



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

DETECCIÓN Y LOCALIZACIÓN DE FUGAS EN REDES DE
DISTRIBUCIÓN DE AGUA POTABLE: UN ENFOQUE
BASADO EN APRENDIZAJE POR REFUERZO

Br. Manuel Salvador Gil Matheus

Mérida, Noviembre, 2019



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

DETECCIÓN Y LOCALIZACIÓN DE FUGAS EN REDES DE
DISTRIBUCIÓN DE AGUA POTABLE: UN ENFOQUE
BASADO EN APRENDIZAJE POR REFUERZO

Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero
Electricista

Br. Manuel Salvador Gil Matheus
Tutor: Dr. José Luis Paredes

Mérida, Noviembre, 2019

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

DETECCIÓN Y LOCALIZACIÓN DE FUGAS EN REDES DE
DISTRIBUCIÓN DE AGUA POTABLE: UN ENFOQUE
BASADO EN APRENDIZAJE POR REFUERZO

Br. Manuel Salvador Gil Matheus

Trabajo de Grado, presentado en cumplimiento parcial de los requisitos exigidos para optar al título de Ingeniero Electricista, aprobado en nombre de la Universidad de Los Andes por el siguiente Jurado.

M.Sc Francisco J. Vilorio M.

Prof. Marco Antonio Molina Paz

Dr. José Luis Paredes Quintero

AGRADECIMIENTOS

A mi hermana Carolina que sin su apoyo moral y económico durante todo el transcurso de la carrera, probablemente la realización de este trabajo de grado nunca se hubiese llevado a cabo. ¡Muchas gracias!.

Al Profesor José Luis Paredes por compartir sus conocimientos, por su orientación y en general, por brindarme la oportunidad de trabajar un tema con el cual pude expandir mis conocimientos, y que ha sido de inimaginable utilidad para mi crecimiento profesional. ¡Muchas gracias!.

A la Universidad de Los Andes, y de forma especial a la Escuela de Ingeniería Eléctrica, cuyos profesores son una muestra de gran profesionalismo, a su personal técnico y administrativo por brindar siempre su colaboración de la mejor manera posible, ¡Muchas gracias!.

A mi familia y amigos, y en general, cualquier persona que de alguna manera haya influido, de forma positiva o negativa, en la generación de un curso de acción con el que culmina este trabajo de grado, y comienza una nueva etapa de mi vida. ¡Muchas gracias!.

A la empresa GoToDigital por brindar el recurso financiero y espacio físico para el desarrollo de este trabajo de grado, ¡Muchas gracias!.

Manuel Salvador Gil Matheus. Detección y localización de fugas en Redes de Distribución de agua potable: Un enfoque basado en Aprendizaje por Refuerzo. Tutor: Dr. José Luis Paredes. Noviembre, 2019.

Resumen

El incremento de la demanda de recursos naturales, específicamente del suministro de agua potable, debido al aumento exponencial de la población en las últimas décadas, ha puesto como punto de máximo interés el uso óptimo del vital líquido. En este sentido, reducir al mínimo el agua que se pierde en los sistemas de distribución de agua potable producto de fugas, cobra vital importancia. El presente proyecto aborda el problema de detección y localización de fugas en redes de distribución de agua potable haciendo uso de los avances en el área de inteligencia artificial, específicamente los desarrollos en algoritmos basados en aprendizaje por refuerzo. Se propone el uso del algoritmo DQN (*Deep Q-Network*) para abordar el problema de detección y localización de fugas, especialmente aquellas cuya dimensión (diámetro del orificio) es reducida. La metodología utilizada consiste en enfocar el problema como una tarea de clasificación, donde las clases consideradas se corresponden con un evento de fuga en alguna de las zonas de la red hidráulica. Además, para aprovechar la flexibilidad de este paradigma del aprendizaje automático, la solución del problema abordado involucra dos tareas distintas: una consiste en la selección inteligente de los subsectores a observar y la otra al proceso de clasificar; de esta forma, el sistema creado es capaz de usar sólo las zonas más relevantes de la red para detectar y localizar fugas en la misma.

