.87	0.71	0.76	0.58	0.28	0.96	0.24	0.34	0.47	0.34	0.91	0.36	0.46	0.52
Clase 5	0.31	0.98	0.89	0.92	0.89	0.38	0.91	0.88	0.87	0.88	0.10	0.86	0.79	0.86	0.77	0.12	0.89	0.89	0.85	0.77
Clase 6	0.32	0.98	0.89	0.89	0.89	0.35	0.92	0.87	0.89	0.83	0.12	0.88	0.88	0.78	0.83	0.23	0.90	0.88	0.89	0.83
Clase 7	0.25	0.96	0.89	0.92	0.89	0.38	0.94	0.89	0.94	0.89	0.11	0.72	0.79	0.76	0.56	0.16	0.81	0.79	0.80	0.79
Clase 8	0.22	0.99	0.98	0.99	0.97	0.25	0.94	0.84	0.89	0.83	0.12	0.99	0.99	0.99	0.92	0.13	0.97	0.92	0.94	0.87

En la arquitectura propuesta, se pudo observar que los modelos basados en LSTM y CNN-1D tienen mejor desempeño que el método basado en observadores, debido a que estos últimos realizan su función a partir de una comparación con umbrales, que al ser fijos pueden generar falsas alarmas, mientras que en las redes LSTM y CNN-1D, esta evaluación se realiza de forma intrínseca, evitando la evaluación de los residuales con umbrales predefinidos. Asimismo, se observa que el método que utiliza observadores de estado es el de peor desempeño, generando una alta tasa tanto de falsos positivos como de falsos negativos.

Los valores de los índices obtenidos para la mejor arquitectura se presentan en las Fig. 52, 53(a), 53(b) y en la Tabla XV. Con respecto a la exactitud, las mejores puntuaciones se obtuvieron cuando ocurrieron ataques simultáneos con valores superiores a $0{,}97$. Lo anterior es un resultado importante porque esta situación ha sido poco explorada. En términos de recall, la clase 7 tiene una puntuación ligeramente justa, mientras que las otras situaciones tienen puntuaciones superiores a $0{,}83$. Además, el F1-Score también tiene valores altos. Los puntajes muestran que la arquitectura propuesta permite una alta especificidad y una alta sensibilidad.

El indicador de alarma se implementó desde el clasificador para conocer el estado del proceso. Dado que el clasificador proporciona la probabilidad de clasificar un dato de entrada en una clase particular, la señal de alarma se genera teniendo en cuenta el valor máximo obtenido del clasificador. En la Fig. 54, el indicador de alarma es 1 cuando

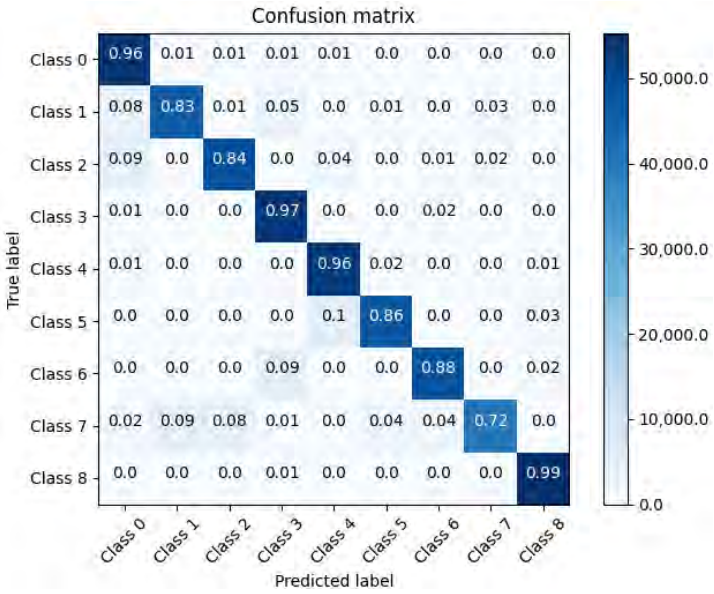
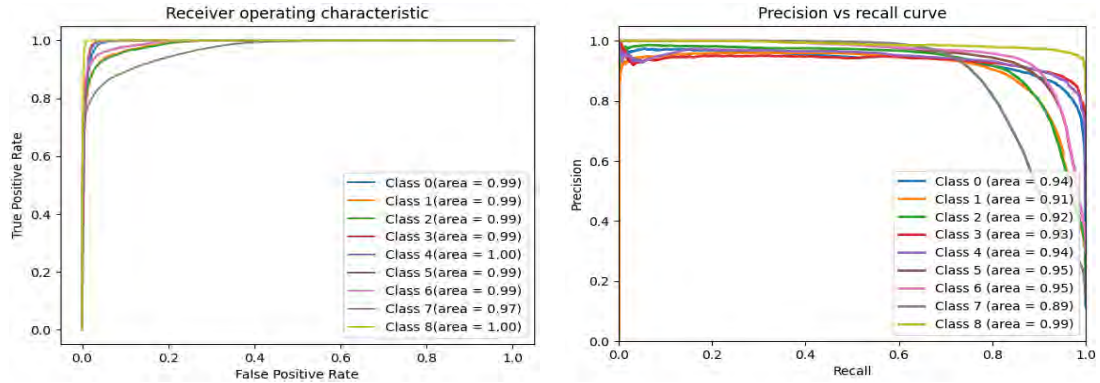


Fig. 52. Matriz de confusión para el sistema de tres tanques.



(a) Curva ROC (b) Curva precisión-recall

Fig. 53. Curvas ROC y de Precisión-Recall usando la arquitectura basada en CNN-1D para el sistema de tanques interconectados.

el sensor 1 o 2 está bajo ataque y 0 cuando no lo está. Además, se discrimina si el ataque es de tipo DoS o de integridad.

La respuesta del proceso cuando es atacado se muestra en la Fig. 55. Las casillas indican la instancia de tiempo, cuando el ataque ocurre en ambos sensores, de acuerdo con las señales de alarma generadas. Los recuadros rojos corresponden a ataques DoS y los recuadros negros corresponden a ataques de integridad.

Además, se observa una vez más que el efecto es diferente, dependiendo de si es un

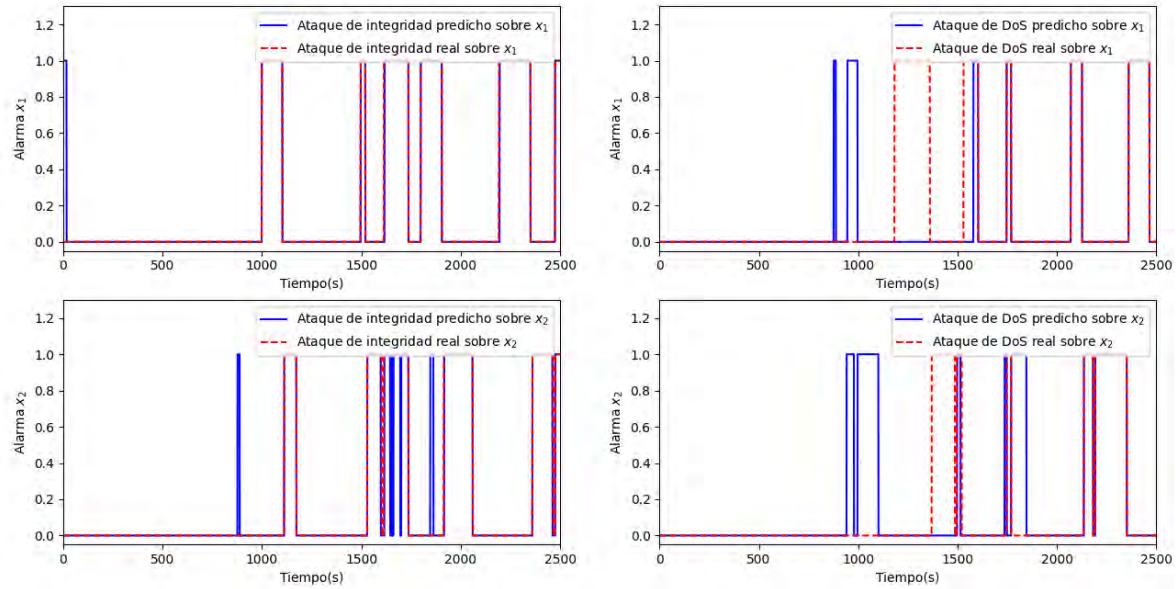


Fig. 54. Generación de alarmas.

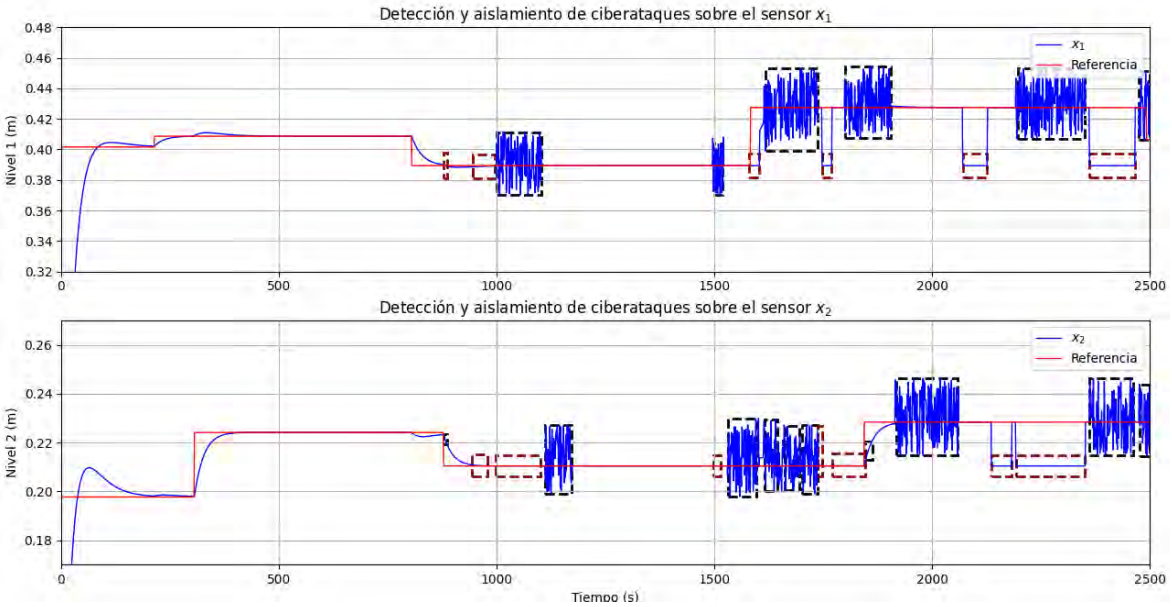


Fig. 55. Respuesta temporal del sistema de tanques bajo ataque.

TABLA XV.
Resumen de métricas para la arquitectura basada en CNN-1D.

	Exactitud	Precisión	Recall	F1 Score	TNR
Clase 0	0.97	0.83	0.96	0.89	0.98
Clase 1	0.96	0.89	0.83	0.86	0.99
Clase 2	0.97	0.89	0.84	0.87	0.99
Clase 3	0.98	0.86	0.97	0.91	0.98
Clase 4	0.98	0.87	0.96	0.91	0.98
Clase 5	0.98	0.91	0.86	0.89	0.99
Clase 6	0.98	0.92	0.88	0.90	0.99
Clase 7	0.96	0.94	0.72	0.81	0.99
Clase 8	0.99	0.94	0.99	0.97	0.99

ataque de DoS o un ataque de integridad. El sistema propuesto en este trabajo funcionó adecuadamente para detectar la ocurrencia del ciberataque, así como la ubicación de la parte del sistema que está siendo afectada y el tipo de ataque que está ocurriendo. Con los resultados obtenidos, se puede evidenciar que el uso de redes convolucionales 1-dimensional presentan un mejor desempeño que arquitecturas que usan redes RNN o LSTM, por esto y los resultados obtenidos en el dataset del SWaT, se eligieron redes convolucionales 1-dimensional para el desarrollo de sistemas de detección y aislamiento de ciberataques que puedan estar presentes en los procesos de control automático inmersos en sistemas ciberfísicos.

5.3 ARQUITECTURA PARA EL DESARROLLO DE SISTEMAS CIBERFÍSICOS Y VERIFICACIÓN DE REQUISITOS TEMPORALES

En esta sección se presenta la arquitectura del sistema ciberfísico, así como la arquitectura de los nodos que lo conforman y su respectivo análisis de planificabilidad para verificar los requisitos temporales de las aplicaciones.

5.3.1 Arquitectura del sistema ciberfísico

En la Fig. 56 se muestra el esquema general del CPS en conjunto con el sistema de detección de ciberataques. El vector de ataques definido como $\mathcal{A} = \{a_k^u, a_k^y\}$ son ciberataques que pueden tener las características presentadas en la Sección 4.2 para los ataques de integridad y ataques de DoS que pueden llegar a afectar las mediciones del

proceso así como las acciones de control en el tiempo k.

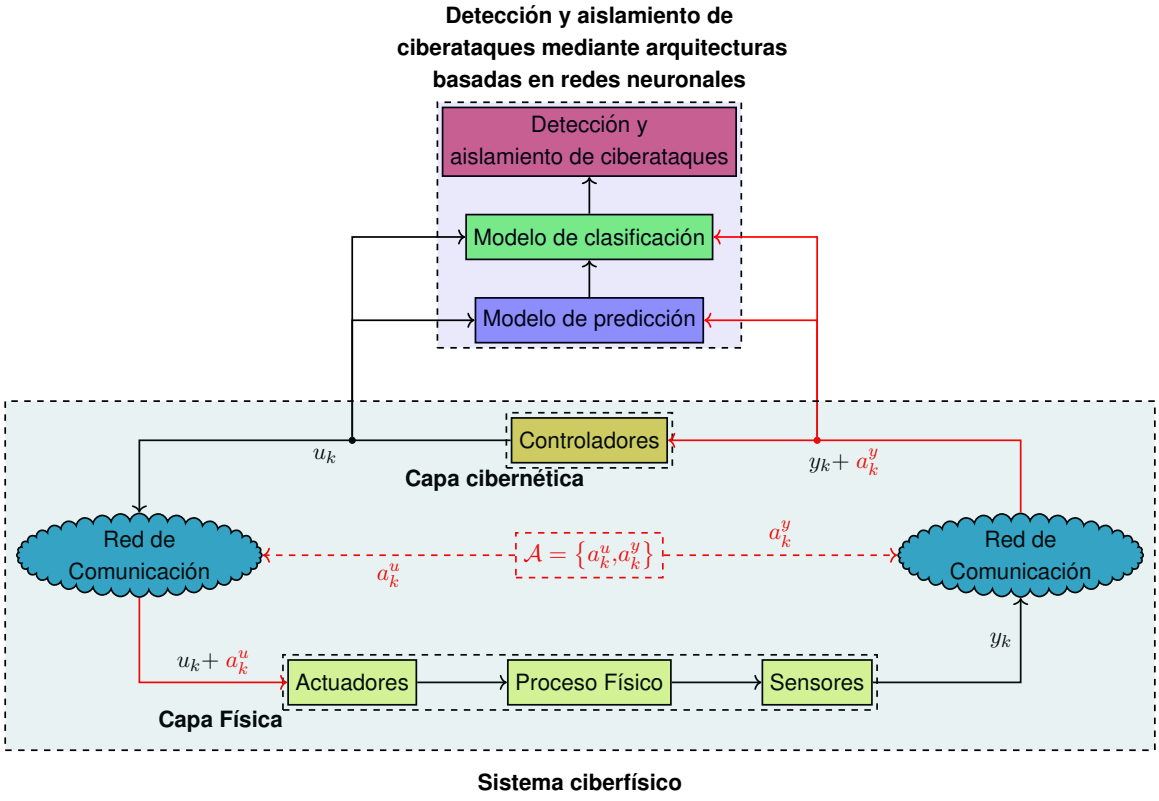


Fig. 56. Sistema ciberfísico con el sistema de detección.

Este esquema, puede tener en cuenta también las perturbaciones a las cuales puede estar sometido el proceso, así como la incertidumbre que se tiene a la hora de encontrar el modelo dinámico del proceso y el ruido asociado al proceso de medición.

5.3.2 Arquitectura de los nodos que conforman el sistema ciberfísico

La arquitectura propuesta para los nodos que conforman el sistema se muestra en la Fig. 57. Esta arquitectura se soporta en un nivel de virtualización en donde se usa un motor de contenedores con lo cual se obtiene una elevada flexibilidad en cuanto a las interfaces y facilidad de integración de las aplicaciones. Las aplicaciones que requieran ser virtualizadas para obtener las características de aislamiento espacial y temporal pueden ser ubicadas en contenedores cuyos recursos son gestionados y asignados por su respectivo motor, ejemplo de estas aplicaciones pueden ser las que lleven a cabo las tareas de medición y control. Por otro lado se pueden ejecutar aplicaciones sobre el sistema operativo host, como lo pueden ser las aplicaciones relacionados con el sistema de detección de ciberataques así como interfaces de usuario.

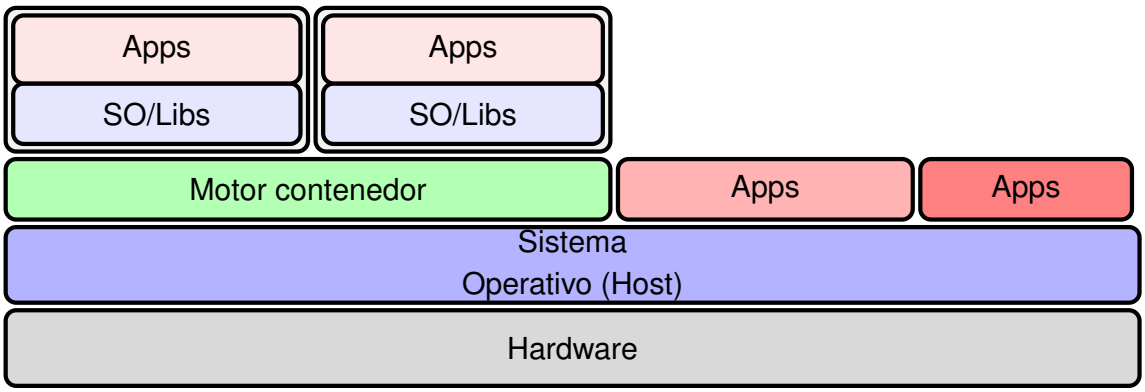


Fig. 57. Arquitectura propuesta para los nodos de la red de control.

Para mantener una reconfiguración rápida y flexible del sistema y mantener habilitada la funcionalidad plug-and-play, los componentes deben admitir la reconfiguración y la adición o eliminación en línea, sin perder ni degradar la solidez de desarrollos anteriores. Por ello en [192] se propuso un marco de microservicio basado en componentes que planea el diseño de soluciones a partir de componentes interconectados para soportar flexibilidad, interoperabilidad y robustez en las aplicaciones exigentes y flexibles de la Industria 4.0.

Como se mencionó en la sección 3.1.2, la política de planificación EDF está tomando cada vez más relevancia debido a sus beneficios en cuanto al manejo de los recursos del sistema, por lo que en esta propuesta se acoge esta política para la planificación de las particiones.

Los microservicios basados en componentes integran los aspectos de comunicación, computación, recursos de almacenamiento, así como capacidades en tiempo real, mediante el uso de tecnologías de contenedores y la implementación de microservicios, así como el aislamiento computacional entre ellos. Se usa un middleware basado en política de publicación/suscripción (publish/subscribe) mediante tópicos, que permite el desacoplamiento de cada microservicio y define un patrón constructivo del software a desarrollar.

En este marco, los microservicios son software integrado en contenedores que permite mejorar la flexibilidad, escalabilidad y portabilidad. Por lo tanto, un contenedor puede implementar uno o más microservicios. Un contenedor representa un contexto en donde se ejecuta algún código o se almacenan algunos datos.

Adicional al soporte de los requisitos indicados, la propuesta incluye aspectos relevantes que permite el desarrollo de aplicaciones con código que son más fáciles de mantener

por la separación de los servicios, se puede actualizar y escalar en diferentes lenguajes de programación, e incluso permite diferentes servicios y niveles de datos, usando por ejemplo un Servicio de Distribución de Datos para sistemas de tiempo real (DDS, Data Distribution Service por sus siglas en inglés), que especifica un middleware que permite desarrollar sistemas distribuidos en tiempo real de forma estandarizada. Este utiliza el paradigma publicación/suscripción, permitiendo disminuir el acoplamiento entre entidades mejorando la eficiencia, flexibilidad, escalabilidad y adaptabilidad.

Para lograr los desafíos mencionados anteriormente, la propuesta se basa en un enfoque holístico, en el que los servicios de oferta y solicitud son globalmente integrados como microservicios, aislados mediante tecnología de contenedores e interconectados con un middleware basado en la patrón de publicación/suscripción como el Servicio de distribución de datos.

Desde un concepto plug-and-play, todos los componentes relacionados con la medición de las variables del proceso como lo son los sensores (analógicos y digitales) se comunican con los componentes donde se implementan los controladores, HMI y base de datos. El controlador debe recibir desde las HMIs las referencias establecidas por los usuarios, aunque pueden venir desde otro lugar, como datos de sensores, y a su vez publicar las respectivas acciones de control. Cinco estructuras de mensajes diferentes fueron considerados para representar la comunicación entre los componentes y microservicios, sin embargo esto se puede ampliar o modificar dependiendo de las necesidades requeridas. Esta estructura se observa en Tabla XVI donde se representa la comunicación entre componentes y microservicios.

TABLA XVI.
Modelo de datos canónicos para el intercambio de información.

Tópicos E/S	**Tópicos de referencias**	**Tópicos de tolerancia al fallo**	**Tópicos de alarmas**	**Tópicos de controladores**
struct topic1{	struct topic2{	struct topic3{	struct topic4{	struct topic5{
string id;	string id;	string id;	string id;	string id;
		string time;	string time;	
bool onoff;			bool onoff;	float64 data;
float64 data;	float64 data;	float64 data;	float64 data;	
}	}	}	}	}

Los diferentes componentes, microservicios y contenedores de la arquitectura así como la relación entre ellos son presentados en la Fig. 58.