Descriptor: aprendizaje por refuerzo, aprendizaje automático, redes hidráulicas, fugas de agua, tareas de clasificación, programación en Python.

ÍNDICE GENERAL

APROBACIÓN	ii
RECONOCIMIENTOS	iii
RESUMEN	iv
INTRODUCCIÓN	1
Capítulo	pp.
1. FUNDAMENTOS	4
1.1. ANTECEDENTES	4
1.2. PLANTEAMIENTO DEL PROBLEMA	6
1.3. JUSTIFICACIÓN	8
1.4. OBJETIVOS	9
1.4.1. General	9
1.4.2. Específicos	9
1.5. METODOLOGÍA	9
1.6. ALCANCE Y LIMITACIONES	10
1.6.1. Alcance	10
1.6.2. Limitaciones	11
1.7. ORGANIZACIÓN DEL DOCUMENTO	11
2. MARCO TEÓRICO	12
2.1. APRENDIZAJE POR REFUERZO	12
2.1.1. Interacción del agente con su ambiente	13
2.1.2. Procesos de decisión de Markov	14
2.1.3. Funciones de valoración	15
2.1.4. Aprendizaje por diferencias temporales y Q-Learning	16
2.1.5. Aproximación de la función Q	17
2.1.6. Redes neuronales como función de aproximación.	18
2.1.7. Mejoras del algoritmo DQN	20
2.1.7.1. Uso de memoria y técnicas de muestreo	20
2.1.7.2. DQN con red objetivo separada	21
2.1.7.3. Doble DQN	22
2.1.7.4. Dueling DQN	22
2.1.8. Evaluación de desempeño	23

3. REDES DE DISTRIBUCIÓN DE AGUA POTABLE	25
3.1. SISTEMAS DE DISTRIBUCIÓN	25
3.1.1. Relación entre presión y caudal	26
3.2. PÉRDIDAS DE AGUA EN LAS REDES DE DISTRIBUCIÓN	27
3.3. DETECCIÓN DE FUGAS	28
3.4. APRENDIZAJE SUPERVISADO PARA DETECCIÓN DE FUGAS	30
3.5. MODELO DE REDES HIDRÁULICAS	30
3.5.1. Pérdidas de presión en las tuberías	31
3.5.2. Modelo de fugas	31
4. DETECCIÓN Y LOCALIZACIÓN DE FUGAS	33
4.1. DETECCIÓN Y LOCALIZACIÓN DE FUGAS COMO UN PROBLEMA DE CLASIFICACIÓN	33
4.1.1. Residuos de presión	33
4.2. SECTORIZACIÓN	34
4.2.1. Red como un grafo	35
4.2.1.1. Propiedades de los grafos.	36
4.2.2. Red hidráulica considerada	37
4.2.2.1. Zonas de fuga	38
4.2.3. Selección de datos	40
4.3. APRENDIZAJE POR REFUERZO	41
4.3.1. Observaciones	41
4.3.2. Acciones	42
4.3.3. Función de recompensa	42
4.3.4. Episodio	43
4.3.5. Ambiente	44
4.4. ARQUITECTURA Y ENTRENAMIENTO DEL AGENTE	46
4.4.1. Entrenamiento	46
4.4.2. Curvas de Aprendizaje.	47
4.5. RESULTADOS	48
4.5.1. Métricas del clasificador	49
4.5.2. Matriz de confusión	49
4.5.3. Evaluación por zona	50
4.5.4. Características usadas para la clasificación	51
CONCLUSIONES	53
RECOMENDACIONES	55
APÉNDICE A. GYM HIDRÁULICO	57
REFERENCIAS	63