El servicio de base de datos mantiene toda la información posible sobre el estado del sistema y se puede consultar a través de los servicios http. La implementación de es-

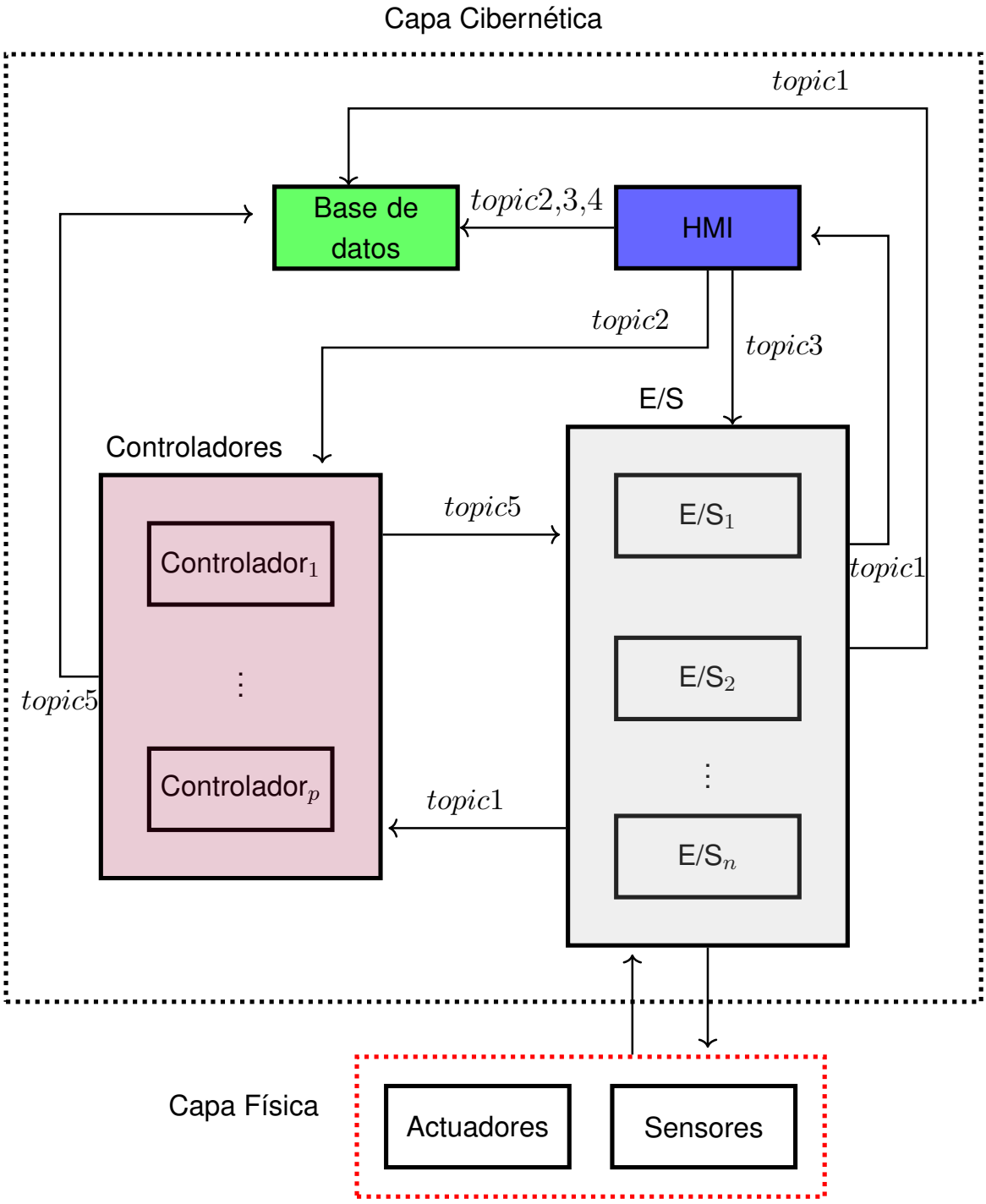


Fig. 58. Componentes, microservicios y contenedores de la arquitectura.

te servicio se puede lograr con una serie de contenedores, implementando una base de datos distribuida y organizada según las necesidades de la aplicación. Por lo tanto, las bases de datos se pueden implementar como micro bases de datos por las áreas o aplicaciones desarrolladas. La flexibilidad y escalabilidad del marco permite la implementación de nuevos datos o series de datos.

Las entradas/salidas (E/S) del sistema se implementan mediante una serie de contenedores que se conectan directamente con los dispositivos finales a través de un bus de campo (Modbus, ISP, I2C, Profibus, etc.) con sensores y actuadores, a través de tareas de medición y de control. Por lo tanto, puede recibir y/o enviar información a otros componentes. Cada contenedor implementa un servicio que solicita periódicamente la lectura de un valor de sensor y lo comunica a otros servicios. El nombre del servicio se identifica por el tipo de variable del sensor, y los parámetros a configurar para este servicio son el período y el identificador de la variable a ser medida. Los servicios del actuador reciben comunicación de los controladores e implementan el acceso directo a los actuadores físicos. La idea general es leer los sensores asociados con un contenedor y luego ofrecer cada sensor como un servicio para clientes que lo necesitan. El enfoque propuesto define un nuevo patrón de construcción de software basado en técnicas plug-and-play, donde los componentes admiten la reconfiguración en línea, así como la adición o remoción de componentes, sin degradar la robustez de desarrollos anteriores.

Igualmente, se definen contenedores para el desarrollo e implementación de diferentes algoritmos de control. Algunos controladores clásicos en automatización industrial pueden ser definidos o adicionados según las necesidades de la aplicación. En un sistema en continuo crecimiento y adaptación a los requisitos y necesidades de la industria, los consumidores y los mercados, los controladores y otros componentes pueden cambiar muy fácilmente o incluso cada vez incluir nuevas funcionalidades o eliminar algunas existentes sin alterar la robustez del sistema. Este componente siempre deja abierta la posibilidad de mejorar e implementar nuevos controladores.

Adicionalmente, se tiene un contenedor que implementa un servicio de interfaz hombre-máquina (HMI), capturando y ofreciendo información de los usuarios al sistema informático. La flexibilidad en el manejo de los datos permite que diferentes tecnologías se pueden adicionar.

Esta arquitectura posibilita la implementación de métodos de detección de ciberataques así como estrategias para mitigar el efecto de los ataques, por cuanto una vez se identifique la ocurrencia de un ataque es posible indicar qué microservicio/componente esta siendo afectado y no se ofrezca más ese servicio. Para lo cual se propone:

• Disponer de réplicas de la arquitectura, por si alguna falla reemplazarla por otra que esté funcionando de manera correcta. Para lograr esto, en el modelo de publicación/-suscripción que se usa en esta propuesta, todos los subscriptores del tópico reciben inmediatamente cualquier mensaje publicado en este. El intercambio de información en este paradigma se puede utilizar para habilitar arquitecturas basadas en eventos o para desacoplar aplicaciones a fin de aumentar el rendimiento, la confiabilidad y la escalabilidad. Este paso se puede llevar a cabo gracias al sistema de detección, el cuál estará en la capacidad de identificar el componente o microservicio que esta siendo afectado. Así, el motor del contenedor el cuál es el responsable de definir los servicios y los tópicos a los que los subscriptores y publicadores tienen acceso, permite parar la ejecución de las aplicaciones involucradas y ejecutar su respectiva copia que en el momento no está siendo afectada.

• Reducir las funcionalidades que estén expuestas en el exterior en particiones y concentrar en una de ellas las comunicaciones exteriores (servidor de datos), y de esta forma usar los mecanismos internos que ofrece el hipervisor para comunicar las diversas aplicaciones.

Un ejemplo sencillo en un proceso de control automático donde se disponga de un sensor, un controlador, un actuador y una interfaz de interacción (HMI) puede ilustrar el proceso de diseño. Un contenedor llamado Sensor dispone de una aplicación que permite leer los datos del sensor físico y lo publica periódicamente. Por su parte un contenedor llamado Controlador se suscribe al sensor y las referencias y a partir de esto desarrolla una acción de control (publicar periódicamente) siguiendo algún tipo de algoritmo de control. Además, se tiene un contenedor llamado Actuador que se suscribe al tópico de control y envía los datos al actuador. Finalmente, la HMI muestra todos los variables en la pantalla y captura las referencias de un usuario, el resultado final de este ejemplo de diseño se muestran en la Tabla XVII y en la Fig. 59.

Cabe destacar que dentro de la misma HMI se puede tener dispuesto el sistema de detección de ciberataques o si es necesario tener otro servicio que ofrezca las funcionalidades de este sistema.

La arquitectura de componentes puede ser diseñada para promover la escalabilidad, además de que permite ofrecer a los microservicios y las aplicaciones el aislamiento temporal y espacial entre ellas. Además, esta plataforma permite tener diseños modulares que facilita actualizaciones de funcionalidades individuales sin afectar los desarrollos ya existentes. Además permite el soporte de aplicaciones que tengan diferentes nivele de criticidad.

TABLA XVII.
Proceso de diseño de los componentes plug-and-play.

Contenedor	Tópico	Tipo	Servicio
Sensor	*sensor*	Periódico	Publicar
Actuador	*control*	Eventual	Subscribir
Controlador	*set_point*	Eventual	Subscribir
	sensor	Eventual	Subscribir
	control	Periódico	Publicar
HMI	*set_point*	Eventual	Publicar
	sensor	Eventual	Subscribir
	control	Eventual	Subscribir

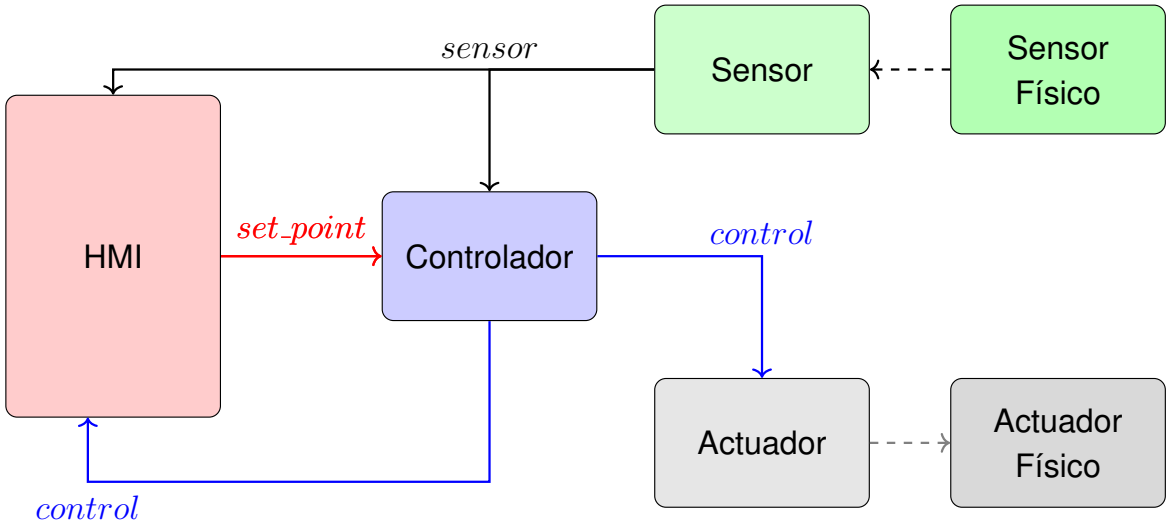


Fig. 59. Ejemplo de aplicaciones contenedorizadas.

5.3.3 Análisis de planificabilidad

Teniendo en cuenta los planteamientos presentados en la sección 3.1.2, para el análisis de planificabilidad de la solución se plantean dos evaluaciones, una evaluación local de los componentes implementados en un mismo nodo y una evaluación extremo-extremo.

Para el análisis de planificabilidad local, considerando la arquitectura de nodo propuesta en este trabajo, consiste en verificar el cumplimiento de los plazos individuales para D_M, D_C y D_A a partir de la Ecuación (1); donde D_M, D_C y D_A son los plazos de las tareas

de Medición, Control y Actuación respectivamente.

Desde el punto de vista de la aplicación de control, en un esquema general los componentes Medición, Control y Actuación, se ejecutan en dicha secuencia y en exclusión mutua (no se puede iniciar la ejecución de la siguiente función si la anterior no ha finalizado). Además, en consecuencia con la arquitectura de síntesis de estos sistemas para buscar una defensa en profundidad, en este trabajo se supuso que todos los nodos del CPS se encuentran conectados a la misma red, y se supone que el tiempo del peor caso de ejecución de los componentes Medición y Control incluye el tiempo de ejecución del componente y el tiempo del envió del mensaje al componente siguiente. En ese sentido el test de planificabilidad extremo-extremo consiste en verificar el cumplimiento de la Ecuación (66).

$$D_{CG} \geq D_M + D_C + D_A \tag{66}$$

Donde D_{CG}, es el plazo extremo-extremo, entre el inicio de la medición y la finalización de la actuación, impuesto por los objetivos de desempeño del algoritmo de control.

5.4 Conclusiones

Este capítulo permitió abordar el segundo objetivo específico, donde se trataron los diferentes aspectos a tener en cuenta para la propuesta del procedimiento de diseño de sistemas ciberfísicos para aplicaciones de control. La cual se soporta en nueva arquitectura que permite la integración de un sistema de detección y aislamiento de ciberataques que se validó a partir del abordaje de casos con bancos de prueba que han sido usados en trabajos similares, logrando de esta forma tener un acercamiento a arquitecturas que permitan tolerar ciberataques en estas aplicaciones.

Se presentó una nueva arquitectura para la detección y aislamiento de ciberataques de DoS e integridad en sistemas ciberfísicos utilizando Redes Neuronales Convolucionales 1-dimensionales, superando así otros modelos basados en aprendizaje automático y métodos basados en modelos, como el uso de Observadores de Entrada Desconocida. Esta arquitectura implica una serie de pasos para lograr su propósito. El primer paso es generar una salida estimada del proceso bajo un modelo de regresión. El siguiente paso es generar una señal residual bajo la comparación de las salidas medidas del proceso con las salidas estimadas. A continuación, se añadió un modelo de clasificación cuyos datos de entrada son diferentes características, como las acciones de control, las salidas estimadas, las salidas medidas del proceso y las señales residuales. Este modelo permitió detectar y aislar diferentes eventualidades que se definieron en clases. Finalmente, a partir de la clase detectada, se generaron señales de alarma que se utilizan para informar de la ocurrencia de un ciberataque, permitiendo definir el tipo de ataque y

la parte del sistema que está siendo afectada por el mismo.

La arquitectura propuesta para el sistema de detección no utiliza información de umbrales para detectar y aislar los ataques, como es el caso de los métodos basados en modelos, como los Observadores de Entrada Desconocida, que suelen utilizar esta información. Estos modelos requieren una selección exhaustiva de estos umbrales, lo que puede provocar tanto falsas detecciones como situaciones anómalas que pasan desapercibidas. Mientras que la arquitectura propuesta proporciona ventajas sobre esto.

El rendimiento de la arquitectura propuesta para el sistema de detección, fue validada por dos bancos de pruebas obteniendo resultados satisfactorios en comparación con otros métodos. Los resultados sobre el conjunto de datos SWaT permitieron observar qué en términos de precisión y exactitud, los índices presentan puntuaciones altas, obteniendo estos una puntuación de $0{,}95$ en promedio. En cuanto a las métricas recall y F1 Score, presentó una puntuación de $0{,}95$, que supera a los métodos propuestos anteriormente por un buen margen. En general, el sistema propuesto tiene una alta tasa de verdaderos positivos y una baja tasa de falsos positivos.

Por otro lado, se destaca la capacidad del sistema para poder detectar y aislar ciberataques que pueden ocurrir simultáneamente en diferentes partes del sistema, lo cual se presentó en el banco de pruebas del sistema de tres tanques. En las clases definidas, la exactitud presenta puntuaciones superiores a $0{,}96$ y la precisión es superior a $0{,}83$, en los casos en que los ataques se producen en una sola parte del sistema, mientras que la puntuación es superior a $0{,}91$ en los casos en que se producen ataques simultáneos. En cuanto a la métrica de F1 Score, las puntuaciones son superiores a $0{,}81$, lo que supone un resultado muy prometedor. Por último, en lo que respecta a la métrica de recall, las puntuaciones son superiores a $0{,}83$, en la mayoría de los casos. Con los casos presentados en este banco de pruebas, se ha podido demostrar la capacidad de la arquitectura propuesta para detectar y localizar ataques que pueden producirse simultáneamente. Esto es interesante porque este tipo de experimentos rara vez se realizan, y mucho menos proporcionan pruebas de sistemas que pueden detectar este tipo de situaciones, que no son ajenas a las eventualidades que pueden ocurrir en la realidad. En los dos casos resaltados, hubo una alta tasa de TNR en cada una de las clases, que oscilaba entre $0{,}98$ y $0{,}99$.

Se propuso una arquitectura para el diseño de sistemas ciberfísicos en donde se puede llegar a tener procesos de automatización flexibles y escalables con el fin de reducir la brecha entre las arquitecturas genéricas y las implementaciones físicas que se tienen actualmente. La arquitectura se basa en un enfoque holístico, en el que un sistema y sus propiedades se analizan como un todo, de forma global e integrada y no sólo como la

simple suma de sus partes. Para lograr diseños e implementaciones rápidas y robustas en los diferentes sectores donde se encuentran este tipo de sistemas, se propone un patrón de diseño de software basado en componentes, la tecnología de contenedores, los conceptos de microservicios y el paradigma publicar/subscribir (publish/subscribe). De este modo se diseña un conjunto de componentes que ofertan y solicitan servicios que pueden ser fácilmente interconectados entre ellos utilizando la técnica plug-and-play.

Así mismo se plantea el análisis de planificabilidad para verificar los requisitos temporales de las aplicaciones dispuestas en la arquitectura. Este análisis permite evaluar localmente los componentes implementados en un mismo nodo así como la evaluación extremo-extremo de la aplicación.

Los resultados expuestos han sido publicados en [192, 193].

6. IMPLEMENTACIÓN DE CASOS DE ESTUDIO

En este capítulo se presenta la manera en la que se abordó el desarrollo de dos casos de estudio utilizando el procedimiento propuesto en este trabajo. Los resultados obtenidos permiten observar la validez de la propuesta realizada. El primero de los casos aborda el banco de pruebas de los tanques interconectados presentado en la sección 5.2.3, mientras que en el segundo se trató un sistema de control de pH de la empresa Punta Delicia, localizada en Colima, México.

En lo que respecta a la arquitectura de los nodos, para el soporte de las particiones se realizaron evaluaciones con dos tecnologías, RTLinux y Singularity sobre Linux. Si bien Singularity no utiliza una política de planificación para abordar soluciones de tiempo real crítico (también denominados hard real time), su gran flexibilidad en cuanto a las interfaces y la facilidad que ofrece para la integración de las aplicaciones ha llevado a que tenga un uso elevado; además, como se presentó en la revisión del estado actual de las tecnologías de desarrollo, los resultados reportados muestran un buen desempeño de esta tecnología en el tratamiento de sistemas no críticos de tiempo real (también denominados soft real time).

6.1 CASO DE ESTUDIO: SISTEMA DE TANQUES INTERCONECTADOS

De la descripción detallada del sistema se pueden identificar diferentes componentes para ofrecer los servicios requeridos para el funcionamiento del proceso. Uno de los componentes definidos en este caso son los sensores de nivel de los tanques que se virtualizarán en tareas asociadas al proceso de medición, las cuales publican estas mediciones de forma periódica. De igual forma el sistema dispone de dos actuadores, en este caso válvulas que son reguladas por dos controladores. De este modo se disponen de dos contenedores más relacionados con los actuadores que permitirán la conexión directa con las válvulas y que se subscriben a los contenedores donde se ejecuta el algoritmo de control. Así mismo se tendrá un contenedor donde se dispone de la tarea de control donde se ejecuta el algoritmo de control establecido en la sección 5.2.3, el cual publica las acciones de control de forma periódica. Adicionalmente se dispondrá de un sistema de detección de ciberataques, que requiere la información tanto de los sensores, las referencias como de las acciones de control, y así generan alarmas a partir de los resultados del sistema de detección. Estas alarmas se publican periódicamente. Finalmente se tiene un sistema de monitoreo que define las referencias a las cuales se quiere llevar al sistema y permite visualizar las variables de interés del sistema. Los diferentes componentes virtualizados así como los servicios que ofrecen cada uno de ellos se describen en la Tabla XVIII.

TABLA XVIII.
Proceso de diseño de los componentes plug-and-play.

Contenedor	Tópico	Tipo	Servicio
Sensor 1	l_1	Periódico (1s)	Publicar
Sensor 2	l_2	Periódico (1s)	Publicar
Sensor 3	l_3	Periódico (1s)	Publicar
Actuador 1	u_1	Eventual	Subscribir
Actuador 2	u_2	Eventual	Subscribir
Controlador	set_point	Eventual	Subscribir
	l_1	Eventual	Subscribir
	l_2	Eventual	Subscribir
	l_3	Eventual	Subscribir
	u_1	Periódico (1s)	Publicar
	u_2	Periódico (1s)	Publicar
Sistema de detección	set_points	Eventual	Subscribir
	l_1	Eventual	Subscribir
	l_2	Eventual	Subscribir
	l_3	Eventual	Subscribir
	u_1	Eventual	Subscribir
	u_2	Eventual	Subscribir
	$alarmas$	Periódico (1s)	Publicar
Monitoreo	set_points	Eventual	Publicar
	l_1	Eventual	Subscribir
	l_2	Eventual	Subscribir
	l_3	Eventual	Subscribir
	u_1	Eventual	Subscribir
	u_2	Eventual	Subscribir
	$alarmas$	Eventual	Subscribir

El diseño de la arquitectura para este caso se muestra en la Fig. 60. Los módulos encontrados en el bloque E/S se conectan directamente a los dispositivos físicos donde se captura las mediciones así como los actuadores del proceso. En este caso el proceso físico es emulado a partir de un modelo descrito por el modelo dinámico del sistema. El acceso a la emulación se realiza a través del protocolo TCP/IP soportado en una red Ethernet.