ÍNDICE DE FIGURAS

Figura	pp.
2.1. Interacción entre un agente y su ambiente.	14
2.2. Arquitectura de una red neuronal.	19
3.1. Red de prueba Net3 de EPANET.	26
3.2. Relación entre caudal de una fuga y duración de la misma	29
3.3. Relación entre presión y caudal de una fuga.	32
4.1. Red de prueba Net1 de EPANET	35
4.2. Patrones de tiempo para diferentes nodos de la red.	38
4.3. Sectorización de la red Net3 de EPANET.	39
4.4. Curvas de aprendizaje.	47
4.5. Duración de los episodios.	48
4.6. Matriz de confusión de la puesta en producción.	50
4.7. Métricas de desempeño por clase	51
4.8. Zonas monitoreadas para la clasificación.	52

ÍNDICE DE TABLAS

Tabla	pp.
4.1. Zonas de fuga en la red de prueba Net3.	40
4.2. Parámetros para el entrenamiento del agente.	46
4.3. Métricas de desempeño	49

www.bdigital.ula.ve

ÍNDICE DE ALGORITMOS

Algoritmo	pp.
2.1. Algoritmo Q-Learning Tabular	17
4.1. Algoritmo de sectorización	37
4.2. Algoritmo para simulación secuencial del ambiente	45

www.bdigital.ula.ve

INTRODUCCIÓN

Las fugas de agua en las redes de distribución de agua potable, son un problema que afecta la operación de estas infraestructuras, ocasionando importantes pérdidas de dinero para las empresas encargadas de su operación, además, se debe agregar los problemas inherentes al desperdicio de un recurso de suma importancia como lo es el agua. Según el Informe Mundial de las Naciones Unidas sobre el Desarrollo de los Recursos Hídricos 2018 [1], como consecuencia del acelerado crecimiento de la población y los efectos del cambio climático, alrededor de 2 mil millones de personas no tienen acceso a servicios de agua potable, por tanto, la gestión de este recurso se torna cada vez más importante.

La magnitud de las pérdidas de agua, producto de las fugas, depende en gran medida de la dimensión de las mismas [2], en este sentido, mientras menor sea la salida de agua, más leve serán los efectos sobre las variables del sistema, por lo tanto, la detección de estas a través de los métodos tradicionales es muy improbable, trayendo como consecuencia la generación de grandes pérdidas de agua en el largo plazo.

La detección y localización de fugas en redes hidráulicas, en general, se realiza a través de la inspección rutinaria en las adyacencias de las tuberías, que se encuentran distribuidas a lo largo de toda la red, con el objeto de detectar cualquier anomalía que pueda ser producto de una fuga de agua. Aprovechando los últimos avances en el área de la computación, han surgido varios acercamientos para resolver el problema de detección y localización de fugas en redes de agua que hacen uso de modelos de las redes en cuestión, en conjunto con algoritmos desarrollados en el área de la inteligencia artificial.

Para aprovechar el éxito de los algoritmos desarrollados en el área del aprendizaje automático, una de las ramas de la inteligencia artificial, el problema de detección y localización de fugas

es enfocado como un problema de clasificación, donde, conocida la existencia de un conjunto de clases o categorías, se busca la regla o conjunto de reglas que permitan asignar cada nueva observación a la clase que pertenece [3]. Tradicionalmente, en este tipo de problemas, donde se dispone de toda la información concerniente a las observaciones de las diferentes clases, los algoritmos basados en aprendizaje supervisado han demostrado gran éxito. Sin embargo, los esquemas seguidos bajo esta metodología, requieren el uso de todas las características de las instancias de cada clase.

La flexibilidad de los algoritmos basados en aprendizaje por refuerzo permite resolver problemas de diversa índole, a través del modelado adecuado de estos como un problema de decisión secuencial. En este paradigma del aprendizaje automático, un agente, es una entidad capaz de tomar acciones basadas en su experiencia y en las nuevas observaciones que reciba desde su entorno o ambiente. El ambiente, define las variables referentes a la problemática que se plantea abordar, además, establece los mecanismos necesarios para que la interacción con un agente sea posible, en otras palabras, se encarga de la generación de una señal de recompensa y las observaciones que el agente recibe.

La señal de recompensa cuantifica la idoneidad de las acciones que el agente pueda tomar, en el sentido de maximizar el valor numérico que recibirá en un futuro. Por otro lado, las observaciones son una representación del estado actual del ambiente, que brindan información necesaria al agente para que este pueda actuar, creando, además, las reglas de transición entre los estados que el agente puede transitar.

En el marco de aprendizaje por refuerzo la detección y localización de fugas, problema tratado en el presente trabajo de grado, se enfoca como una tarea de clasificación. El ambiente o entorno viene definido por las variables concernientes a la red hidráulica en cuestión, que se obtienen a través de la simulación de un modelo de la red. En cada instante de la simulación hidráulica, un agente toma observaciones desde la red, y en función de estas, asigna una etiqueta de clase, que se corresponde con alguno de los escenarios considerados. Es decir, un evento de fuga en algún punto de la red.

El desarrollo de este trabajo de grado se encuentra organizado en cuatro capítulos. La

justificación y exposición de motivos se realiza en el capítulo I. Los conceptos referentes a la teoría de aprendizaje por refuerzo, que se encuentran más estrechamente relacionados con el problema abordado, junto con la descripción de algunas de las técnicas desarrolladas para acelerar y estabilizar los algoritmos utilizados se detallan en el capítulo II. La terminología asociada con el funcionamiento de las redes de distribución de agua potable, la definición formal de fuga de agua y el efecto de estas sobre el comportamiento de las variables de la red, junto con la descripción de los aspectos concernientes al modelado de redes hidráulicas, haciendo uso librerías de Python, se describen en el capítulo III.

En el capítulo IV se expone la metodología implementada para resolver el problema de detección y localización de fugas en las redes hidráulicas. En este sentido, se detalla el procedimiento para sectorizar la red en zonas las fugas, proceso que ayuda a reducir el número de clases en el problema de clasificación considerado, además, se especifica el esquema seguido para detectar y localizar las fugas en las redes de distribución de agua potable, también, se exponen los resultados de los modelos obtenidos, evaluando el desempeño a través de las matrices de confusión, ampliamente usadas para la evaluación de clasificadores multiclase [4].

www.bdigital.ula.ve

CAPÍTULO 1

FUNDAMENTOS

Este capítulo describe las bases necesarias para realizar la investigación, se realiza el planteamiento del problema, las razones o justificación de la problemática tratada, se establecen los objetivos a desarrollar, la metodología a seguir, los alcances y las limitaciones que se tuvieron durante la ejecución del proyecto.

1.1 ANTECEDENTES

El problema de clasificación es ampliamente abordado con los algoritmos de aprendizaje automático basados en la metodología de aprendizaje supervisado, en estos, cada patrón o dato de entrada debe ir acompañado con la respuesta o etiqueta de clase, de esta forma, se pueden desarrollar arquitecturas que obtienen una alta exactitud en el proceso de clasificación. Por ejemplo, en [5], [6], se entrenan clasificadores basados en aprendizaje supervisado obteniendo resultados en el rango de 70-80 % durante el proceso de puesta en producción. No obstante, al utilizar metodologías que consisten en la combinación de más de un algoritmo de clasificación, obtienen exactitudes por encima del 90 %. Sin embargo el mejor resultado lo obtiene el algoritmo de bosques aleatorios¹ alcanzando exactitudes del orden del 100 %.