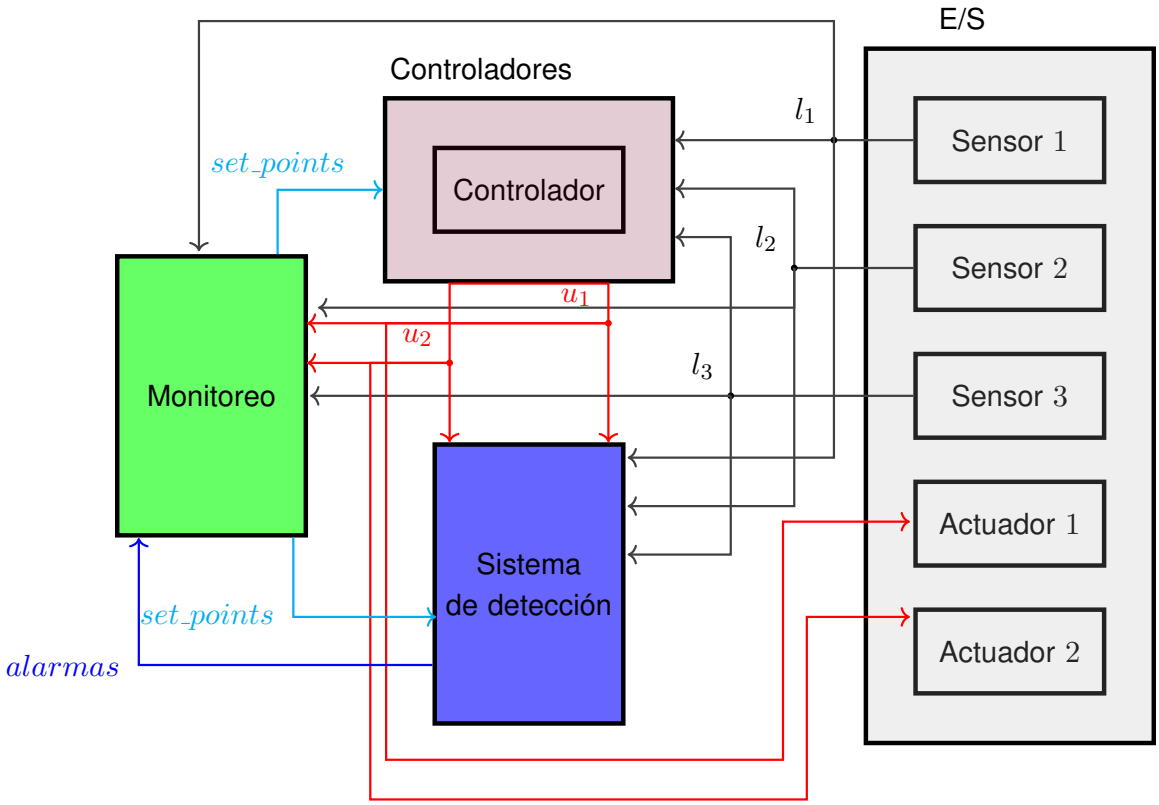


Fig. 60. Arquitectura basada en componentes para el sistema de tanques interconectados.

Las aplicaciones que están contenedorizadas presentan los algoritmos 2, 3, 4 y 5.

6.1.1 Resultados y discusión

En esta sección, se presentan la verificación del caso práctico desarrollado con el marco propuesto y se realiza una comparación cualitativa con las implementaciones tradicionales. Es importante destacar que el objetivo principal es verificar la funcionalidad del procedimiento de diseño propuesto así como la arquitectura basada en microservicios en donde la comunicación se realiza a través de un middleware que integra la tecnología de contenedores y el enfoque de publicar/subscribir a través de tópicos.

Algorithm 2: Algoritmo de medición

Input: Salidas del proceso y tiempo de muestreo: x_1,x_2,x_3, T_s

Output: Salidas del proceso: l_1,l_2, l_3

Inicialización : Condiciones iniciales

1: $k \leftarrow 0$
2: **while** (1) **do**
3: leer valor del sensor $x_{1,2,3}$ en $t \leftarrow kT_s$
4: $l_1 \leftarrow x_1(t)$
5: $l_2 \leftarrow x_2(t)$
6: $l_3 \leftarrow x_3(t)$
7: $k \leftarrow k + 1$
8: Esperar al siguiente tiempo de muestreo $(t \leftarrow kT_s)$
9: **end while**
10: **return** publicar l_1,l_2,l_3

La implementación es realizada en diferentes plataformas. El sistema de monitoreo y el sistema de detección, se ejecuta sobre un computador con procesador Intel Core i5-7300HQ- 2.5GHz. Mientras que los módulos que implementan el algoritmo de control, los dispositivos de E/S y la emulación del proceso, se ejecutan sobre contenedores virtualizados en el sistema embebido Raspberry 4piB.

En la Fig. 61 se observan la respuesta temporal de las variables del proceso que son susceptibles a ciberataques, así como la generación de las alarmas en este caso para ataques de tipo integridad. Se puede observar que las variables del proceso en los intervalos de tiempo donde no hay ocurrencia de ciberataques, son capaces de seguir la referencia establecidas, esto permite observar que el algoritmo de control implementado bajo esta arquitectura, en situaciones normales, permite llevar el proceso a los estados deseados, garantizando la funcionalidad del mismo así como los servicios asociados a la medición y a la actuación. Por otra parte, se destaca la capacidad del sistema de detección a la hora de generar la alarma, mostrando una vez más la fortaleza que tiene el sistema de detección planteado y la posibilidad de generar alarmas una vez se encuentra que algún componente del sistema está bajo ataque.

El análisis comparativo con otras implementaciones tradicionales se muestran en la Tabla XIX. Los microservicios y las aplicaciones se encuentran aisladas tanto temporal como espacialmente, una de otra. La implementación muestra que no está limitada a nuevos tipos de datos, nuevas estrategias de control, nuevos métodos de visualización, entre otras características que se deseen adicionar al proceso. Además permite tener diversas funciones que pueden ser programadas en diferentes lenguajes de programación. El

Algorithm 3: Algoritmo de control

Input: Referencias,salidas del proceso y tiempo de muestreo: $q_1, q_2, l_1, l_2, l_3, T_s$

Output: Acciones de control: u_1,u_2

Inicialización : Condiciones iniciales

$k \leftarrow 0$, $ai_{1_{k1}} \leftarrow 0$, $ai_{2_{k1}} \leftarrow 0$

$K1_{11} \leftarrow 21{,}6$, $K1_{12} \leftarrow 3$, $K1_{13} \leftarrow -5$, $K1_{21} \leftarrow 2{,}9$, $K1_{22} \leftarrow 19$, $K1_{23} \leftarrow -4$

$K2_{11} \leftarrow -0{,}95$, $K2_{12} \leftarrow -0{,}32$, $K2_{21} \leftarrow -0{,}3$, $K2_{22} \leftarrow -0{,}91$

while (1) **do**

  Subscribirse al tópico l_1,l_2,l_3

  Subscribirse al tópico $set\_points \leftarrow q_1$,$q_2$

  $e_1 \leftarrow q_1 - l_1$

  $e_2 \leftarrow q_2 - l_2$

  $ai_1 \leftarrow e_1 + ai_{1_{k1}}$

  $ai_2 \leftarrow e_2 + ai_{2_{k1}}$

  $ui_1 \leftarrow K2_{11} * ai_1 + K2_{12} * ai_2$

  $ui_2 \leftarrow K2_{21} * ai_1 + K2_{22} * ai_2$

  $ui_1 \leftarrow -ui_1$

  $ui_2 \leftarrow -ui_2$

  $up_1 = K1_{11} * l_1 + K1_{12} * l_2 + K1_{13} * l_3$

  $up_2 = K1_{21} * l_1 + K1_{22} * l_2 + K1_{23} * l_3$

  $up_1 = -up_1$

  $up_2 = -up_2$

  $u_1 = 10^{-4} * (ui_1 + up_1)$

  $u_2 = 10^{-4} * (ui_2 + up_2)$

  $k \leftarrow k + 1$

  Esperar al siguiente tiempo de muestreo $(t \leftarrow kT_s)$

end while

return publicar u_1,u_2

diseño permite la modularidad entre componentes, donde fácilmente se pueden tener actualizaciones de funcionalidades individuales.

6.1.2 Análisis de planificabilidad

Los contenedores que se usaron en el desarrollo de la propuesta usan políticas tipo FIFO (First In First Out). Se realizará una comparación del rendimiento de las aplicaciones soportadas sobre esta arquitectura y sobre RTLinux, el cuál usa un planificador del tipo EDF.

Algorithm 4: Sistema de detección

Input: Salidas del proceso, acciones de control, referencias: x_1,x_2,x_3, u_1,u_2, q_1, q_2

Output: Alarmas: $a_{s1-DoS}, a_{s2-DoS}, a_{s1-int}, a_{s2-int}$

Inicialización : Condiciones iniciales

$k \leftarrow 0$
Cargar modelo de predicción de las salidas del sistema mod_pred
Cargar modelo de predicción de las salidas desacopladas del sistema mod_pred_d
Cargar modelo de clasificación mod_class
while (1) **do**
  Subscribirse al tópico l_1,l_2,l_3
  Subscribirse al tópico u_1,u_2
  Subscribirse al tópico set_points
  $t \leftarrow kT_s$
  if $t > T_s$ **then**
    Estimar estados $\hat{x}_1, \hat{x}_2, \hat{x}_3$ con mod_pred
    Estimar estados desacoplados $\hat{x}_{1d}, \hat{x}_{2d}$ con mod_pred_d
    Generar los residuales correspondientes res, res_1, res_2
    Generar la pertenencia de la clase con mod_class y guardar la clase $Clase$
    $a_{s1-DoS} \leftarrow 0$
    $a_{s2-DoS} \leftarrow 0$
    $a_{s1-int} \leftarrow 0$
    $a_{s2-int} \leftarrow 0$
    if $Clase == 1 \parallel Clase == 5 \parallel Clase == 7$ **then**
      $a_{s1-int} \leftarrow 1$
    end if
    if $Clase == 2 \parallel Clase == 6 \parallel Clase == 7$ **then**
      $a_{s2-int} \leftarrow 1$
    end if
    if $Clase == 3 \parallel Clase == 6 \parallel Clase == 8$ **then**
      $a_{s1-DoS} \leftarrow 1$
    end if
    if $Clase == 4 \parallel Clase == 5 \parallel Clase == 8$ **then**
      $a_{s2-DoS} \leftarrow 1$
    end if
  end if
  $k \leftarrow k + 1$
  Esperar al siguiente tiempo de muestreo $(t \leftarrow kT_s)$
end while
return publicar Alarmas: $a_{s1-DoS}, a_{s2-DoS}, a_{s1-int}, a_{s2-int}$

Algorithm 5: Sistema de monitoreo

Input: Salidas del proceso, acciones de control,alarmas:

l_1,l_2,l_3, u_1,u_2, a_{s1-DoS}, a_{s2-DoS}, a_{s1-int},a_{s2-int}

Output: Gráfica

Inicialización : Condiciones iniciales

$k \leftarrow 0$

while (1) **do**

  Definir referencias q_1, q_2

  Subscribirse al tópico l_1,l_2,l_3

  Subscribirse al tópico u_1,u_2

  Subscribirse al tópico $alarmas$

  $t \leftarrow kT_s$

  Graficar las variables de interés con respecto al tiempo

  $k \leftarrow k + 1$

  Esperar al siguiente evento

end while

return publicar set_points, q_1, q_2

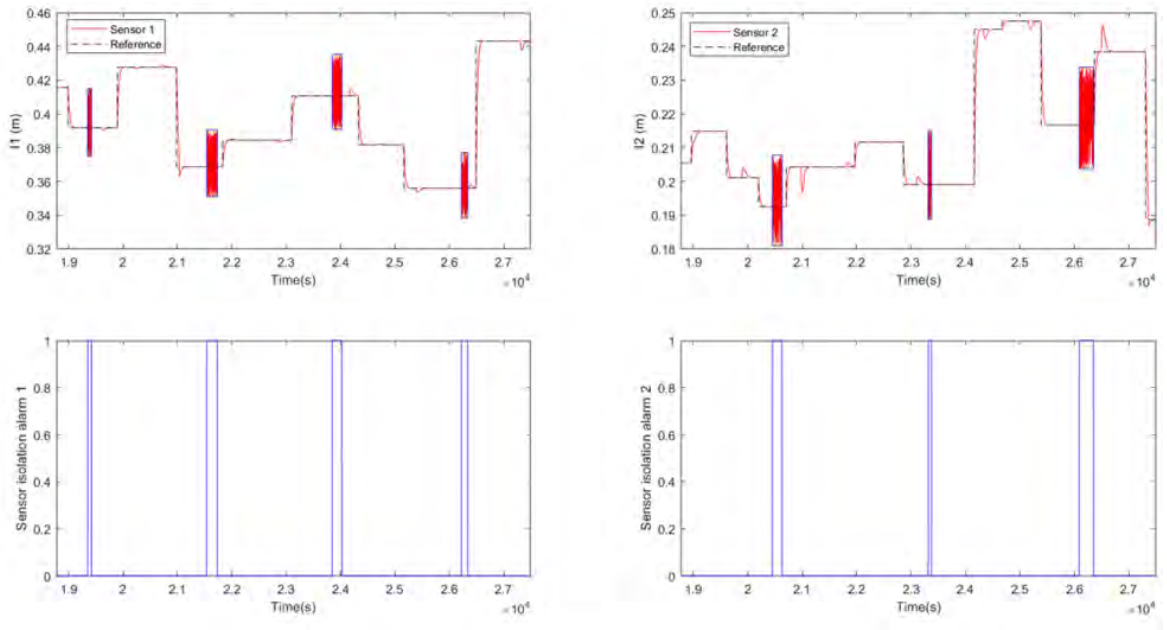


Fig. 61. Sistema de monitoreo del sistema de tanques interconectados.

TABLA XIX.
Comparación con implementaciones tradicionales.

Requerimientos	Propuesta	Implementaciones tradicionales	Comentarios
Promover la escalabilidad	Cumple	Parcial	Las implementaciones tradicionales usualmente requieren de hardware adicional
Aislamiento temporal y espacial	Cumple	No cumple	Las implementaciones tradicionales son monolíticas
Nuevas estrategias	Cumple	Parcial	Las implementaciones tradicionales tienen sus propios drivers
Lenguajes de programación	Cumple	Cumple	
Modularidad	Cumple	Parcial	Las implementaciones tradicionales son monolíticas

La propuesta desarrollada se orienta a la detección de ataques en las variables asociadas a las aplicaciones de control. Se supone que todos los nodos se encuentran conectados a la misma red. La anterior suposición se hace con base en que los ataques requieren que el atacante haya accedido a un equipo dentro de la red del sistema de control industrial. Por lo que la seguridad en los otros niveles de la red se propone que sea abordada bajo las estrategias comentadas en el capítulo 3. Por otro lado, y siguiendo el planteamiento para el cual se realiza la aportación, en el que todos los nodos se encuentren conectados a la misma red de control, en el envío de mensajes la distancia entre nodos está acotada a un único salto.

Para realizar los análisis de planificabilidad locales es necesario contar con plazos individuales de las tareas de Medición, Control y Actuación. Por tal razón algoritmos como el Deadlinemin [194], permiten encontrar límites inferiores para los plazos individuales de estas tareas en cada nodo, mientras que los límites superiores se establecen en función del cumplimiento del D_{CG}.

Aplicando el método presentado en la sección 5.3.3, para este caso el periodo de muestreo del sistema es de $1s$ y el D_{CG} para un rendimiento adecuado será de $1s$. Las tareas de Medición, Control y Actuación, presentan un tiempo de ejecución de $WCET_M = 1ms$, $WCET_C = 2{,}2ms$ y $WCET_A = 0{,}2ms$, respectivamente. Se establece como plazo de partida para la ejecución del algoritmo 6 $D_M = D_C = D_A = 1000ms$. Con el fin de aumentar la carga en los procesadores se generan dos tareas periódicas en cada nodo, de-

Algorithm 6: Algoritmo Deadlinemin

Input: Γ, τ_i

Output: D_i^{min}

Inicialización :

1: $\mathcal{R} :=$ Calcular ICI(τ)
2: $deadline := C_i$
3: $k := \left\lceil \frac{\mathcal{R}}{T_i} \right\rceil$
4: $D_i^{min} := 0$
5: **for** s in $0 : k-1$ **do**
6: $t := min\{\ sT_i + D_i, \mathcal{R}\}$
7: $deadline = C_i$
8: **while** $t > sT_i + C_i$ **do**
9: **if** $t - H_\tau(t) < C_i$ **then**
10: $deadline := H_\tau(t) + C_i - sT_i$
11: break
12: **end if**
13: $t := t - 1$
14: **end while**
15: $D_i^{min} := max\{\ D_i^{min}, deadline\}$
16: **end for**
17: **return** D_i^{min}

notadas por τ_1 y τ_2, cuyos parámetros temporales son $WCET_i = 2ms, D_i = T_i = 10ms$. Los datos presentados en la Tabla XX permiten observar el cumplimiento de planificabilidad local y extremo-extremo de la aplicación.

TABLA XX.
Resultados del análisis de planificabilidad $T_s = 1seg$.

Nodo	Planificabilidad local	$\mathbf{D_{mín}}$	$\mathbf{D_{máx}}$
Medición	Si	$1ms$	$5ms$
Control	Si	$2{,}2ms$	$6{,}2ms$
Actuación	Si	$0{,}2ms$	$4{,}2ms$

Para realizar una comparación entre las aplicaciones que se ejecutan sobre Singularity y RTLinux, se ejecutó la aplicación 3000 veces para realizar un test de latencias y ver los tiempos de ejecución de las aplicaciones. En las Fig. 62, 63 y en la Tabla XXI se puede observar el compilado de los tiempos de ejecución de las aplicaciones concernientes a

la medición, a los algoritmos de control y al proceso de actuación, tanto de las aplicaciones soportados sobre el contenedor Singularity y RTLinux. Se puede observar que las aplicaciones cumplen con los requisitos temporales que se definieron para el correcto funcionamiento del sistema de control en el proceso de los tanques.

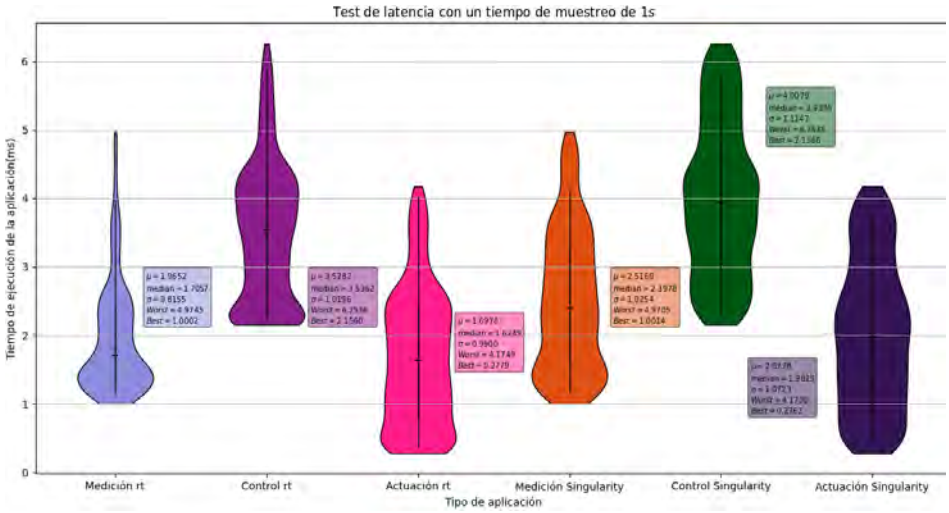


Fig. 62. Latencias de las aplicaciones con $T_s = 1s$.

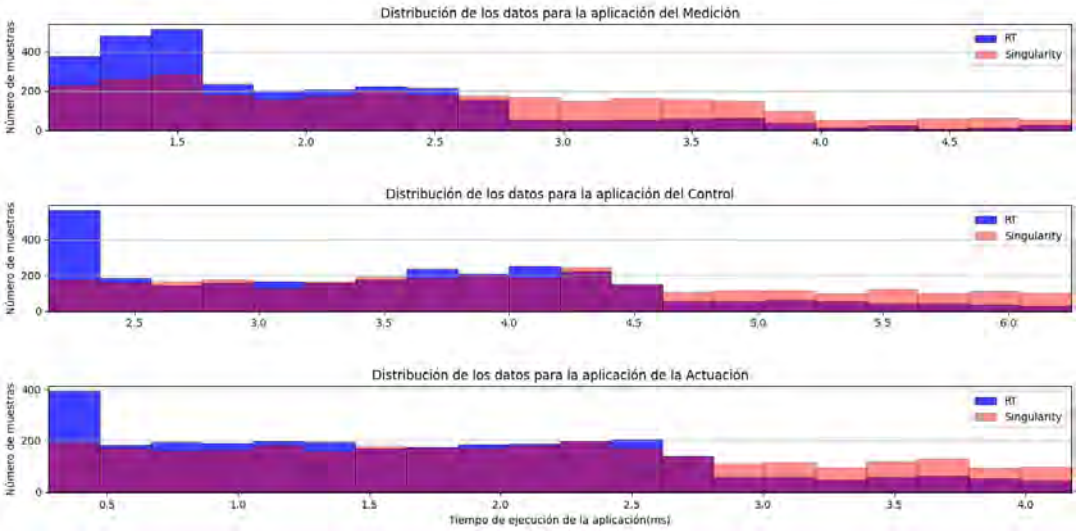


Fig. 63. Histograma del test de latencia con $T_s = 1s$.

Para evaluar el desempeño de las dos arquitecturas para cargas más exigentes, se modificó la dinámica del proceso de tal forma que se requería disminuir el periodo de muestreo. Para esto se opto por una dinámica que requiere un tiempo de muestreo de $5ms$. Al realizar la comparación con las aplicaciones que se encuentran soportadas sobre un

TABLA XXI.
Resumen de latencias (valores en ms).

Aplicación	μ	Mediana	σ	Máx.	Mín.
Medición RT	1.9652	1.7057	0.8155	4.9745	1
Control RT	3.5282	3.5362	1.0196	6.2536	2.156
Actuación RT	1.6978	1.6249	0.99	4.1749	0.2779
Medición Singularity	2.516	2.3978	1.0254	4.9705	1.0014
Control Singularity	4.0079	3.9395	1.1147	6.2535	2.1566
Actuación Singularity	2.0228	1.9825	1.0723	4.173	0.2762

sistema operativo RT, se obtuvieron los resultados que se muestran en las Fig. 64, 65 y en la Tabla XXII.

De este modo se observa que las aplicaciones en Singularity no cumplen del todo las restricciones temporales afectando el rendimiento del sistema de control.

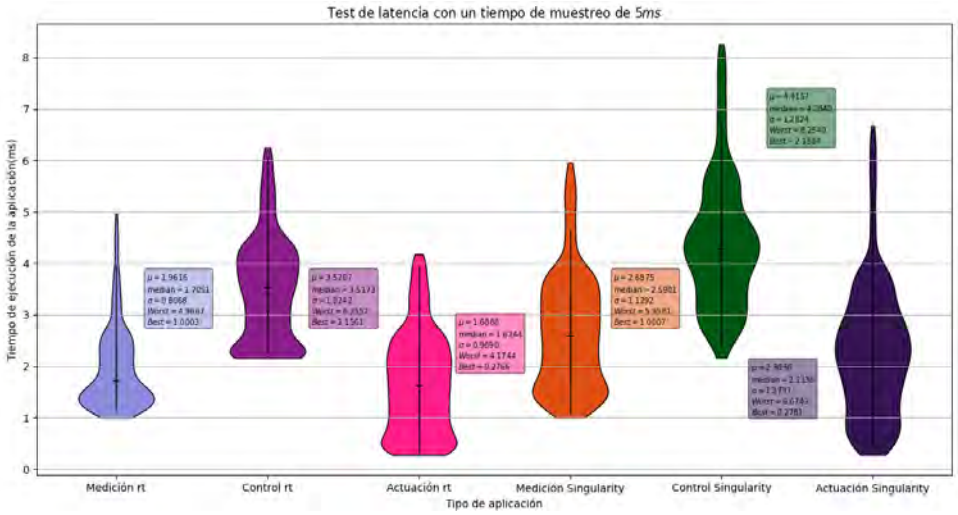


Fig. 64. Latencias de las aplicaciones con $T_s = 5ms$.

El comportamiento dinámico del sistema se observa en la Fig. 66 donde se observa el efecto del no cumplimiento de los plazos temporales requeridos para el funcionamiento de la aplicación, donde se empiezan a presentar oscilaciones que pueden llegar a ser indeseadas.

En este caso se observa que requisitos temporales asociados a tiempos de muestreo por debajo de $5ms$, la aplicación soportada sobre los contenedores no están en la capacidad

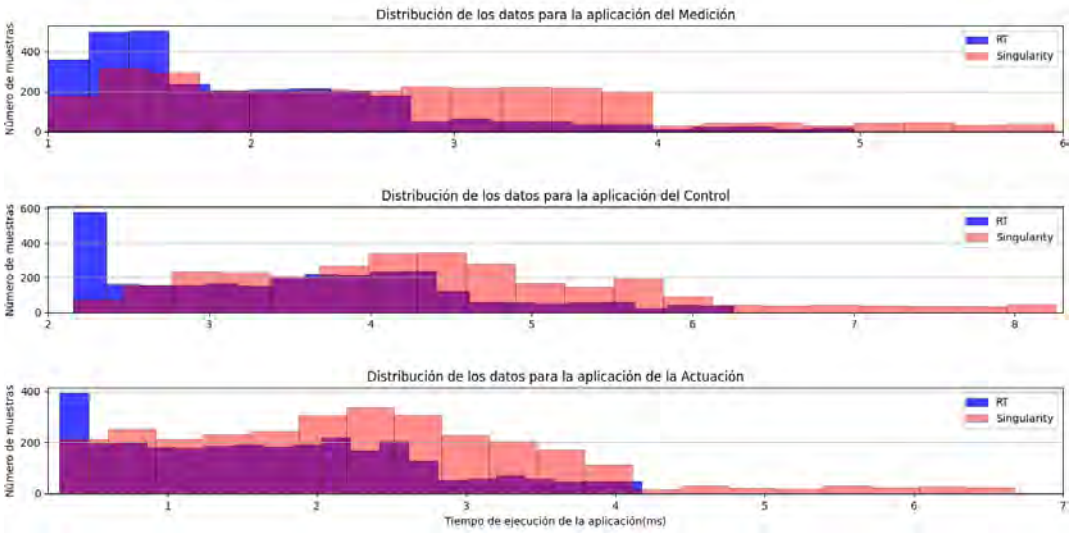


Fig. 65. Histograma del test de latencia con $T_s = 5ms$.

TABLA XXII.
Resumen de latencias (valores en ms).

Aplicación	μ	Mediana	σ	Máx.	Mín.
Medición RT	1.9616	1.7051	0.8068	4.9667	1.0003
Control RT	3.5207	3.5175	1.0242	6.2552	2.1561
Actuación RT	1.688	1.6244	0.989	4.1744	0.2766
Medición Singularity	2.6875	2.5901	1.1292	5.9581	1.0007
Control Singularity	4.4157	4.284	1.2824	8.254	2.1584
Actuación Singularity	2.3036	2.2336	1.2737	6.6742	0.2781

de cumplirlo.

6.2 CASO DE ESTUDIO: PUNTA DELICIA

La empresa Punta Delicia, localizada en Colima, México, es una empresa de producción de diferentes tipo de bebidas. Diferentes procesos se ven involucrados en la planta de producción. Sin embargo, para demostrar el planteamiento y funcionamiento del sistema de detección de ciberataques así como el procedimiento de diseño, este trabajo se enfocó en la etapa donde finalmente la bebida es almacenada. Dentro de los procesos de formulación para una bebida final, el producto debe estabilizarse a un cierto valor de pH, esto se realiza en un tanque de mezcla con capacidad de $2000L$. Los valores de referen-

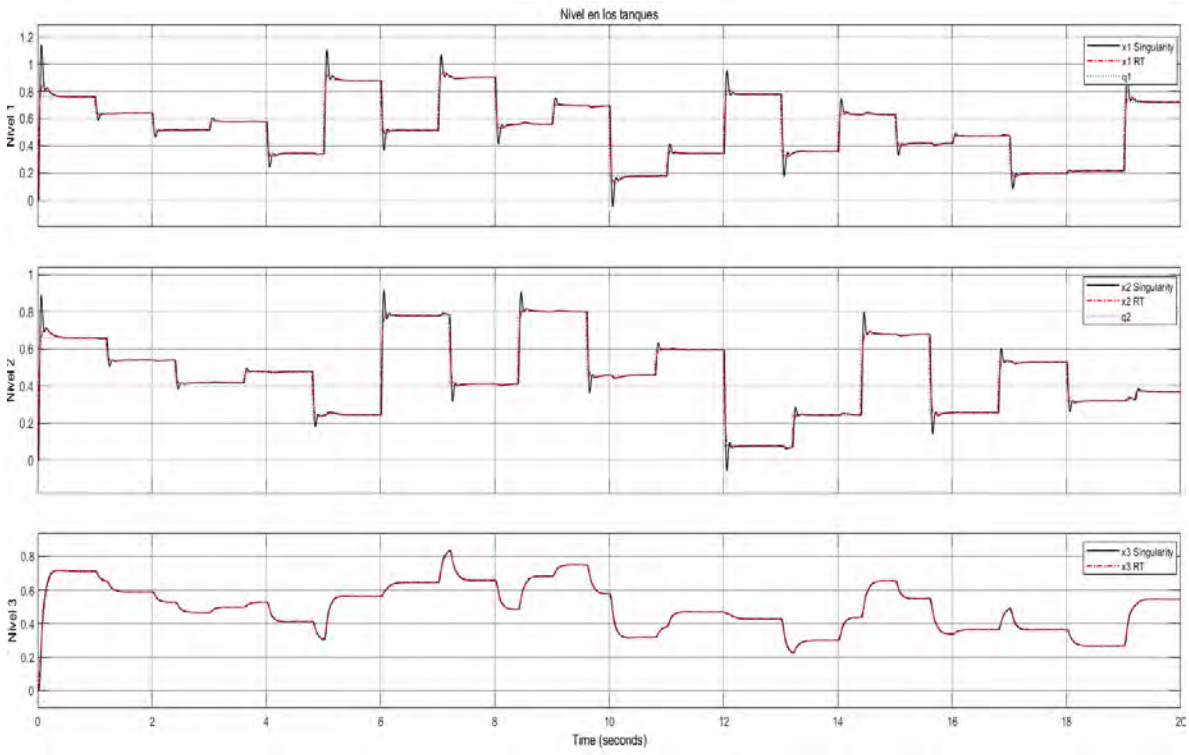


Fig. 66. Respuesta dinámica del sistema con $T_s = 5ms$.

cia de pH pueden oscilar entre 4 y 9 y dependerán de las características deseadas del producto. Desde un punto de vista industrial, estas consideraciones exigen y requieren un controlador y una implementación robusta [195].

6.2.1 Diseño del controlador

Considerar un proceso de neutralización de pH como se muestra en la Fig. 67. El flujo las tasas de corrientes ácidas, compensadoras (buffer), de base y efluentes se indican mediante q_1, q_2, q_3 y q_4, respectivamente. La salida del proceso es el valor de pH de la corriente efluente, y el caudal de la corriente de base y ácida, q_1 y q_3, son las entradas de control. Se deriva un modelo dinámico, Ecuación (67), utilizando las leyes de conservación y el equilibrio de las reacciones. Las hipótesis de modelado incluyen mezcla perfecta, volumen constante del tanque de neutralización (V) y solubilidad completa de los iones implicados [196].

$$\dot{x}_i = \frac{q_1}{V}(w_{1i} - x_i) + \frac{q_2}{V}(w_{2i} - x_i) + \frac{(\alpha_i - x_i)}{V}q_3 \tag{67}$$

Donde w_{1i}, w_{2i} y α_i son las concentraciones de ácido, buffer y base, respectivamente.

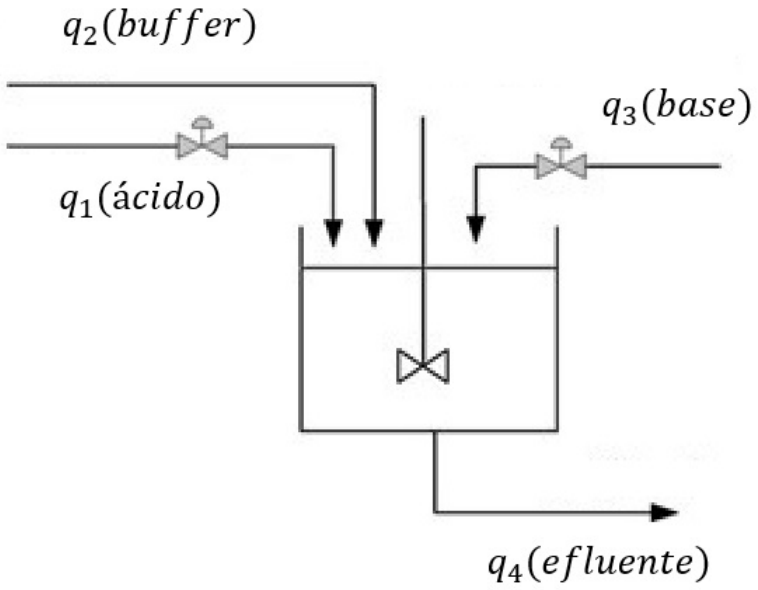


Fig. 67. Proceso de neutralización de pH.

Los estados x_i son las concentraciones de las reacciones invariantes.

El control del pH es fundamental y cada uno de los productos que se procesan en la empresa tienen sus propios requisitos, características y consideraciones. Además, las dinámicas presente en estos procesos contienen no linealidades y su modelado, así como la estimación de parámetros y el control son tareas desafiantes. El algoritmo de control que se está implementado en la empresa de Punta Delicia es un esquema de control basado sobre la estructura de control conocida por el nombre de sincronización Maestro-Esclavo. En este caso el esclavo es el proceso real, mientras que el maestro es generado por una simulación del modelo matemático del proceso en lazo cerrado. De este modo el objetivo es el control de pH con una mínima información del proceso, considerando una referencia variable en el tiempo que varía entre la región básica y la ácida. Este algoritmo fuerza a las mediciones del proceso a seguir una referencia deseada variable en el tiempo, a pesar de las incertidumbres.

Para ello, se utiliza el esquema derivado de [197].

$$\begin{aligned}
\dot{x}_{1m} &= \frac{q_1}{V}(w_{11} - x_{1m}) + \frac{q_2}{V}(w_{21} - x_{1m}) + \frac{(\alpha_1 - x_{1m})}{V}u \\
\dot{x}_{2m} &= \frac{q_1}{V}(w_{12} - x_{2m}) + \frac{q_2}{V}(w_{22} - x_{2m}) + \frac{(\alpha_2 - x_{2m})}{V}u \\
\dot{x}_{3m} &= \frac{q_1}{V}(w_{13} - x_{3m}) + \frac{q_2}{V}(w_{23} - x_{3m}) + \frac{(\alpha_3 - x_{3m})}{V}u
\end{aligned} \tag{68}$$

$$\begin{aligned}
\dot{x}_1 &= \frac{q_2}{V}(w_{21} - x_1) + \frac{(\alpha_1 - x_1)}{V}u_1 + \frac{(w_{11} - x_1)}{V}u_2 \\
\dot{x}_2 &= \frac{q_2}{V}(w_{22} - x_2) + \frac{(\alpha_2 - x_2)}{V}u_1 + \frac{(w_{12} - x_2)}{V}u_2 \\
\dot{x}_3 &= \frac{q_2}{V}(w_{23} - x_3) + \frac{(\alpha_3 - x_3)}{V}u_1 + \frac{(w_{13} - x_3)}{V}u_2
\end{aligned} \tag{69}$$

La Ecuación (68) corresponde al modelo del maestro y la Ecuación (69) corresponde al modelo del esclavo, que es el proceso real. La salida del proceso viene dada por la Ecuación (70), donde $y = pH$.

$$h(x,y) = -x_1 + x_2 - x_3 C_{x_3} + 10^{-y} - 10^{y-pK_w} = 0 \tag{70}$$

Donde C_{x_3} es una función de pH y las constantes de disociación para la i-ésima especie, en este caso para el anión del ácido diprótico débil (H2A) se describe como:

$$C_{x_3} = \frac{2 + 10^{pK_2 - y}}{1 + 10^{pK_2 - y} + 10^{pK_1 + pK_2 - 2y}} \tag{71}$$

En estas ecuaciones, los estados del sistema están dados por $x_{1,2,3,1m,2m,3m}$ los cuáles representan la reacción invariante de i-ésima especie, mol/l, u y u_1 es el caudal de la corriente base, ml/s, u_2 es el caudal de la corriente ácida, ml/s, y y representa el valor de pH. Los parámetros del sistema están definidos en la Tabla XXIII [195].

La estructura del controlador se muestra en la Fig. 68 y esta basado en el trabajo que se presenta en [195].

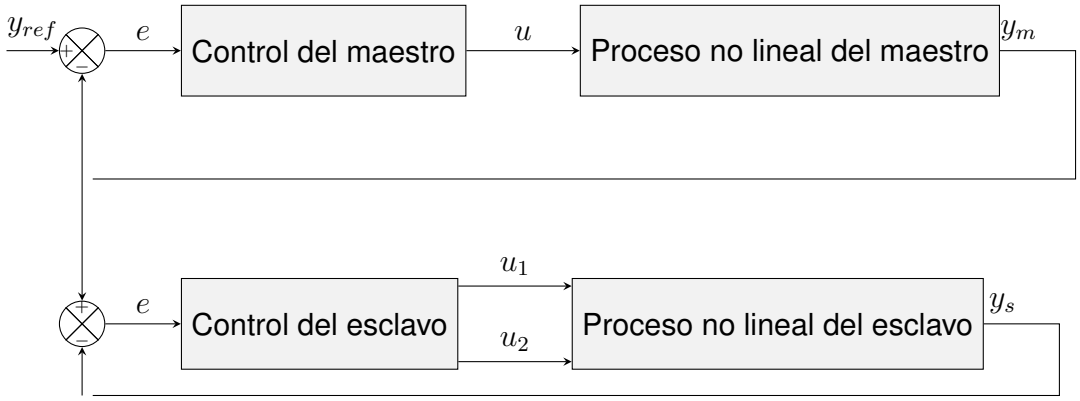


Fig. 68. Estructura de control.

6.2.1.1 Controlador del maestro

En esta sección se describe el controlador implementado para el sistema maestro. Tener en cuenta que el esquema global es un controlador por sincronización Maestro-Esclavo.

TABLA XXIII.
Parámetros del sistema.

Parámetro	Nomenclatura	Valor	Unidades
Concentración de la i-ésima especie de la corriente de base	α_1, α_2, α_3	0 0,00305 0,00005	mol/l
Volumen del tanque de mezclas	V	2900	ml
Constante de disociación del agua	pK_w	16,6	
Caudal de ácido	q_1	16,66	ml/s
Caudal de compensación(buffer)	q_2	0,55	ml/s
Concentración de la i-ésima especie en la corriente ácida	w_{11}, w_{12} , w_{13}	0 0,003 0,6	mol/l
Concentración de la i-ésima especie en la corriente de compensación	w_{21}, w_{22}, w_{23}	0 0,03 0,03	mol/l
Constantes de equilibrio para las reacciones químicas en el sistema	pK_1, pK_2	6,34 10,25	

Para esto se requiere primero controlar el maestro. En este caso se usa linealización por retroalimentación entrada-salida que es un enfoque comúnmente utilizado para controlar sistemas no lineales. El enfoque implica proponer una transformación del sistema no lineal en un sistema lineal equivalente mediante un cambio de variables y una entrada de control adecuada. Esto difiere completamente de la linealización convencional (mediante series de Taylor) en donde la linealización de retroalimentación se logra mediante transformaciones de estado exactas y retroalimentación, en lugar de aproximaciones lineales de la dinámica del sistema.

El objetivo es convertir el sistema no lineal descrito por la Ecuación (72) a uno lineal por transformación de estado y redefiniendo la acción de control. El modelo lineal resultante describe la dinámica del sistema globalmente.

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{72}$$

Se asume que el sistema es linealizable y tiene un grado relativo r (número de veces que se deriva la salida y para que aparezca la señal de entrada u, $r \leq n$, donde n es el número de estados del sistema). Por lo tanto la transformación de la entrada u viene dada por la Ecuación (73).

$$u = \frac{v - L_f^r y}{L_g L_f^{r-1} y} \tag{73}$$

Donde $L_f^r y$ y $L_g L_f^{r-1} y$ son derivadas de Lie. De este modo mediante la geometría diferencial se pueda generar una ecuación diferencial lineal que relacione la salida y con una nueva entrada v,Ecuación (74).

$$y^r = v \tag{74}$$

Donde y^r es la derivada de orden r aplicada a y [198, 199].

Para el controlador del maestro, a partir del modelo planteado en las Ecuación (68) se obtienen las siguientes funciones:

$$f(x) = \begin{bmatrix} \frac{q_1}{V}(w_{11} - x_{1m}) + \frac{q_2}{V}(w_{21} - x_{1m}) \\ \frac{q_1}{V}(w_{12} - x_{2m}) + \frac{q_2}{V}(w_{22} - x_{2m}) \\ \frac{q_1}{V}(w_{13} - x_{3m}) + \frac{q_3}{V}(w_{23} - x_{3m}) \end{bmatrix} \tag{75}$$

$$g(x) = \begin{bmatrix} \frac{\alpha_1 - x_{1m}}{V} \\ \frac{\alpha_2 - x_{2m}}{V} \\ \frac{\alpha_3 - x_{3m}}{V} \end{bmatrix} \tag{76}$$

Para $h(x)$ se tiene la Ecuación (70), la cuál es una función implícita.

El sistema en este caso tiene un grado relativo $r = 1$, debido que al derivar la salida una vez se obtiene la señal de entrada. Por lo tanto la acción de control será (77).

$$u = \frac{v - L_f y}{L_g y} \tag{77}$$

Así se requieren el siguiente conjunto de derivadas de Lie.

$$L_g y = \frac{\partial y}{\partial x} g(x) = -\frac{h_x}{h_y} g(x) \tag{78}$$

$$L_f y = \frac{\partial y}{\partial x} f(x) = -\frac{h_x}{h_y} f(x) \tag{79}$$

La transformación esta dada por la Ecuación (80).

$$\frac{dy}{dt} = -\frac{h_x}{h_y}\frac{dx}{dt} = v \tag{80}$$

De este modo se puede diseñar un controlador que permita seguir la referencia, usando el modelo descrito por la Ecuación (80). En este caso se diseña un seguidor por asignación de polos por retroalimentación de estados, que garantice la estabilidad del modelo lineal entre la variable y y la acción de control v. Los polos deseados son $p_d = [-4 - 10]$ y las respectivas ganancias de la parte proporcional y la parte integral, son:

$$K = [K_p \,|K_i] = [14 \mid -40] \tag{81}$$

Este controlador es discretizado con un tiempo de muestreo de $0{,}1s$ y las ganancias discretas son:

$$K_d = [K_{pd} \,|K_{id}] = [9{,}6180 \mid -2{,}0840] \tag{82}$$

El resultado de implementar el controlador del maestro, se observa en la Fig. 69. Se puede observar que la salida sigue la referencia.

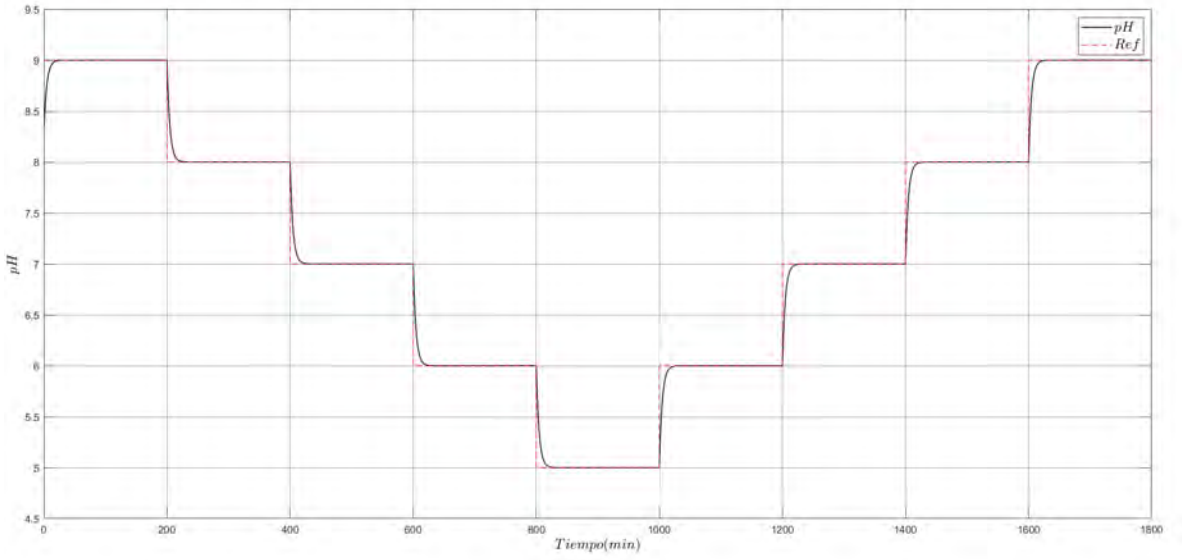


Fig. 69. Respuesta temporal del proceso de pH.

6.2.1.2 Controlador del esclavo-Sincronización maestro-esclavo

El propósito de esta sección es diseñar un controlador que permita lograr la sincronización entre el esclavo y el maestro de modo que se requiere que el error del sistema converja asintóticamente a cero, Ecuación (83).

$$\lim_{t \to \infty} e(t) = 0 \tag{83}$$

A partir del modelo del maestro y del esclavo presentados en las Ecuaciones (68) y (69), el error de cada estado del sistema es definido como sigue:

$$e = \begin{bmatrix} e_{x_1} \\ e_{x_2} \\ e_{x_3} \end{bmatrix} = \begin{bmatrix} x_1 - x_{1_m} \\ x_2 - x_{3_m} \\ x_3 - x_{3_m} \end{bmatrix} \tag{84}$$

Así, la dinámica del error puede ser descrita como:

$$\begin{aligned} \dot{e}_{x_1} &= \dot{x}_1 - \dot{x}_{1_m} \\ \dot{e}_{x_2} &= \dot{x}_2 - \dot{x}_{3_m} \\ \dot{e}_{x_3} &= \dot{x}_3 - \dot{x}_{3_m} \end{aligned} \tag{85}$$

Las expresiones dadas en la Ecuación (85) se pueden reescribir como:

$$\dot{e} = Be + F(x,x_m) + u(t) \tag{86}$$

Donde B son las partes comunes de las matrices del sistema maestro-esclavo, $F(x,x_m)$ contiene las funciones no lineales y los términos no comunes y $u(t)$ es la entrada de control. Con una apropiada acción de control $u(t)$ se puede obtener una señal de error que converja a cero, y así lograr la sincronización entre los dos sistemas.

Así, el problema de la sincronización es el diseño del controlador apropiado que elimine las partes no lineales y no comunes y tenga otras partes que permitan lograr la estabilidad del sistema, de tal forma que:

$$u(t) = -F(x,x_m) + v(t) \tag{87}$$

Donde $v(t) = -Ke(t)$ es un controlador lineal y k es la matriz de ganancia de retroalimentación. Así la Ecuación (86) puede ser reescrita como sigue:

$$\dot{e} = Be + v(t) = Be(t) - Ke(t) = (B - K)e(t) = Me(t) \tag{88}$$

Donde $M = B - K$. De acuerdo a la Ecuación (88), la cual es una ecuación diferencial lineal de primer orden, si la matriz M del sistema es Hurwitz, es decir que todos sus autovalores sean negativos, de acuerdo a la teoría de control lineal, el error del sistema será asintóticamente estable, lo cual permite obtener una sincronización asintóticamente global de los sistemas [200, 201].

Para lograr la sincronización maestro-esclavo en este caso las ecuaciones de los errores dinámicos están dados por las expresiones:

$$\begin{aligned} \dot{e}_{x_1} &= \frac{q_2}{V}(w_{21} - x_1) + \frac{(\alpha_1 - x_1)}{V}u_1 + \frac{(w_{11} - x_1)}{V}u_2 - \left(\frac{q_1}{V}(w_{11} - x_{1m}) + \frac{q_2}{V}(w_{21} - x_{1m}) + \frac{(\alpha_1 - x_{1m})}{V}u\right) \\ \dot{e}_{x_2} &= \frac{q_2}{V}(w_{22} - x_2) + \frac{(\alpha_2 - x_2)}{V}u_1 + \frac{(w_{12} - x_2)}{V}u_2 - \left(\frac{q_1}{V}(w_{12} - x_{2m}) + \frac{q_2}{V}(w_{22} - x_{2m}) + \frac{(\alpha_2 - x_{2m})}{V}u\right) \\ \dot{e}_{x_3} &= \frac{q_2}{V}(w_{23} - x_3) + \frac{(\alpha_3 - x_3)}{V}u_1 + \frac{(w_{13} - x_3)}{V}u_2 - \left(\frac{q_1}{V}(w_{13} - x_{3m}) + \frac{q_3}{V}(w_{23} - x_{3m}) + \frac{(\alpha_3 - x_{3m})}{V}u\right) \end{aligned} \tag{89}$$

Cancelando los términos semejantes y reescribiendo los errores dinámicos de la forma (88), se obtiene las siguientes relaciones:

$$\begin{aligned}\dot{e}_{x_1} &= -\frac{q_2}{V}e_{x_1} + v_1 \\ \dot{e}_{x_2} &= -\frac{q_2}{V}e_{x_2} + v_2 \\ \dot{e}_{x_3} &= -\frac{q_2}{V}e_{x_3} + v_3\end{aligned} \tag{90}$$

Donde:

$$\begin{aligned}v_1 &= \frac{(w_{11}-x_1)}{V}u_2 + \frac{(\alpha_1-x_1)}{V}u_1 - \frac{(\alpha_1-x_{1m})}{V}u - \frac{(w_{11}-x_{1m})}{V}q_1 \\ v_2 &= \frac{(w_{12}-x_2)}{V}u_2 + \frac{(\alpha_2-x_2)}{V}u_1 - \frac{(\alpha_2-x_{2m})}{V}u - \frac{(w_{12}-x_{2m})}{V}q_1 \\ v_3 &= \frac{(w_{13}-x_3)}{V}u_2 + \frac{(\alpha_3-x_3)}{V}u_1 - \frac{(\alpha_3-x_{3m})}{V}u - \frac{(w_{13}-x_{3m})}{V}q_1\end{aligned} \tag{91}$$

De este modo se puede diseñar un controlador de tal forma que:

$$\begin{aligned}v &= F(x,x_m) + u = -Ke \\ v &= -\begin{bmatrix}k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33}\end{bmatrix}\begin{bmatrix}e_{x_1} \\ e_{x_2} \\ e_{x_3}\end{bmatrix}\end{aligned} \tag{92}$$

Al usar la relación $\dot{e} = (B-K)e(t) = Me(t)$ se puede encontrar una matriz K que permita garantizar la estabilidad del sistema seleccionando una matriz M cuyos autovalores sean todos negativos, y así se lograría la sincronización. De este modo:

$$\dot{e} = \begin{bmatrix}-\frac{q_2}{V}-k_{11} & -k_{12} & -k_{13} \\ -k_{21} & -\frac{q_2}{V}-k_{22} & -k_{23} \\ -k_{31} & -k_{32} & -\frac{q_2}{V}-k_{33}\end{bmatrix}\begin{bmatrix}e_{x_1} \\ e_{x_2} \\ e_{x_3}\end{bmatrix} = \begin{bmatrix}-1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1\end{bmatrix}\begin{bmatrix}e_{x_1} \\ e_{x_2} \\ e_{x_3}\end{bmatrix} \tag{93}$$

De este modo solucionando el conjunto de Ecuaciones representadas en (93) se obtiene que $k_{11} = k_{22} = k_{33} = 1 - q_2/V$ y el resto de constantes son cero. Así desde las ecuaciones (87) y (91) se puede definir la ley de control para lograr la sincronización entre el sistema maestro y esclavo. La ley de control es expresada en la Ecuación (94).

$$\begin{bmatrix}u_1 \\ u_2\end{bmatrix} = G^{+}(v - F(x,x_m)) \tag{94}$$

Donde G^+ es la pseudoinversa de la matriz G definida en la Ecuación (95), v es el vector definido en la Ecuación (92) y $F(x,x_m)$ es el vector descrito en la Ecuación (96).

$$G = \begin{bmatrix} \frac{\alpha_1 - x_1}{V} & \frac{w_{11} - x_1}{V} \\ \frac{\alpha_2 - x_2}{V} & \frac{w_{12} - x_2}{V} \\ \frac{\alpha_3 - x_3}{V} & \frac{w_{13} - x_3}{V} \end{bmatrix} \tag{95}$$

$$F(x,x_m) = -\begin{bmatrix} \frac{q_1}{V}(w_{11} - x_{1m}) + \frac{u}{V}(\alpha_1 - x_{1m}) \\ \frac{q_1}{V}(w_{12} - x_{2m}) + \frac{u}{V}(\alpha_2 - x_{2m}) \\ \frac{q_1}{V}(w_{13} - x_{3m}) + \frac{u}{V}(\alpha_3 - x_{3m}) \end{bmatrix} \tag{96}$$

El resultado de implementar la sincronización maestro-esclavo, se observa en la Figura 70.

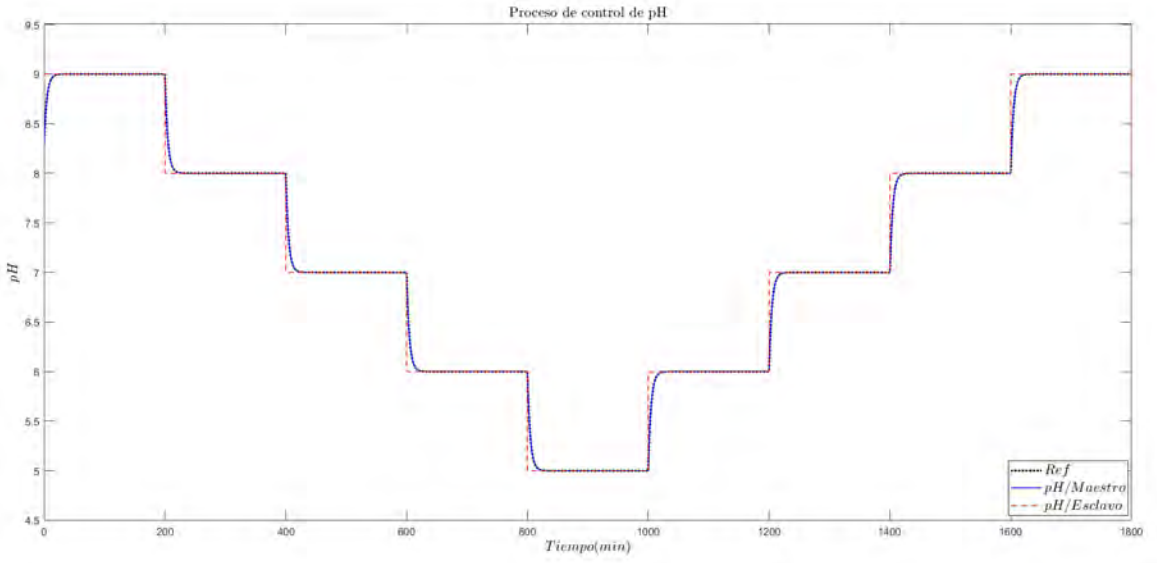


Fig. 70. Respuesta temporal del proceso de pH mediante la sincronización maestro-esclavo.

Los cambios de referencia se realizan de manera aleatoria, bajo una distribución uniforme en el rango de pH mencionado anteriormente. En está gráfica se puede observar que el sistema esclavo logra sincronizarse con el maestro permitiendo seguir los cambios de referencia.

6.2.2 Sistema de detección de ciberataques en el proceso de control de pH

La generación de ataques se realizó en la salida del proceso real, que es la medida que usa el controlador esclavo para ejecutar su algoritmo. Para la generación de estos ata-

ques se plantean tres casos: operación normal (Class 0), ataques de integridad (Class 1) y ataques de Denegación de Servicio (Class 2). La distribución de los datos generados se muestra en la Figura 71.

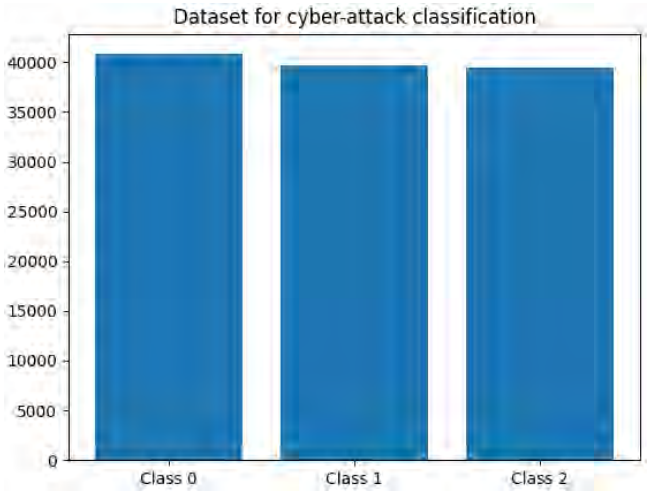


Fig. 71. Distribución de datos.

Cada uno de estos casos se generó de forma aleatoria, garantizando un balance entre las diversas clases. La Figura 72 muestra el comportamiento del proceso bajo ataque, se puede observar que el proceso real no es capaz de seguir la referencia cuando ocurren los ciberataques, esto es debido a que el controlador no se encuentra en la capacidad de abordar estas situaciones.

Para el diseño del sistema de detección y aislamiento de ciberataques se requiere de un modelo que permita estimar la salida del proceso $\hat{y}_k$ y de esta manera realizar una comparación con la salida del proceso real y_k, generando una señal residual $res_k = |\hat{y}_k - y_k|$ para que el clasificador detecte las situaciones planteadas. El modelo de la arquitectura para la detección y el aislamiento es la misma que se presentó en el capítulo anterior.

El modelo que estima la salida del proceso se encarga de realizar una regresión usando una arquitectura basada en redes neuronales convolucionales 1 dimensional. Está arquitectura está compuesta por el siguiente conjunto de capas: la primera capa es una capa convolucional que tiene 8 filtros con un tamaño de 9, seguida de una capa de batch normalization y una función de activación Leaky Relu. Se repite está misma estructura, pero variando la cantidad de filtros en la capa convolucional, en este caso 16 con el mismo tamaño del anterior. La estructura sigue con una capa Dropout o de inactivación (0.15) para evitar el sobre entrenamiento. Seguido a esto se agrega una capa flatten para

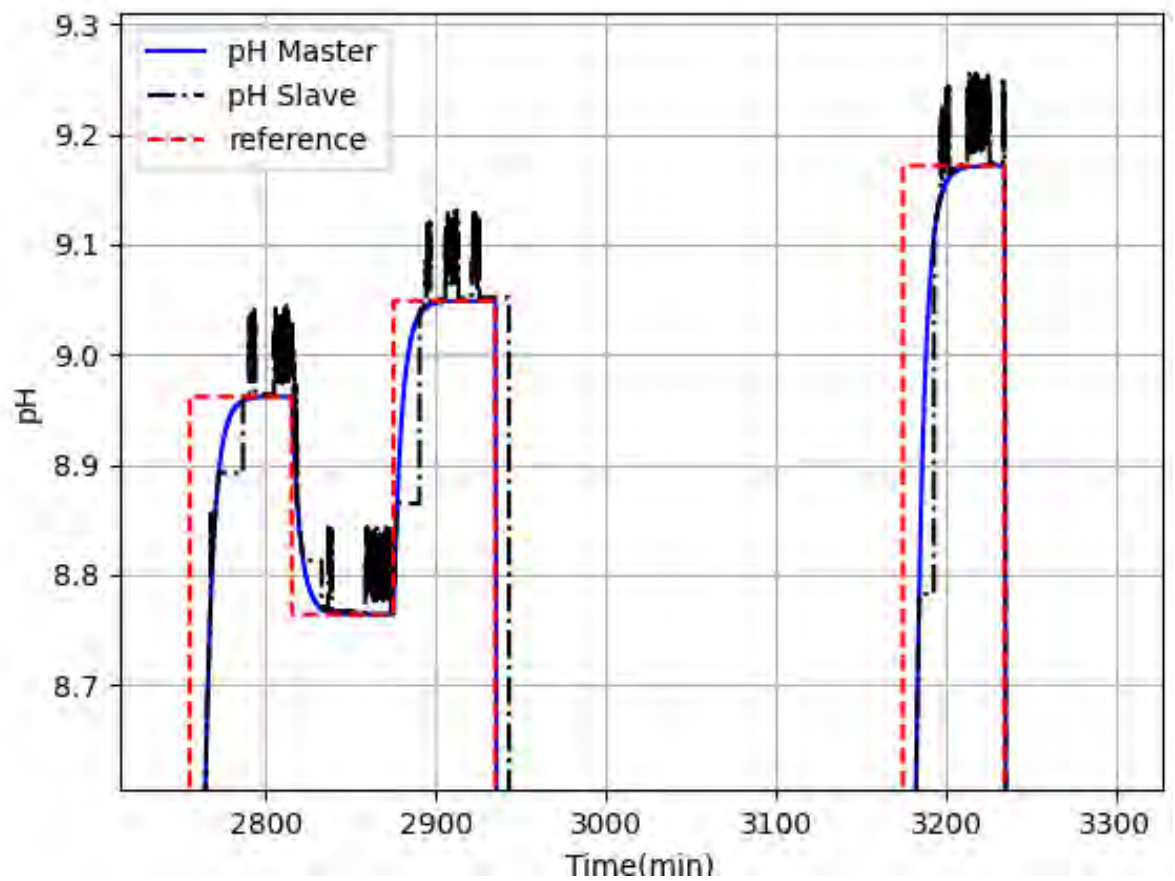


Fig. 72. Proceso de control de pH bajo ataque.

posteriormente agregar una capa totalmente conectada que permite estimar la salida.

La entrada de esta red para estimar la salida está compuesta por el siguiente vector de características (97).

$$\begin{aligned} input = & [y_s(k-3)\ y_s(k-2)\ y_s(k-1) \\ & u_1(k-3)\ u_1(k-2)\ u_1(k-1) \\ & u_2(k-3)\ u_2(k-2)\ u_2(k-1)]^T \end{aligned} \tag{97}$$

A partir de esto se obtiene un modelo de predicción que permite estimar $\hat{y}_k$. La variable y_s representa la salida del proceso y las señales u_1 y u_2 representan las acciones de control generadas desde el controlador esclavo.

Posteriormente se entrena el modelo de clasificación que permitirá distinguir las tres clases presentadas en la Fig. 71. Este modelo presenta una arquitectura basada en redes convolucionales, capas de normalización, capas de agrupación por promedio y redes totalmente conectadas. La entrada de esta red está compuesta por el siguiente

vector de características (98):

$$input = [y_s(k)\ y_m(k)\ \hat{y}(k)\ y_{ref}(k)\ res_s(k)\ res_m(k)]^T \tag{98}$$

Donde $y_m(k),\ \hat{y}(k),\ y_{ref}(k),\ res_s(k),\ res_m(k)$ son la salida del proceso que modela el maestro, la salida estimada por el modelo de predicción, la referencia, el residual generado desde el esclavo y el residual generado desde el maestro, respectivamente. A partir de estas características el modelo de clasificación permite obtener una probabilidad de pertenencia asociada a cada una de las clases previamente definidas y de está manera detectar si existe una anomalía relacionada con ciberataques determinando a su vez el tipo de ciberataque se está llevando a cabo.

A partir de los datos de entrenamiento, el rendimiento del clasificador bajo las métricas de precisión, exactitud, recall y F1 score son mostrados en la Tabla XXIV.

TABLA XXIV.
Resumen de métricas.

	Exactitud	**Precisión**	**Recall**	**F1 Score**
Clase 0	0.98	0.97	0.99	0.98
Clase 1	0.98	0.96	0.95	0.96
Clase 2	0.99	0.99	0.96	0.98

Esto permite observar que el clasificador es capaz de reconocer las situaciones normales y anómalas en un gran porcentaje, posibilitando disminuir la tasa de falsos positivos y falsos negativos. Además de esto, en las Fig. 73 y 74.

6.2.3 Microservicios basados en componentes para la implementación de procesos de control de pH

Para aproximar la propuesta a un entorno real, se determina el uso de componentes basados en micro servicios como presentamos en [192] y como se describió en el capítulo anterior, lo cuál permite bajo este enfoque, una fácil implementación e integración del sistema de detección sin alterar los desarrollos anteriores que se tengan dentro del proceso.

Los bioprocesos industriales requieren una modularidad avanzada, flexibilidad y escalabilidad de producción y debe ser capaz de mantener la fiabilidad de muchos elementos interconectados y dispositivos. Además de los requisitos impuestos por el enfoque de control de pH, los requisitos de la implementación se describen a continuación:

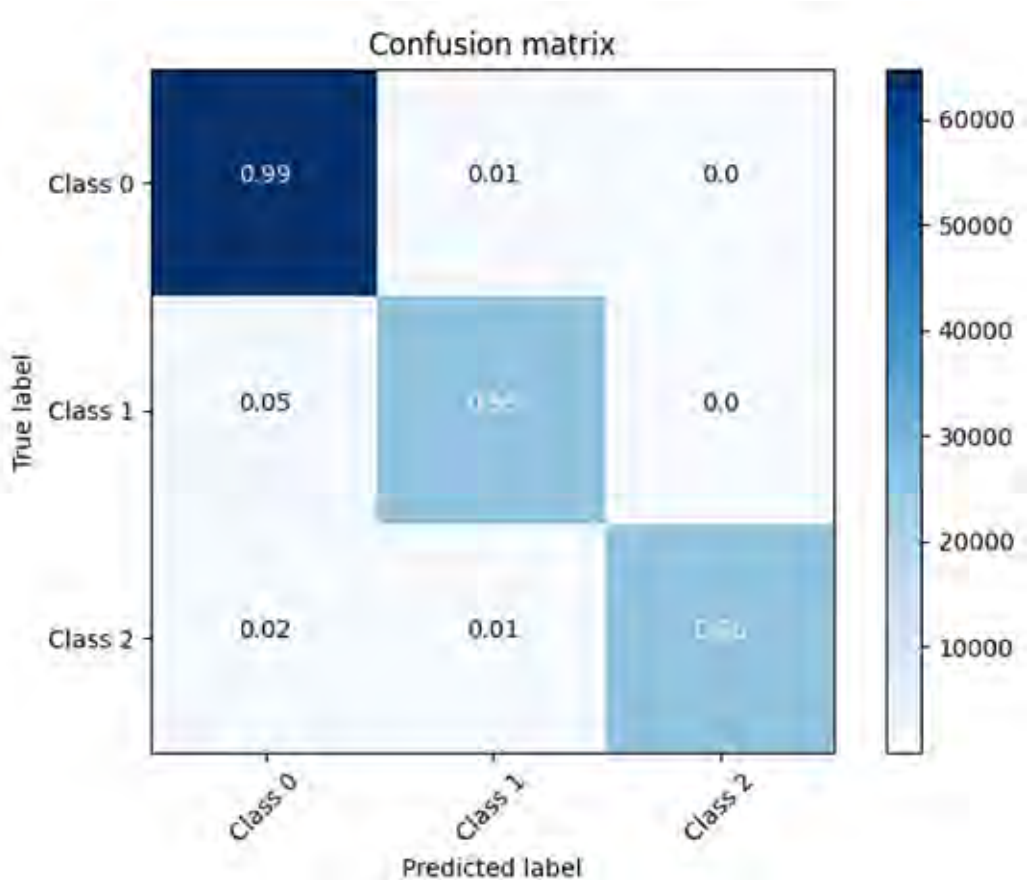


Fig. 73. Matriz de confusión para el sistema de control de pH.

• La arquitectura de los componentes debe diseñarse para promover la escalabilidad.

• Las características deben estar completamente aisladas unas de otras en el tiempo y el espacio.

• La innovación no debe verse restringida con respecto al apoyo a nuevos tipos de insumos, nuevas plataformas de destino, nueva visualización, nuevas estrategias, etc. Además, las funcionalidades deben implementarse en el lenguaje de programación más efectivo.

• La plataforma debe ser lo más modular posible para facilitar las actualizaciones y mejoras de las funcionalidades individuales. Además, agregar nuevas funciones debe ser lo transparente como sea posible para el sistema actualmente en ejecución.

• La ejecución mínima requerida en los procesos más rápidos debe ser de al menos 1 segundo.

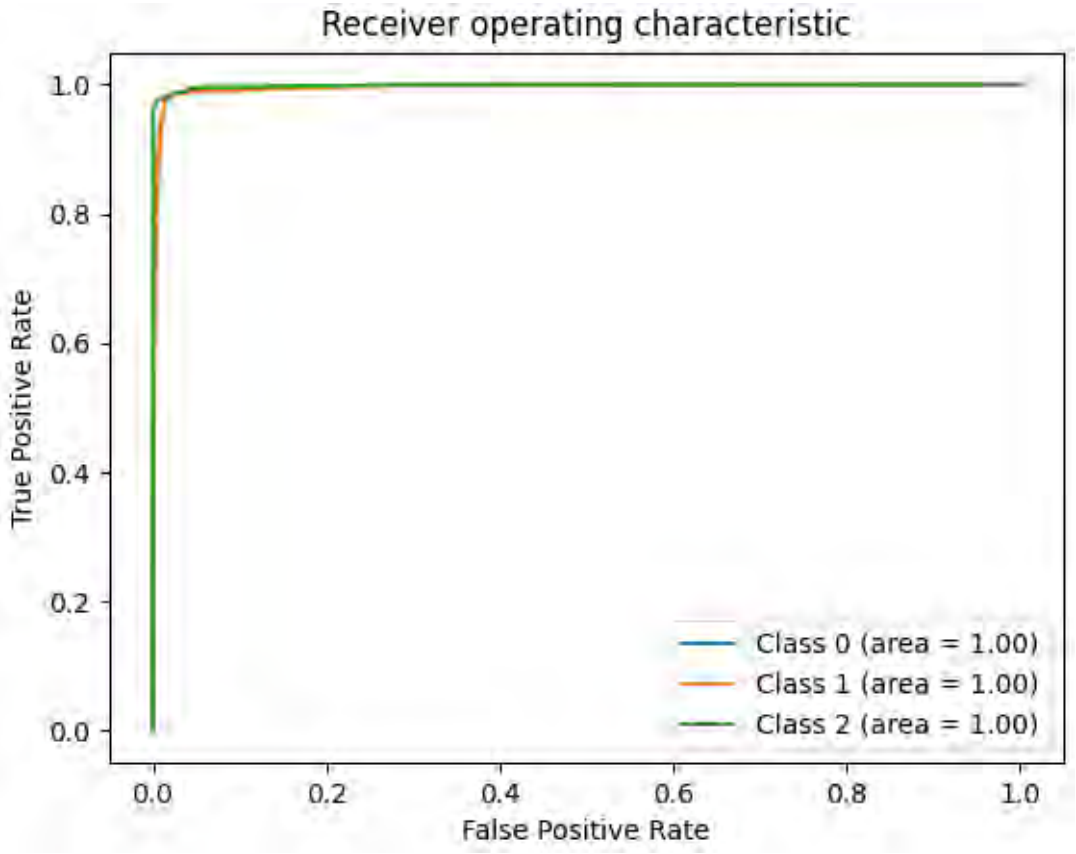


Fig. 74. Curva ROC para el sistema de control de pH.

El cumplimiento de estos requisitos permitirá escalar el proceso de control de pH a nivel industrial. El marco básico presentado en el capítulo anterior, permite guiar el diseño a partir de los componentes definidos. Por lo tanto, hacer una identificación funcional de los componentes y los diferentes servicios ofrecidos y requeridos por cada uno de ellos, con su respectiva funcionalidad, permite tener un modelo de implementación.

La Tabla XXV muestra los diferentes contenedores necesarios para la implementación, los diferentes servicios ofrecidos y requeridos y el tipo de servicio proporcionados por la arquitectura.

La arquitectura del caso de estudio implementado se presenta en la Fig. 75, donde se puede apreciar el ensamblaje de los diferentes componentes, los cuales se integran a través de las diferentes conexiones entre ellos. El bloque de E/S se compone de tres contenedores: Medición, donde se tiene un tópico que publica la lectura de la medida del pH, med_pHs, periódicamente, y dos contenedores más, Válvula 1 y 2, que se subscriben a las acciones de control del esclavo y permite actuar directamente sobre las válvulas peristálticas respectivas. De manera similar, se tiene dos contenedores donde se ubican

TABLA XXV.
Implementación por contenedores, tópicos y servicios.

Contenedor	Tópico	Tipo	Servicio
Medición	med_pHs	Periódico(1s)	Ofrece
Válvula 1	$u1\_pHs$	Eventual	Requiere
Válvula 2	$u2\_pHs$	Eventual	Requiere
Control maestro	set_points	Eventual	Requiere
	c_pHm	Periódico (1s)	Ofrece
Control esclavo	$u1\_pHs$	Periódico (1s)	Ofrece
	$u2\_pHs$	Periódico (1s)	Ofrece
	med_pHs	Eventual	Requiere
	c_pHm	Eventual	Requiere
Sistema de monitoreo	set_points	Eventual	Ofrece
	med_pHs	Eventual	Requiere
	$u1\_pHs$	Eventual	Requiere
	$u2\_pHs$	Eventual	Requiere
	c_pHm	Eventual	Requiere

los controladores del maestro y el esclavo. El contenedor asociado al sistema maestro se subscribe al servicio de set_points, donde se da la referencia a seguir, mientras que por otro lado publica la salida del maestro, c_pHm, de forma periódica. Mientras que el contenedor asociado al esclavo, tiene dos subscripciones de los tópicos med_pHs y c_pHm asociados a los contenedores de Medición y Control maestro anteriormente descritos. Este contenedor ofrece dos servicios periódicos, $u1\_pHs$ y $u2\_pHs$, donde se efectúa el algoritmo de control de sincronización maestro-esclavo. Finalmente se tiene un sistema de monitoreo que permite implementar gráficas de monitoreo para ver el estado de las variables de interés del proceso. Igualmente en este sistema se encuentra embebido el sistema de detección que se presentó en la sección anterior con el fin de localizar posibles eventualidades de ciberataques que estén ocurriendo en el proceso.

De este modo el sistema de detección toma los datos desde los diferentes componentes distribuidos en micro servicios que se tengan implementados y que vienes desde la estructura que se mostró en la Fig. 68. A partir de estos datos el sistema realiza los pasos previamente mencionados, permitiendo generar alarmas cuando exista la ocurrencia de ciberataques en el proceso y teniendo garantía del cumplimiento de los plazos tempora-

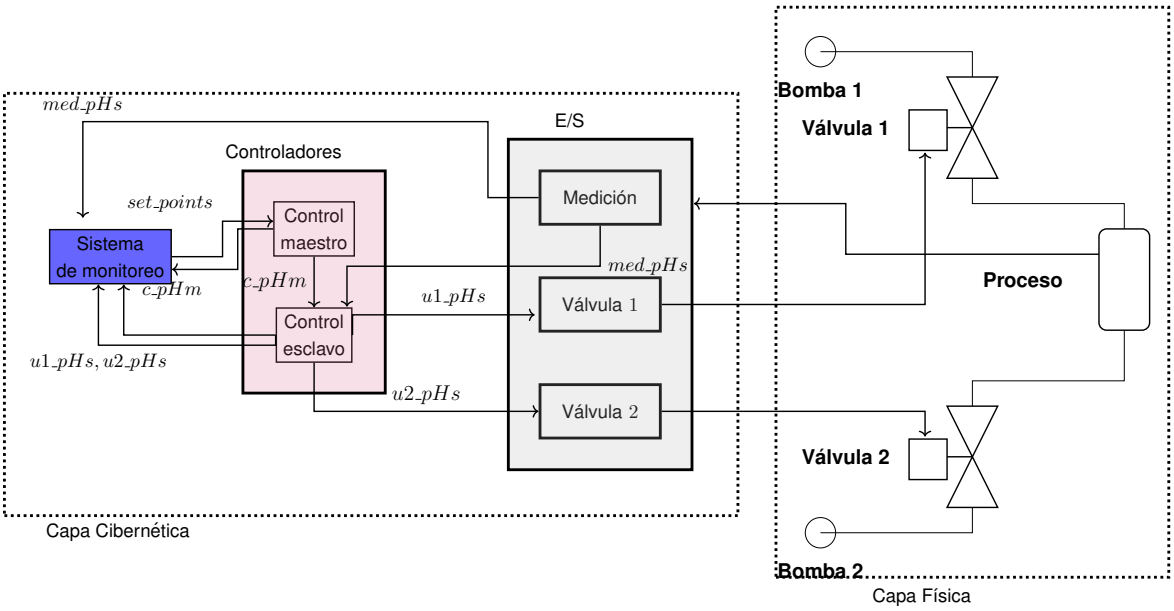


Fig. 75. Diseño de arquitectura DDS basada en componentes para el proceso de control de pH.

les.

En particular, para la implementación del control de pH y todas las funcionalidades especificadas en la Tabla XXV, contenedores Singularity se han utilizado para esta implementación y aunque se pueden implementar otros contenedores, desde el punto de vista del tamaño y la facilidad de la migración, se ha optado por Singularity en lugar de otros.

6.2.4 Resultados y discusión

En esta sección se presentan la verificación del caso de estudio desarrollado con el marco propuesto, se realiza una comparación cualitativa con implementaciones tradicionales y finalmente se observa un test de latencia de las aplicaciones para verificar el cumplimiento de los plazos temporales. Vale la pena mencionar que el enfoque en este estudio de caso no fue en la robustez y optimización del algoritmo de control, sino la funcionalidad de la arquitectura así como el cumplimiento de los requisitos de diseño e implementación.

Como puede verse en los resultados de este estudio de caso, la arquitectura desarrollada logra proporcionar una integración consistente entre una infraestructura de hardware/software y un proceso de control a escala industrial. A través de esta integración, la lógica de los algoritmos de control y del sistema de detección fue implementada en

Python, debido a la facilidad del manejo para las expresiones matemáticas y la amplia gama de bibliotecas y librerías que ofrecen para el análisis de ciencia de datos y el desarrollo de máquinas de aprendizaje.

La Fig. 76 muestra el monitoreo de la variable del proceso, así como la evolución en el tiempo de las señales residuales y las alarmas correspondientes al tipo de ataque que se puede estar llevando a cabo. Estos residuales son obtenidos al comparar la salida estimada desde el modelo de predicción con las variables del proceso, tanto del esclavo como del maestro y a partir de ellas y de las otras características nombradas en la Ecuación 98 determinar la presencia de un ciberataque a partir de unas señales de alarma.

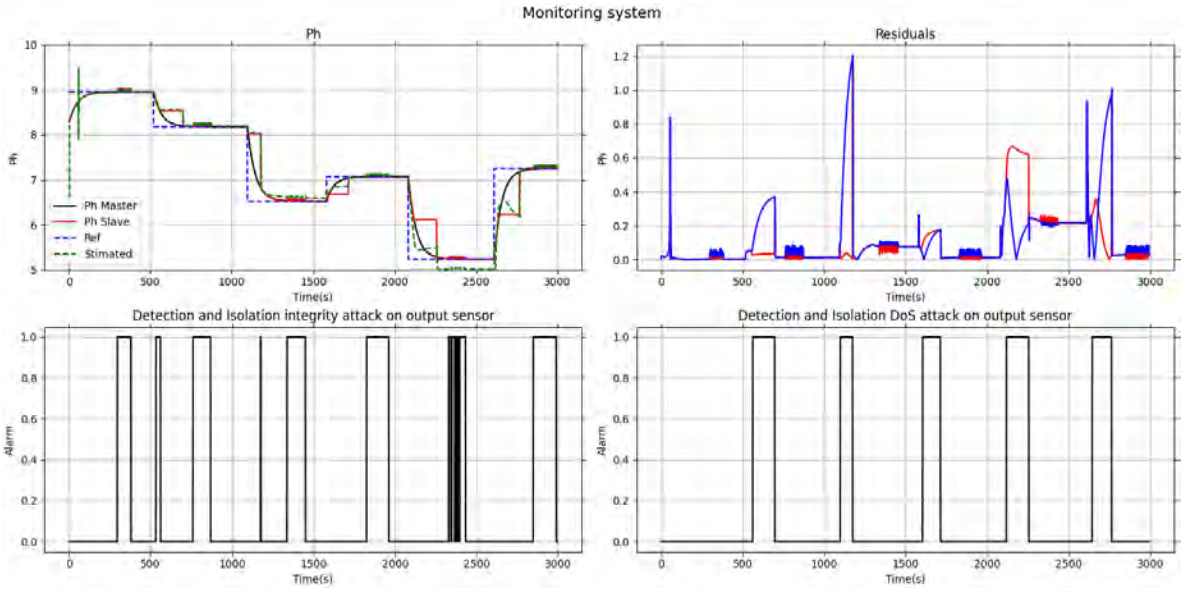


Fig. 76. Monitoreo del proceso y alarmas generados desde el sistema de detección.

En la misma Fig. 76 se muestra la generación de las alarmas cuando ocurre un ciberataque. En este caso se observa que hay intervalos de tiempos donde se efectúan ataques tanto de integridad y DoS que genera que el sistema esclavo no alcance la referencia de pH deseada. En este tiempo el sistema detecta la ocurrencia de este fenómeno poniendo en 1 la alarma relacionada con la respectiva situación.

Estos resultados permiten validar la propuesta en el ámbito mencionado permitiendo tener un monitoreo en linea de la ocurrencia de estos ciberataques para posteriormente tomar las acciones requeridas.

Adicionalmente se presenta una comparación cualitativa con implementaciones tradicionales, basadas sobre los requisitos de la implementación del control de pH que indican

el grado de flexibilidad y escalabilidad del marco. Los siguientes requisitos se utilizan para esta comparación:

1. La arquitectura de los componentes debe diseñarse para promover la escalabilidad (escalamiento horizontal).
2. Los microservicios y las aplicaciones deben estar completamente aislados temporal y espacialmente entre sí.
3. La implementación no debe limitarse al soporte de nuevos tipos de entrada, nuevas plataformas de destino, nuevas interfaces de visualización, nuevas estrategias, etc. Además, las funcionalidades deben implementarse en el lenguaje de programación más eficiente.
4. El diseño debe ser lo más modular posible para facilitar las actualizaciones y mejoras de las funcionalidades individuales. Además, agregar nuevas funciones debe ser lo más transparente posible para la ejecución actual sistema.
5. La ejecución mínima requerida en los procesos más rápidos debe ser de al menos 1 segundo de rendimiento.

La Tabla XXVI muestra los resultados de la comparación cualitativa entre la propuesta y otras implementaciones tradicionales, basado en los requerimientos del proceso de control de pH, este trabajo indica el grado de flexibilidad y escalabilidad de la arquitectura.

TABLA XXVI.
Comparación con implementaciones tradicionales.

Requerimientos	Propuesta	Implementaciones tradicionales	Comentarios
1	Cumple	Parcial	Las implementaciones tradicionales usualmente requieren de hardware adicional
2	Cumple	No cumple	Las implementaciones tradicionales son monolíticas
3	Cumple	Parcial	Las implementaciones tradicionales tienen sus propios drivers
4	Cumple	No cumple	Las implementaciones tradicionales son privadas
5	Cumple	Cumple	

Como se puede ver, las implementaciones tradicionales no cumplen con todos los requerimientos. Además, los componentes basados en los microservicios, permiten operaciones de reconfiguración sin necesidad de detener las operaciones actuales y sin tener que volver a programar funciones adicionales. Con componentes plug-and-play, se pueden incorporar muchas características nuevas. Para las pruebas realizadas, el servicio de control es plenamente factible, sin alterar desarrollos anteriores o los que están funcionando actualmente. Finalmente, los resultados experimentales corresponden a los resultados formales obtenidos del modelo de simulación, lo que indica que la plataforma implementada cumplió con los requisitos del sistema de control y modularidad avanzada, flexibilidad y escalabilidad de la producción.

Finalmente se desarrolla un test de latencias para verificar que los tiempos de ejecución de las diferentes aplicaciones que se tienen en los contenedores previamente definidos, estén por debajo de los tiempos de muestreo de las aplicaciones. En las Fig. 77, 78 y en la Tabla XXVII se puede observar el compilado de los tiempos de ejecución de las aplicaciones concernientes a la medición, a los algoritmos de control y al sistema de detección.

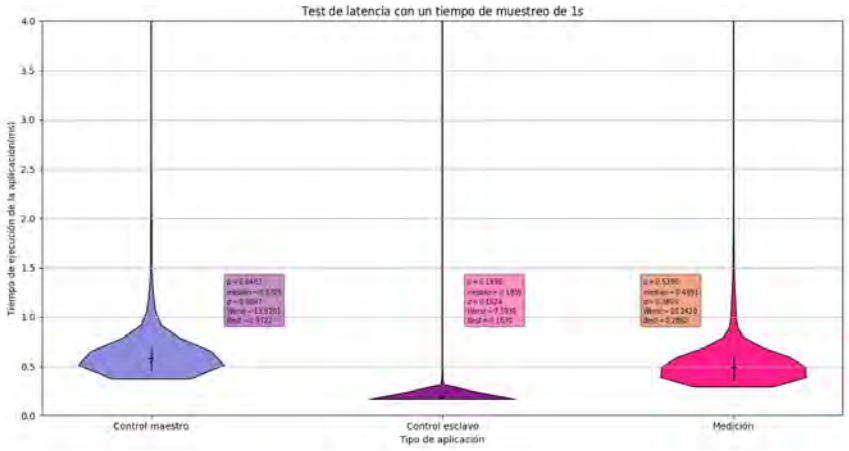


Fig. 77. Latencias de las aplicaciones que se ejecutan en los contenedores.

En la Fig. 77 se muestran los diagramas de violín de los tiempos de ejecución de las aplicaciones de los controladores y la medición. En ella se observa una linea horizontal que representa la mediana de los datos tomados, los límites de la linea vertical representan los cuartiles Q1 y Q3. En la Fig. 78 se observa la distribución de los datos, donde se observa que siguen una distribución con asimetría positiva. Se muestra en términos generales, que los tiempos de ejecución se encuentran por debajo de los $2ms$. La tarea que implica más tiempo de ejecución es la concerniente al sistema de monitoreo, lo cual es lógico porque dentro de este sistema se encuentran diferentes tareas inmersas

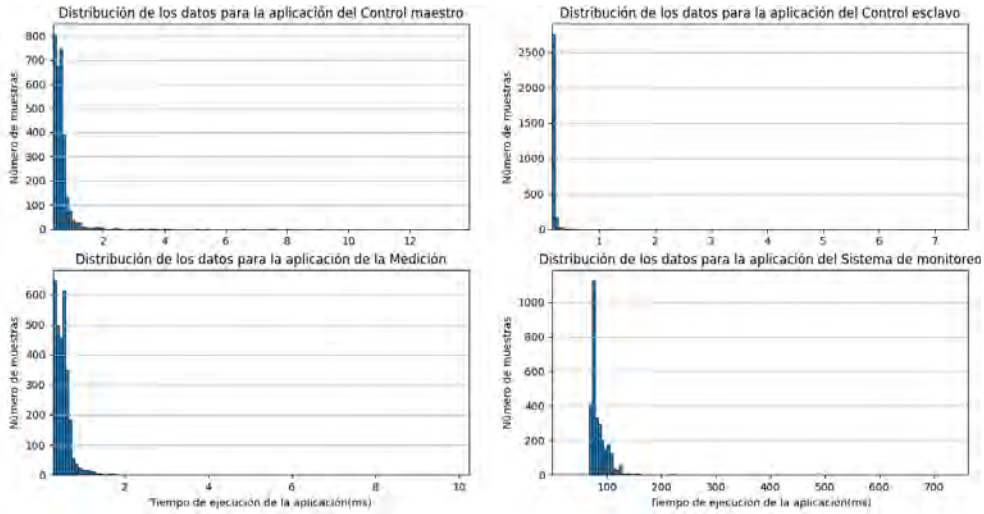


Fig. 78. Histograma del test de latencia.

TABLA XXVII.
Resumen de latencias (valores en ms).

Aplicación	μ	**Mediana**	σ	**Máx.**	**Mín.**
Control maestro	0.6483	0.5725	0.5087	13.9291	0.3722
Control esclavo	0.199	0.1859	0.1524	7.5939	0.163
Medición	0.539	0.4891	0.385	10.242	0.2862
Sistema de monitoreo	85.6775	77.5933	24.83	0.002	763.3198

(gráfica de las variables de interés, el sistema de detección, generación de alarmas,etc. El sistema de detección se puede descomponer en otro componente), que presentan mayor tiempo de cómputo que tareas más simples, como lo son tareas relacionadas con la medición y los algoritmos de control.

En los casos presentados se observa que el tiempo de ejecución de las aplicaciones virtualizadas en los contenedores están por debajo del tiempo de $1s$ cumpliendo con los requerimientos temporales de diseño planteados.

Debido al soporte tecnológico desplegado en Punta Delicia, en cuya planta de control los nodos del sistema están integrados son el soporte de Singularity y DDS, y a las exigencias temporales del caso de control abordado, para el cual los periodos de muestreo no son tan exigentes, se optó por realizar la evaluación solo con Singularity.

6.3 CONCLUSIONES

Este capítulo permitió abordar el tercer objetivo específico, en donde se definieron un par de casos de estudio a implementar, se desarrolló un modelo y un prototipo que permitió evaluar mediante simulación y experimentación el desempeño del sistema frente a los ataques definidos y soportado bajo el procedimiento de diseño establecido.

Los resultados mostraron que la plataforma proporciona un entorno de diseño, análisis y pruebas de alta fidelidad para el flujo de información cibernética y su efecto en la operación física en una planta de procesamiento de bebidas con alta demanda de adaptabilidad, flexibilidad y eficiencia de sus procesos, como se verificó experimentalmente.

Se pudo observar que el sistema de detección de ciberataques tiene un desempeño correcto que permite detectar la ocurrencia de ciberataques del tipo integridad o DoS sobre la variable del proceso. Adicionalmente, se puede clasificar el tipo de ataque que se esta llevando a cabo.

Este capítulo proporciona una contribución al desarrollo e implementación de aplicaciones de automatización industrial que cierran la brecha entre las arquitecturas genéricas y realizaciones físicas mediante el uso de tecnologías de contenedores, los conceptos de microservicios y desacoplamiento de cada microservicio con un middleware basado en la metodología de publicar/subscribir. Para demostrar la aplicabilidad de la arquitectura propuesta se desarrollaron e implementaron dos casos de estudio para lo cual se utilizó el enfoque plug-and-play desde una definición de componentes con sus relaciones concernientes, hasta la implementación e integración con sus respectivas tecnologías implicadas.

Finalmente, los resultados experimentales corresponden a los resultados formales obtenidos del modelo de simulación, lo que indica que la plataforma implementada cumplió con los requisitos del sistema de control. La validación de este proceso permite demostrar que cada desarrollo realizado puede tratarse de forma independiente hasta que los procesos se escalen hasta su punto ideal, reduciendo costos de desarrollo y aplicación final.

7. CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN

En los últimos años se han incorporado sistemas ciberfísicos a las aplicaciones de control automático, lo que ha posibilitado mayor flexibilidad en las aplicaciones al tiempo que ha originado nuevos desafíos, entre los cuales se destacan los relacionados con la ocurrencia de ciberataques que afectan los sistemas control, algo que ya ha generado algunas afectaciones en aplicaciones reales.

Para abordar esta problemática se han utilizado enfoques que han presentado buen desempeño en otros contextos, como las estrategias utilizadas en ambientes de oficina, algunas de las cuales se enfocan principalmente a evitar la ocurrencia de los ataques. Sin embargo, las particularidades de las aplicaciones de automatización industrial plantean requisitos que dificultan la implementación apropiada de estas estrategias y por lo tanto se reduce su efectividad en términos de seguridad, por lo que se identifica la necesidad del desarrollo de nuevos enfoques para detectar y tolerar ciberataques, que se soporten en el análisis de datos y en el conocimiento del sistema.

En este trabajo se presenta una propuesta de diseño de este tipo de aplicaciones, que integra una nueva arquitectura para soportar el desarrollo de estas aplicaciones con estrategias para la detección y el aislamiento de ciberataques de integridad y DoS, la cual además posibilita la implementación de estrategias de tolerancia a los ataques.

La principal contribución de este trabajo consiste en el desarrollo de un procedimiento de diseño de aplicaciones de control soportadas en sistemas ciberfísicos que posibilita la implementación de estrategias de detección y tolerancia de ciberataques, el cual integra un enfoque modular y de fácil adaptación. Este procedimiento permite identificar los diversos componentes del sistema los cuales se establecen como microservicios, e integra un planteamiento para evaluar la planificabilidad de los componentes de la aplicación de control. Las etapas del procedimiento detallan el desarrollo de sistemas de detección y aislamiento de ciberataques que son usados para generar alarmas, a partir de las cuales es posible definir qué elemento del sistema está siendo afectado, posibilitando el uso de estrategias soportadas en réplicas de componentes que permitan el reemplazo de los mismos para tolerar ciberataques. En la literatura se reportan propuestas de arquitecturas para contribuir a la solución de la problemática mencionada, pero en ninguna se abordan las etapas que se incluyeron en este trabajo para vincular las estrategias de detección de ciberataques y la posterior implementación de la solución.

La arquitectura propuesta se soporta en tecnologías de virtualización, específicamente en contenedores, a partir de lo cual se definen los componentes para el desarrollo de las soluciones, y se presentan los modelos que permiten el intercambio de información entre

ellos. Este enfoque permitió implementar en la solución un método de detección y aislamiento de ciberataques basados en redes neuronales convolucionales 1-dimensional y posibilita la incorporación de acciones para mitigar el efecto de los ataques que se detecten, indicando que componente/microservicio está siendo afectado, para no ofrecerlo más o cambiarlo por una réplica que esté funcionando correctamente.

A partir de la arquitectura planteada, se propuso un procedimiento que guía el diseño de aplicaciones de control soportadas en sistemas ciberfísicos que permitan tolerar ciberataques. Este procedimiento permite identificar los diversos componentes/microservicios que integran la solución e integra un planteamiento para analizar la planificabilidad en lo que respecta a la aplicación de control. Del mismo modo involucra etapas de diseño relacionados con sistemas de detección de ciberataques que permiten generar alarmas y definir qué elemento del sistema está siendo afectado, posibilitando el uso de estrategias soportadas en réplicas de componentes que permitan el reemplazo de los mismos para tolerar ciberataques.

Los resultados de la revisión bibliográfica, las simulaciones y los experimentos, obtenidos durante el desarrollo de esta tesis han sido presentados en [157, 160, 161, 193, 202] y [192].

Los resultados obtenidos permiten concluir que:

- Un diseño de software basado en componentes, los conceptos de microservicios, la tecnología de contenedores y la filosofía publicar/subscribir permiten diseñar arquitecturas que soporten los elementos que componen un sistema de control en el entorno de sistemas ciberfísicos.

- Se pueden desarrollar sistemas de detección basado en el aprendizaje profundo que permitan detectar y aislar ciberataques que se presenten de manera simultánea en varias partes del sistema ciberfísico y qué además sean de naturaleza diferente, como lo son los ataques de integridad y DoS, cuya efecto sobre el proceso es diferente.

- El procedimiento de diseño propuesto permitió establecer una serie de pasos que involucra los sistemas de detección de ciberataques, las arquitecturas para el desarrollo de estos sistemas y la verificación de los requisitos temporales. La validación del procedimiento se realizó a través de dos casos de estudio.

Ante las perspectivas de las aplicaciones, los resultados obtenidos y considerando los desafíos de investigación presentados en el capítulo 3, se pueden destacar tres líneas futuras de investigación:

- El desarrollo de algoritmos para la planeación de la activación de componentes para tolerar la ocurrencia de ciberataques, en donde se tengan en consideración los recursos de los nodos, los impactos en el tráfico de la red y en los retrasos en los envíos de los mensajes.

- El desarrollo de sistemas de detección que permitan abordar la dependencia que tienen los sistemas basados en datos. El sistema de detección y aislamiento presentado se basa fuertemente en los datos que se usan en el entrenamiento de los mismos. Si por alguna razón se corrompen los datos, se puede reflejar negativamente en los modelos finales y disminuir sus capacidades para detectar y aislar el ciberataque. Esto es un tema importante y actual que recibe el nombre de Adversarial Machine Learning (AML, por sus siglas en inglés).

- El desarrollo de máquinas de aprendizaje que actualice en linea los modelos antiguos que permiten detectar los ciberataques, usando nuevos datos. Es importante tener en cuenta lo mencionado, dado que el cambio en el comportamiento de los elementos pueden llegar a ser clasificados como la ocurrencia de una anomalía, que luego provocaría una alta tasa de falsos positivos.

REFERENCIAS

[1] A. Humayed, J. Lin, F. Li, y B. Luo, "Cyber-physical systems security - a survey," *IEEE Internet Things J*, vol. 4, no. 6, pp. 1802–1831,.

[2] A. Cardenas, S. Amin, Z.-S. Lin, Y. Huang, C.-Y. Huang, y S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security, ASIACCS 2011*, p. 355–366.

[3] Y. Shoukry, "Smt-based observer design for cyber-physical systems under sensor attacks," in *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems, ICCPS 2016 - Proceedings*.

[4] H. Fawzi, P. Tabuada, y S. Diggavi, "Security for control systems under sensor and actuator attacks," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC*, p. 3412–3417.

[5] Z. Vale, H. Morais, M. Silva, y C. Ramos, "Towards a future scada," in *2009 IEEE Power and Energy Society General Meeting, PES '09*.

[6] Y.-L. Huang, A. A. Cárdenas, S. Amin, Z.-S. Lin, H.-Y. Tsai, y S. Sastry, "Understanding the physical and economic consequences of attacks on control systems," *International Journal of Critical Infrastructure Protection*, vol. 2, no. 3, pp. 73–83, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874548209000213

[7] L. Hu, Z. Wang, y W. Naeem, "Security analysis of stochastic networked control systems under false data injection attacks," in *2016 UKACC International Conference on Control, UKACC Control 2016*.

[8] H. Ge, D. Yue, X. Xie, S. Deng, y Y. Zhang, "Analysis of cyber physical systems security via networked attacks," in *2017 36th Chinese Control Conference (CCC*, p. 4266–4272.

[9] X. Jin y W. Haddad, "An adaptive control architecture for leader-follower multi-agent systems with stochastic disturbances and sensor and actuator attacks," in *2018 Annual American Control Conference (ACC*, p. 980–985.

[10] S. Rebaï, H. Voos, y M. Darouach, "A contribution to cyber-security of networ-ked control systems: An event-based control approach," in *2017 3rd Internatio-nal Conference on Event-Based Control, Communication and Signal Processing (EBCCSP*, p. 1–7.

[11] A. Al-Wosabi y Z. Shukur, “Software tampering detection in embedded systems - a systematic literature review,” *J. Theor. Appl. Inf. Technol*, vol. 76, pp. 211–221,.

[12] W. Knowles, D. Prince, D. Hutchison, J. Disso, y K. Jones, “A survey of cyber security management in industrial control systems,” *Int. J. Crit. Infrastruct. Prot*, vol. 9, pp. 52–80,.

[13] H. Orojloo y M. Azgomi, “A method for evaluating the consequence propagation of security attacks in cyber–physical systems,” *Futur. Gener. Comput. Syst*, vol. 67, pp. 57–71,.

[14] J. Chapman, S. Ofner, y P. Pauksztelo, “Key factors in industrial control system security,” in *2016 IEEE 41st Conference on Local Computer Networks (LCN*, p. 551–554.

[15] G. Bernieri, M. Conti, y F. Pascucci, “A novel architecture for cyber-physical security in industrial control networks,” in *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI*, p. 1–6.

[16] P.-Y. Chen, S. Yang, y J. McCann, “Distributed real-time anomaly detection in networked industrial sensing systems,” *IEEE Trans. Ind. Electron*, vol. 62, pp. 1,.

[17] X. Zhai, “Exploring icmetrics to detect abnormal program behaviour on embedded devices,” *J. Syst. Archit*, vol. 61, no. 10, pp. 567–575,.

[18] Y. Chen, C. Poskitt, y J. Sun, “Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system,” in *Proc. - IEEE Symp. Secur. Priv*, vol. 2018-May, pp. 648–660,.

[19] A. A. C. L. F. Cómbita, J. Giraldo y N. Quijano, “Response and reconfiguration of cyber-physical control systems: A survey,” in *2015 IEEE 2nd Colombian Conference on Automatic Control (CCAC*, p. 1–6.

[20] M. Krotofil, J. Larsen, y D. Gollmann, “The process matters: Ensuring data veracity in cyber-physical systems,” in *ASIACCS 2015 - Proc. 10th ACM Symp. Information, Comput. Commun. Secur*, pp. 133–144,.

[21] M. Ahmadian, M. Shajari, y M. Shafiee, “Industrial control system security taxonomic framework with application to a comprehensive incidents survey,” *Int. J. Crit. Infrastruct. Prot*, vol. 29, pp. 100 356,.

[22] N. Falliere, L. Murchu, y E. Chien, “W32. stuxnet dossier,” *Symantec Secur. Response*, vol. 14, no. February, pp. 1–69,.

[23] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” *IEEE Secur. Priv*, vol. 9, no. 3, pp. 49–51,.

[24] D. Zhang, P. Shi, Q.-G. Wang, y L. Yu, “Analysis and synthesis of networked control systems: A survey of recent advances and challenges,” *ISA Trans*, vol. 66, pp. 376–392,.

[25] Y. Ashibani y Q. Mahmoud, “Cyber physical systems security: Analysis, challenges and solutions,” *Comput. Secur*, vol. 68, pp. 81–97,.

[26] H. Mora, J. Colom, D. Gil, y A. Jimeno-Morenilla, “Distributed computational model for shared processing on cyber-physical system environments,” *Comput. Commun*, vol. 111, pp. 68–83,.

[27] H. Karimipour y V. Dinavahi, “Robust massively parallel dynamic state estimation of power systems against cyber-attack,” *IEEE Access*, vol. 6, pp. 2984–2995,.

[28] E. Lee, “The past, present y future of cyber-physical systems: A focus on models,” *Sensors*, vol. 15, no. 3, pp. 4837–4869,.

[29] P. Mosterman y J. Zander, “Cyber-physical systems challenges: a needs analy-sis for collaborating embedded software systems,” *Softw. Syst. Model*, vol. 15, no. 1, pp. 5–16,.

[30] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, y A. Hahn, “Nist special publication 800-82: Guide to industrial control systems (ics) security.”

[31] L. Sha, “Real time scheduling theory: A historical perspective,” *Real-Time Syst*, vol. 28, no. 2-3 SPEC. ISS., pp. 101–155,.

[32] J. Liu, “Real- time systems.”

[33] M. Spuri, “Holistic Analysis for Deadline Scheduled Real-Time Distributed Systems,” INRIA, Research Report RR-2873, 1996, projet REFLECS. [Online]. Available: https://hal.inria.fr/inria-00073818

[34] N. Audsley, A. Burns, M. Richardson, K. Tindell, y A. J. Wellings, “Applying new scheduling theory to static priority pre-emptive scheduling,” *Software Engineering Journal*, vol. 8, pp. 284–292, 1993.

[35] P. Albertos, A. Crespo, I. Ripoll, M. Valles, y P. Balbastre, “Rt control scheduling to reduce control performance degrading,” in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 5, 2000, pp. 4889–4894 vol.5.

[36] A. Deorankar y S. Thakare, “Survey on anomaly detection of (iot)- internet of things cyberattacks using machine learning,” in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC*, p. 115–117.

[37] R. Mitchell y I.-R. Chen, “A survey of intrusion detection techniques for cyber-physical systems,” *ACM Comput. Surv*, vol. 46.

[38] L. Cao, X. Jiang, Y. Zhao, S. Wang, D. You, y X. Xu, “A survey of network attacks on cyber-physical systems,” *IEEE Access*, vol. 8, pp. 44 219–44 227,.

[39] B. Zarpelão, R. Miani, C. Kawakani, y S. Alvarenga, “A survey of intrusion detection in internet of things,” *J. Netw. Comput. Appl*, vol. 84, pp. 25–37,.

[40] S. Tan, J. Guerrero, P. Xie, R. Han, y J. Vasquez, “Brief survey on attack detection methods for cyber-physical systems,” *IEEE Syst. J*, pp. 1–11,.

[41] J. Yaacoub, O. Salman, H. Noura, N. Kaaniche, A. Chehab, y M. Malli, “Cyber-physical systems security: Limitations, issues and future trends,” *Microprocess. Microsyst*, vol. 77, pp. 103 201,.

[42] H. Noura, D. Theilliol, J.-C. Ponsart, y A. Chamseddine, “Fault-tolerant control systems: Design and practical applications.”

[43] I. Samy, I. Postlethwaite, y D. Gu, “Detection and accommodation of sensor faults in uavs - a comparison of nn and ekf based approaches,” in *Proceedings of the IEEE Conference on Decision and Control*, p. 4365–4372.

[44] V. Justin, N. Marathe, y N. Dongre, “Hybrid ids using svm classifier for detecting dos attack in manet application,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC*, p. 775–778.

[45] D. Zhang, D. Tang, L. Tang, R. Dai, J. Chen, y N. Zhu, “PCA-SVM-Based Approach of Detecting Low-Rate DoS Attack,” in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS*, p. 1163–1170.

[46] A. Theissler, “Anomaly detection in recordings from in-vehicle networks.”

[47] D. Dudek, “Collaborative detection of traffic anomalies using first order markov chains,” in *2012 Ninth International Conference on Networked Sensing (INSS*, p. 1–4.

[48] P. Alpano, J. Pedrasa, y R. Atienza, “Multilayer perceptron with binary weights and activations for intrusion detection of cyber-physical systems,” in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2017-December, p. 2825–2829.

[49] F. Farivar, M. Haghighi, A. Jolfaei, y M. Alazab, “Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber-Physical Systems and Industrial IoT,” *IEEE Trans. Ind. Informatics*, vol. 16, no. 4, pp. 2716–2725,.

[50] J. Shin, Y. Baek, J. Lee, y S. Lee, “Cyber-Physical Attack Detection and Recovery Based on RNN in Automotive Brake Systems.”

[51] D. Shi, W. Fan, Y. Xiao, T. Lin, y C. Xing, “Intelligent scheduling of discrete automated production line via deep reinforcement learning,” *Int. J. Prod. Res*, vol. 58, no. 11, pp. 3362–3380,.

[52] J. Bland, M. Petty, T. Whitaker, K. Maxwell, y W. Cantrell, “Machine learning cyberattack and defense strategies,” *Comput. Secur*, vol. 92, pp. 101 738,.

[53] S. Vieira, W. Pinaya, y A. Mechelli, “Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications,” *Neuroscience and Biobehavioral Reviews*, vol. 74, pp. 58–75,.

[54] M. J. J. Douglass, *Book Review: Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd edition by Aurélien Géron*, 2020, vol. 43, no. 3.

[55] R. Yamashita, M. Nishio, R. Do, y K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4, pp. 611–629,.

[56] B. Polyak, “Some methods of speeding up the convergence of iteration methods,” *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0041555364901375

[57] Y. Nesterov, “A method for unconstrained convex minimization problem with the rate of convergence o($1/k^2$),” 1983.

[58] J. Duchi, E. Hazan, y Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 07 2011.

[59] D. Kingma y J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.

[60] X. Fang, M. Xu, S. Xu, y P. Zhao, “A deep learning framework for predicting cyber attacks rates,” *Eurasip J. Inf. Secur*, vol. 2019, no. 1.

[61] T. Lee, V. P. Singh, y K. H. Cho, “Deep Learning for Time Series,” pp. 107–131, 2021.

[62] F. Bianchi, E. Maiorino, M. Kampffmeyer, A. Rizzi, y R. Jenssen, “An overview and comparative analysis of recurrent neural networks for short term load forecasting.”

[63] H. P. Breivold, A. Jansen, K. Sandström, y I. Crnkovic, “Virtualize for architecture sustainability in industrial automation,” in *2013 IEEE 16th International Conference on Computational Science and Engineering*, 2013, pp. 409–415.

[64] “International Society of Automation (ISA), ANSI/ISA-95.00.01-2000,” Enterprise-Control System Integration - Part 1-5, Tech. Rep., 2007.

[65] F. Hofer, M. Sehr, A. Iannopollo, I. Ugalde, A. Sangiovanni-Vincentelli, y B. Russo, “Industrial control via application containers: Migrating from bare-metal to iaas,” in *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 2019-Decem, pp. 62–69,.

[66] T. Goldschmidt, S. Hauck-Stattelmann, S. Malakuti, y S. Grüner, “Container-based architecture for flexible industrial control applications,” *Journal of Systems Architecture*, vol. 84, pp. 28–36, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762117304988

[67] M. Caliskan, M. Ozsiginan, y E. Kugu, “Benefits of the virtualization technologies with intrusion detection and prevention systems,” in *AICT 2013 - 7th International Conference on Application of Information and Communication Technologies, Conference Proceedings*.

[68] Z. Gu y Q. Zhao, “A state-of-the-art survey on real-time issues in embedded systems virtualization,” *J. Softw. Eng. Appl*, vol. 5, pp. 277–290,.

[69] Y. Bock, J. Broeckhove, y P. Hellinckx, “Hierarchical real-time multi-core scheduling through virtualization: A survey,” in *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, vol. 3PGCIC, p. 611–616.

[70] B. Cilku y P. Puschner, “Towards temporal and spatial isolation in memory hierarchies for mixed-criticality systems with hypervisors,” *Work. Real-Time Mix. Crit. Syst. (ReTiMiCS), Int. Conf. Embed. Real-Time Comput. Syst. Appl*, pp. 25–28,.

[71] A. Crespo, I. Ripoll, y M. Masmano, “Partitioned embedded architecture based on hypervisor: The xtratum approach,” in *EDCC-8 - Proc. 8th Eur. Dependable Comput. Conf*, pp. 67–72,.

[72] I. Arslan y I. Özbilgin, “Sanallaştirma ve güvenlik: Bir sanallaştirma platformu yapisinin incelenmesi,” in *2nd International Conference on Computer Science and Engineering, UBMK 2017*, p. 221–226.

[73] T. Tasci, J. Melcher, y A. Verl, “A container-based architecture for real-time control applications,” in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2018, pp. 1–9.

[74] V. Struhár, M. Behnam, M. Ashjaei, y A. V. Papadopoulos, “Real-Time Containers: A Survey,” in *Fog-IoT*, 2020.

[75] P. González-Nalda, I. Etxeberria-Agiriano, I. Calvo, y M. Otero, “A modular cps architecture design based on ros and docker,” *Int. J. Interact. Des. Manuf*, vol. 11, no. 4, pp. 949–955,.

[76] K. Lab, “Threat landscape for industrial automation systems in h1 2020,” in *ICS Cert*, pp. 1–27,.

[77] Y. Wan, J. Cao, G. Chen, y W. Huang, “Distributed observer-based cyber-security control of complex dynamical networks,” *IEEE Trans. Circuits Syst. I Regul. Pap*, vol. 64, no. 11, pp. 2966–2975,.

[78] L. Marinos, M. Lourenço, y E. Threat, Tech. Rep., landscape Report 2018 15 Top Cyberthreats and Trends, no. January. 2018.

[79] D. Inc, “Trisis malware,” pp. 1–19,.

[80] D. Martínez, P. Balbastre, F. Blanes, J. Simo, y A. Crespo, “Procedimiento de diseño para minimizar el consumo de potencia y los retrasos en wsan.”

[81] “Attackers deploy new ics attack framework ‘triton’ and cause operational disruption to critical infrastructure — fireeye inc,” .

[82] E. Lisova, E. Uhlemann, W. Steiner, J.Åkerberg, y M. Björkman, “A survey of security frameworks suitable for distributed control systems,” in *2015 International Conference on Computing and Network Communications (CoCoNet*, p. 205–211.

[83] J. Slay y M. Miller, “Lessons learned from the maroochy water breach,” in *IFIP International Federation for Information Processing*, vol. 253, p. 73–82.

[84] “Risi - the repository of industrial security incidents.”

[85] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, y N. Weaver, “Inside the slammer worm,” *IEEE Security and Privacy*, vol. 1, no. 4, pp. 33–39,.

[86] “Dam breaks at missouri power plant - cbs news.”

[87] A. Nicholson, S. Webber, S. Dyer, T. Patel, y H. Janicke, “Scada security in the light of cyber-warfare,” *Comput. Secur*, vol. 31, no. 4, pp. 418–436,.

[88] I. Security, “Sistemas de control industrial (ics) principal objetivo de los ciberataques infraestructuras críticas y ciberseguridad industrial : principales conceptos ics – industrial control systems ¿ qué son ?”

[89] T. Daniela, “Communication security in scada pipeline monitoring systems,” in *Proceedings - RoEduNet IEEE International Conference*.

[90] S. Kia, H. Henao, y G. Capolino, “Survey of real-time fault diagnosis techniques for electromechanical systems,” in *2017 IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD*, p. 290–297.

[91] M. Krotofil, A. Cárdenas, J. Larsen, y D. Gollmann, “Vulnerabilities of cyber-physical systems to stale data-determining the optimal time to launch attacks,” *Int. J. Crit. Infrastruct. Prot*, vol. 7, no. 4, pp. 213–232,.

[92] B. Genge, I. Kiss, y P. Haller, “A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures,” *Int. J. Crit. Infrastruct. Prot*, vol. 10, pp. 3–17,.

[93] J. Lin, W. Yu, X. Yang, G. Xu, y W. Zhao, “On false data injection attacks against distributed energy routing in smart grid,” in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, p. 183–192.

[94] S. Amin, S. Amin, y S. Amin, “Smart grid: Overview, issues and opportunities. advances and challenges in sensing, modeling, simulation, optimization and control,” *Eur. J. Control*, vol. 17, no. September, pp. 547–567,.

[95] M. El-Hawary, “The smart grid—state-of-the-art and future trends,” in *2016 Eighteenth International Middle East Power Systems Conference (MEPCON*, p. –.

[96] Z. Liu, F. Wen, y G. Ledwich, “Optimal siting and sizing of distributed generators in distribution systems considering uncertainties,” *IEEE Trans. Power Deliv*, vol. 26, no. 4, pp. 2541–2551,.

[97] A. Moreno-Munoz, V. Pallares-Lopez, J. Rosa, R. Real-Calvo, M. Gonzalez-Redondo, and I. Moreno-Garcia, “Embedding synchronized measurement technology for smart grid development,” *IEEE Trans. Ind. Informatics*, vol. 9, no. 1, pp. 52–61,.

[98] A. Ashok, A. Hahn, y M. Govindarasu, “Cyber-physical security of wide-area monitoring, protection and control in a smart grid environment,” *J. Adv. Res*, vol. 5, no. 4, pp. 481–489,.

[99] A. Srivastava, T. Morris, T. Ernster, C. Vellaithurai, S. Pan, y U. Adhikari, “Modeling cyber-physical vulnerability of the smart grid with incomplete information,” *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 235–244,.

[100] Y. Kwon, H. Kim, K. Koumadi, Y. Lim, y J. Lim, “Automated vulnerability analy-sis technique for smart grid infrastructure,” in *2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT*, p. 1–5.

[101] Z. Li, M. Shahidehpour, y F. Aminifar, “Cybersecurity in distributed power systems,” *Proc. IEEE*, vol. 105, no. 7, pp. 1367–1388,.

[102] C. Peng, H. Sun, M. Yang, y Y. Wang, “A survey on security communication and control for smart grids under malicious cyber attacks,” *IEEE Trans. Syst. Man, Cybern. Syst*, vol. 49, no. 8, pp. 1554–1569,.

[103] S. Ahmed, Y. Lee, S. Hyun, y I. Koo, “Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest,” *IEEE Trans. Inf. Forensics Secur*, vol. 14, no. 10, pp. 2765–2777,.

[104] W. Shang, J. Cui, C. Song, J. Zhao, y P. Zeng, “Research on industrial control anomaly detection based on fcm and svm,” in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/-BigDataSE*, p. 218–222.

[105] W. Yan, L. Mestha, y M. Abbaszadeh, “Attack detection for securing cyber physical systems,” *IEEE Internet Things J*, vol. PP, pp. 1,.

[106] T. Nguyen, S. Wang, M. Alhazmi, M. Nazemi, A. Estebsari, y P. Dehghanian, “Electric power grid resilience to cyber adversaries: State of the art,” *IEEE Access*, vol. 8, pp. 87 592–87 608,.

[107] B. Shinde, S. Wang, P. Dehghanian, y M. Babakmehr, “Real- time detection of critical generators in power systems: A deep learning hcp approach,” p. 1–6.

[108] G. Befekadu, V. Gupta, y P. Antsaklis, “Risk-sensitive control under markov modulated denial-of-service (dos) attack strategies,” *IEEE Trans. Automat. Contr*, vol. 60, no. 12, pp. 3299–3304,.

[109] S. Checkoway, “Comprehensive experimental analyses of automotive attack surfa-ces,” in *Proceedings of the 20th USENIX Conference on Security*, p. 6.

[110] K. Costa, J. Papa, C. Lisboa, R. Munoz, y V. Albuquerque, “Internet of things: A survey on machine learning-based intrusion detection approaches,” *Comput. Networks*, vol. 151, pp. 147–157,.

[111] A. Tabassum, A. Erbad, y M. Guizani, “A survey on recent approaches in intrusion detection system in iots,” in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC*, p. 1190–1197.

[112] J. Petit y S. Shladover, “Potential cyberattacks on automated vehicles,” *IEEE Trans. Intell. Transp. Syst*, vol. 16, no. 2, pp. 546–556,.

[113] W. Wu, “A survey of intrusion detection for in-vehicle networks,” *IEEE Trans. Intell. Transp. Syst*, vol. 21, no. 3, pp. 919–933,.

[114] M. Kang y J. Kang, “A novel intrusion detection method using deep neural network for in-vehicle network security,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring*, p. 1–5.

[115] A. Taylor, S. Leblanc, y N. Japkowicz, “Anomaly detection in automobile control network data with long short-term memory networks,” in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA*, p. 130–139.

[116] Y. Hu, “Detecting stealthy attacks on industrial control systems using a permutation entropy-based method,” *Futur. Gener. Comput. Syst*, vol. 108, pp. 1230–1240,.

[117] A. Zammali, A. Bonneval, y Y. Crouzet, “A diversity-based approach for communication integrity in critical embedded systems,” in *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, p. 215–222.

[118] H. P. Breivold y K. Sandström, “Virtualize for test environment in industrial automation,” in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014, pp. 1–8.

[119] G. Heiser, “The role of virtualization in embedded systems,” pp. 11–16, 04 2008.

[120] M. Wahler, R. Eidenbenz, C. Franke, y Y.-A. Pignolet, “Migrating legacy control software to multi-core hardware,” in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2015, pp. 458–466.

[121] A. Moga, T. Sivanthi, y C. Franke, “Os-level virtualization for industrial automation systems: are we there yet?” *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016.

[122] T. Goldschmidt y S. Hauck-Stattelmann, “Software Containers for Industrial Control,” in *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2016, pp. 258–265.

[123] J. Wu y T.-I. Yang, “Dynamic CPU allocation for Docker containerized mixed-criticality real-time systems,” in *2018 IEEE International Conference on Applied System Invention (ICASI)*, 2018, pp. 279–282.

[124] M. Cinque y D. Cotroneo, “Towards Lightweight Temporal and Fault Isolation in Mixed-Criticality Systems with Real-Time Containers,” in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2018, pp. 59–60.

[125] M. Cinque, R. D. Corte, A. Eliso, y A. Pecchia, “RT-CASEs: Container-Based Virtualization for Temporally Separated Mixed-Criticality Task Sets,” in *ECRTS*, 2019.

[126] J. Melcher, “Design y Implementation of a Container-based Architecture for Real-Time Control Applications,” Master’s thesis, University of Stuttgart, Institute of Software Technology.

[127] F. Hofer, M. A. Sehr, A. Iannopollo, I. Ugalde, A. Sangiovanni-Vincentelli, y B. Russo, “Industrial Control via Application Containers: Migrating from Bare-Metal to IAAS,” in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2019, pp. 62–69.

[128] M. G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, y C. A. F. De Rose, “Performance evaluation of container-based virtualization for high performance computing environments,” in *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 2013, pp. 233–240.

[129] D. G. Pivoto, L. F. de Almeida, R. da Rosa Righi, J. J. Rodrigues, A. B. Lugli, y A. M. Alberti, “Cyber-physical systems architectures for industrial internet of things applications in industry 4.0: A literature review,” *Journal of Manufacturing Systems*, vol. 58, pp. 176–192, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520302119

[130] J. Lee, B. Bagheri, y H.-A. Kao, “A cyber-physical systems architecture for industry 4.0-based manufacturing systems,” *Manufacturing Letters*, vol. 3, pp. 18–23, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221384631400025X

[131] J. Contreras, J. Melo, y J. Díaz Pastrana, “Developing of industry 4.0 applica-tions,” *International Journal of Online Engineering (iJOE)*, vol. 13, p. 30, 11 2017.

[132] “The industrial internet of things volume g1: Reference architecture,” 2019.

[133] F. Buchi, S. Fries, y D. Kroeselberg, “Cyber security standards and regulations in energy automation systems.”

[134] S. Adepu, F. Brasser, L. Garcia, M. Rodler, L. Davi, A.-R. Sadeghi, y S. Zonouz, “Control Behavior Integrity for Distributed Cyber-Physical Systems,” in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, 2020, pp. 30–40.

[135] S. Weerakkody, Y. Mo, y B. Sinopoli, “Detecting integrity attacks on control systems using robust physical watermarking,” in *53rd IEEE Conference on Decision and Control*, 2014, pp. 3757–3764.

[136] S. Weerakkody, B. Sinopoli, S. Kar, y A. Datta, “Information flow for security in control systems,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 5065–5072.

[137] Y. Gao, “Analysis of security threats and vulnerability for cyber-physical systems,” in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, ICCSNT 2013*, p. 50–55.

[138] M. Al-Mhiqani, “Cyber-security incidents: A review cases in cyber-physical systems,” *Int. J. Adv. Comput. Sci. Appl*, vol. 9, no. 1, pp. 499–508,.

[139] A. A. Al-Wosabi, Z. Shukur, y M. Ibrahim, “Framework for software tampering detection in embedded systems,” in *2015 International Conference on Electrical Engineering and Informatics (ICEEI*, p. 259–264.

[140] D. Wang, Z. Wang, B. Shen, F. Alsaadi, y T. Hayat, “Recent advances on filtering and control for cyber-physical systems under security and resource constraints,” *J. Franklin Inst*, vol. 353, no. 11, pp. 2451–2466,.

[141] S. Sridhar y G. Manimaran, “Data integrity attacks and their impacts on scada control system,” *IEEE PES Gen. Meet. PES*, pp. 0–5,.

[142] G. Francia, D. Thornton, y T. Brookshire, “Wireless vulnerability of scada systems,” in *Proceedings of the 50th Annual Southeast Regional Conference*, ser. ACM-SE ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 331–332. [Online]. Available: https://doi.org/10.1145/2184512.2184590

[143] V. Urias, B. Van Leeuwen, y B. Richardson, “Supervisory command and data acquisition (scada) system cyber security analysis using a live, virtual, and constructive (lvc) testbed,” in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, 2012, pp. 1–8.

[144] Z. Lu, X. Lu, W. Wang, y C. Wang, “Review and evaluation of security threats on the communication networks in the smart grid,” in *2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, 2010, pp. 1830–1835.

[145] Y. Liu, P. Ning, y M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 21–32. [Online]. Available: https://doi.org/10.1145/1653662.1653666

[146] C. Li, A. Raghunathan, y N. K. Jha, “Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system,” in *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*, 2011, pp. 150–156.

[147] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy,
B. Kantor, D. Anderson, H. Shacham, y S. Savage, “Experimental security analy-sis of a modern automobile,” in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 447–462.

[148] T. Hoppe, S. Kiltz, y J. Dittmann, “Security threats to automotive can networks—practical examples and selected short-term countermeasures,” *Reliability Engineering System Safety*, vol. 96, no. 1, pp. 11–25, 2011, special Issue on Safecomp 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832010001602

[149] A. Teixeira, D. Pérez, H. Sandberg, y K. Johansson, “Attack models and scenarios for networked control systems,” 04 2012.

[150] Z. Fan, P. Kulkarni, S. Gormus, C. Efthymiou, G. Kalogridis, M. Sooriyabandara, Z. Zhu, S. Lambotharan, y W. H. Chin, “Smart grid communications: Overview of research challenges, solutions, and standardization activities,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 21–38, 2013.

[151] J. S. Obando-Ceron, J. J. Arias-Castro, D. Martínez-Castro, P. A. Manrique-Castillo, y J. C. M-Moreno, “Evaluación del rendimiento de modulos solares h íbridos (fv/t) para el abastecimiento energético de autoclaves hospitalarias,” in *2018 IEEE ANDESCON*, 2018, pp. 1–9.

[152] H. Tian, F. Mancilla-David, K. Ellis, E. Muljadi, y P. Jenkins, “Detailed performan-ce model for photovoltaic systems: Preprint,” 2012.

[153] J. S. Obando Ceron y J. J. Arias Castro, “Prototipo de un sistema de abastecimiento energético para Autoclave Hospitalario soportado en paneles solares híbridos (FV/T),” Master’s thesis, Universidad Autónoma de Occidente, mar 2017. [Online]. Available: http://hdl.handle.net/10614/9449

[154] F. Mohamed, “Microgrid Modelling and Simulation,” Licentiate Thesis, Helsinki University of Technology, Finland, March 2006.

[155] B. Kuang, Y. Wang, y Y. Tan, “An h∞ controller design for diesel engine sys-tems,” vol. 1, 02 2000, pp. 61 – 66 vol.1.

[156] M. Saeed, S. Fawzy, y M. El-Saadawi, “Modeling and simulation of biogas-fueled power system,” *International Journal of Green Energy*, vol. 16, no. 2, pp. 125–151, 2019.

[157] C. M. Paredes, R. E. Alzate, D. M. Castro, A. F. Bayona, y D. R. García, “Detec-tion and isolation of dos and integrity attacks in cyber-physical microgrid system,” in *2019 IEEE 4th Colombian Conference on Automatic Control (CCAC)*, 2019, pp. 1–6.

[158] M. Karbasforooshan y M. Monfared, “Design and implementation of a single-phase shunt active power filter based on pq theory for current harmonic compensation in electric distribution networks,” in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2017, pp. 6389–6394.

[159] F. Alexis y D. Mera, “Modelamiento y control de una microrred en modo isla,” Masther Thesis, June 2015.

[160] C. M. Paredes, A. F. Bayona, D. Martínez, A. Crespo, J. Simo, y A. González, “Socio-economic and technological impact of a microgrid in isolated communities using simulation modeling,” in *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, vol. 1, 2021, pp. 649–656.

[161] C. M. Paredes, A. F. Bayona, D. Martínez, A. Crespo, A. González, y J. Simo, “Approach to an emulation model to evaluate the behavior and impact of

microgrids in isolated communities," *Energies*, vol. 14, no. 17, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/17/5316

[162] L. F. Cómbita, A. A. Cárdenas, and N. Quijano, "Mitigation of sensor attacks on legacy industrial control systems," in *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, 2017, pp. 1–6.

[163] Y. Li, J. Li, X. Luo, X. Wang, y X. Guan, "Cyber attack detection and isolation for smart grids via unknown input observer," in *2018 37th Chinese Control Conference (CCC)*, 2018, pp. 6207–6212.

[164] Z. Wang, Y. Zhao, K. Yang, J. Yao, Z. Ding, y K. Zhang, "Uio-based cyber attack detection and mitagation scheme for load frequency control system," in *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, 2019, pp. 1257–1262.

[165] X. Xie, B. Wang, T. Wan, y W. Tang, "Multivariate abnormal detection for industrial control systems using 1d cnn and gru," *IEEE Access*, vol. 8, pp. 88 348–88 359, 2020.

[166] B. Siegel, "Industrial anomaly detection: A comparison of unsupervised neural network architectures," *IEEE Sensors Letters*, vol. 4, no. 8, pp. 1–4, 2020.

[167] G. Bernieri, M. Conti, y F. Turrin, "Evaluation of machine learning algorithms for anomaly detection in industrial networks," in *2019 IEEE International Symposium on Measurements Networking (M N)*, 2019, pp. 1–6.

[168] A. N. Sokolov, I. A. Pyatnitsky, y S. K. Alabugin, "Research of classical machine learning methods and deep learning models effectiveness in detecting anomalies of industrial control system," in *2018 Global Smart Industry Conference (GloSIC)*, 2018, pp. 1–6.

[169] N. Elmrabit, F. Zhou, F. Li, y H. Zhou, "Evaluation of machine learning algorithms for anomaly detection," in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 2020, pp. 1–8.

[170] F. Li, Q. Li, J. Zhang, J. Kou, J. Ye, W. Song, y H. A. Mantooth, "Detection and Diagnosis of Data Integrity Attacks in Solar Farms Based on Multilayer Long Short-Term Memory Network," *IEEE Transactions on Power Electronics*, vol. 36, no. 3, pp. 2495–2498, 2021.

[171] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, y S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[172] D. E. Kim y M. Gofman, “Comparison of shallow and deep neural networks for network intrusion detection,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 204–208.

[173] J. Nie, P. Ma, B. Wang, y Y. Su, “A covert network attack detection method based on lstm,” in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020, pp. 1690–1693.

[174] A. R. Javed, M. Usman, S. U. Rehman, M. U. Khan, y M. S. Haghighi, “Anomaly detection in automated vehicles using multistage attention-based convolutional neural network,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[175] J. Goh, S. Adepu, K. Junejo, y A. Mathur, “A dataset to support research in the design of secure water treatment systems,” 10 2016.

[176] D. Shalyga, P. Filonov, y A. Lavrentyev, “Anomaly detection for water treatment system based on neural network with automatic architecture optimization,” 2018.

[177] M. Kravchik y A. Shabtai, “Detecting cyber attacks in industrial control systems using convolutional neural networks,” ser. CPS-SPC ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 72–83. [Online]. Available: https://doi.org/10.1145/3264888.3264896

[178] Q. Lin, S. Adepu, S. Verwer, y A. Mathur, “Tabor: A graphical model-based approach for anomaly detection in industrial control systems,” in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ser. ASIACCS ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 525–536. [Online]. Available: https://doi.org/10.1145/3196494.3196546

[179] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, y J. Sun, “Anomaly detection for a water treatment system using unsupervised machine learning,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 1058–1065.

[180] M. Macas y C. Wu, “An unsupervised framework for anomaly detection in a water treatment system,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 1298–1305.

[181] M. Kravchik y A. Shabtai, “Efficient cyber attacks detection in industrial control systems using lightweight neural networks,” *ArXiv*, vol. abs/1907.01216, 2019.

[182] E. Mousavinejad, X. Ge, Q.-L. Han, F. Yang, y L. Vlacic, “Resilient tracking control of networked control systems under cyber attacks,” *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 2107–2119, 2021.

[183] S. Bezzaoucha Rebaï, H. Voos, y M. Darouach, “Attack-tolerant control and observer-based trajectory tracking for cyber-physical systems,” *European Journal of Control*, vol. 47, pp. 30–36, 2019. [Online]. Available: https: // www.sciencedirect.com/science/article/pii/S0947358018300293

[184] S. B. Rebaï y H. Voos, “Chapter 13 - observer-based event-triggered attack-tolerant control design for cyber-physical systems,” in *New Trends in Observer-Based Control*, ser. Emerging Methodologies and Applications in Modelling, O. Boubaker, Q. Zhu, M. S. Mahmoud, J. Ragot, H. R. Karimi, and J. Dávila, Eds. Academic Press, 2019, pp. 439–462. [Online]. Available: https://www.sciencedirect.com/ science/article/pii/B9780128170380000135

[185] A. K. Prajapati y B. Roy, “Multi-fault diagnosis in three coupled tank system using unknown input observer,” *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 47–52, 2016, 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016. [Online]. Available: https: //www.sciencedirect.com/science/article/pii/S2405896316300271

[186] Y. Zhang, Z. Wang, L. Ma, y F. E. Alsaadi, “Annulus-event-based fault detection, isolation and estimation for multirate time-varying systems: Applications to a three-tank system,” *Journal of Process Control*, vol. 75, pp. 48–58, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959152418305432

[187] M. Stone, “Cross-validatory choice and assessment of statistical predictions,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 111–147, 1974. [Online]. Available: http://www.jstor.org/stable/2984809

[188] H. Wainer, *Journal of Educational Statistics*, vol. 13, no. 4, pp. 358–364, 1988. [Online]. Available: http://www.jstor.org/stable/1164710

[189] R. Taormina, S. Galelli, N. O. Tippenhauer, E. Salomons, A. Ostfeld, D. G. Eliades, M. Aghashahi, R. Sundararajan, M. Pourahmadi, M. K. Banks, B. M. Brentan, E. Campbell, G. Lima, D. Manzi, D. Ayala-Cabrera, M. Herrera, I. Montalvo, J. Izquierdo, E. Luvizotto, S. E. Chandy, A. Rasekh, Z. A. Barker, B. Campbell, M. E. Shafiee, M. Giacomoni, N. Gatsis, A. Taha, A. A. Abokifa, K. Haddad, C. S. Lo, P. Biswas, M. F. K. Pasha, B. Kc, S. L. Somasundaram, M. Housh, y Z. Ohar,

"Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks," *Journal of Water Resources Planning and Management*, vol. 144, no. 8, p. 04018048, aug 2018.

[190] "BATADAL - Datasets," http://www.batadal.net/data.html.

[191] K. Kamycki, T. Kapuscinski, y M. Oszust, "Data augmentation with suboptimal warping for time-series classification," *Sensors*, vol. 20, no. 1, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/98

[192] V. Ibarra-Junquera, A. González, C. M. Paredes, D. Martínez-Castro, y R. A. Nuñez-Vizcaino, "Component-based microservices for flexible and scalable automation of industrial bioprocesses," *IEEE Access*, vol. 9, pp. 58 192–58 207, 2021.

[193] C. M. Paredes, D. Martínez-Castro, V. Ibarra-Junquera, y A. González-Potes, "Detection and isolation of dos and integrity cyber attacks in cyber-physical systems with a neural network-based architecture," *Electronics*, vol. 10, no. 18, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/18/2238

[194] P. Balbastre, I. Ripoll, y A. Crespo, "Minimum deadline calculation for periodic real-time tasks in dynamic priority systems," *IEEE Transactions on Computers*, vol. 57, no. 1, pp. 96–109, 2008.

[195] H. Serrano-Magaña, A. González-Potes, V. Ibarra-Junquera, P. Balbastre, D. Martínez-Castro, y J. Simó, "Software components for smart industry based on microservices: A case study in ph control process for the beverage industry," *Electronics*, vol. 10, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/7/763

[196] A. Nejati, M. Shahrokhi, y A. Mehrabani, "Comparison between backstepping and input–output linearization techniques for ph process control," *Journal of Process Control*, vol. 22, no. 1, pp. 263–271, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959152411001703

[197] V. Ibarra-Junquera, S. Jørgensen, J. Virgen-Ortíz, P. Escalante-Minakata, y J. Osuna-Castro, "Following an optimal batch bioreactor operations model," *Chemical Engineering and Processing*, vol. 62, p. 114–128, 08 2012.

[198] J.-J. E. Slotine y W. Li, "Applied nonlinear control," 1991.

[199] Z. Ding, *Nonlinear and Adaptive Control Systems*, 04 2013.

[200] N. Griba, F. Hamidi, K. Menighed, B. Boussaid, y M. N. Abdelkrim, “Synchronization of chaotic systems: a survey study,” in *2019 International Conference on Signal, Control and Communication (SCC)*, 2019, pp. 262–267.

[201] J. Pena Ramirez, E. Garcia, y J. Alvarez, “Master-slave synchronization via dynamic control,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 80, p. 104977, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1007570419302965

[202] C. M. Paredes Valencia, “Procedimiento de diseÑo de sistemas ciberfÍsicos de tiempo real tolerantes a ataques cibernÉticos,” *Encuentro Internacional de Educación en Ingeniería*, ago. 2019. [Online]. Available: https://acofipapers.org/index.php/eiei/article/view/276