Sin embargo, varios acercamientos a este problema bajo el enfoque de aprendizaje por refuerzo, han demostrado gran potencia para abordar tareas relacionadas con procesos de clasificación [7], [8], [9], [10], no obstante, este tipo de aprendizaje, si bien no se encuentra enmarcado en la metodología supervisada, requiere bases de datos con datos etiquetados, esto con el objetivo de diseñar una función de recompensa que aliente al agente a obtener una alta exactitud.

¹En inglés *Random Forest*

Uno de los problemas que se debe tener en cuenta a la hora de abordar tareas de clasificación, está relacionado con la distribución de las instancias existentes en las diferentes clases, en otras palabras, el número de ejemplos que pertenecen a cada clase, dentro de un problema en particular, es diferente.

Si bien, existen varias técnicas desarrolladas para atacar esta problemática, que, en principio, permiten que las metodologías tradicionales de clasificación sigan teniendo relativo éxito, por ejemplo, el uso de muestreo estratificado. Alternativamente, en [9] abordan este problema mediante el uso de algoritmos basados en aprendizaje por refuerzo, para ello, elaboran una estrategia que consiste en la definición de una función de recompensa, que asigna mayor importancia a la clasificación correcta de las instancias que pertenecen a la clase minoritaria o con menor número de muestras. De esta forma, la sensibilidad hacia a las instancias pertenecientes a esta clase es mayor.

Para lograr esto, la recompensa o penalización que se entrega al agente por la clasificación de una instancia de la clase minoritaria es máxima, de otra forma, la recompensa consiste en un valor numérico que se ajusta según el grado de importancia otorgado a las instancias de la clase mayoritaria, siendo menor que el máximo valor que se puede entregar como recompensa.

Una de las características ofrecidas por los algoritmos basados en aprendizaje por refuerzo es su flexibilidad a la hora de abordar los problemas desde diversos ángulos, por ejemplo, en [7] se enfoca la tarea de clasificación de series temporales, como un problema de optimización del tiempo involucrado en el proceso, de esta manera, se crea un agente capaz de realizar la clasificación usando para ello el menor tiempo posible. Así, el agente decide bien sea predecir una etiqueta de clase o bien esperar por más muestras.

Nuevamente, la definición de recompensa es fundamental para orientar al agente a desarrollar determinados comportamientos, en este sentido, la función de recompensa, definida en [7], entrega al agente una recompensa negativa, proporcional a los pasos de tiempo que le tome al agente realizar la clasificación, de esta manera, no sólo se optimiza el tiempo de clasificación, sino que además, la cantidad de características que conforman una serie temporal, usadas para poder asignar una etiqueta de clase, es reducida.

En este sentido, dado que todo proceso de clasificación necesita de un conjunto de datos para poder obtener una serie de reglas generales o características de cada una de las clases consideradas [3], y que, en general, la disposición de estos datos es limitada y tienen un costo asociado, la reducción de información necesaria para poder asignar dentro de una categoría a determinada muestra cobra vital importancia.

En [8] se plantea este problema del uso de características costosas, y utilizan aprendizaje por refuerzo para diseñar un sistema que selecciona de forma concienzuda, las características estrictamente necesarias para el proceso de clasificación. Teniendo como objetivo alcanzar una alta exactitud, y, además, minimizar el costo en el que se incurre en el proceso de incorporar nuevas características. Para representar el escenario que se describe, las características de cada instancia se asocian con un valor de costo, así, la recompensa entregada al agente por seleccionar determinada característica es relacionada con este valor.

El proceso de clasificación que se lleva a cabo usando sólo determinadas características del conjunto de datos involucra dos tareas diferentes, que es el proceso de clasificación como tal, y la selección de características. En estudios como [7], [8] la arquitectura de agente implementada se encarga de ambas tareas, sin embargo, también es posible implementar sistemas de clasificación que discriminen las características que se utilizan, separando estas dos acciones.

En [10] el esquema de clasificación planteado utiliza aprendizaje por refuerzo para seleccionar, desde un conjunto de datos, las mejores características que ayuden al proceso de clasificación. La tarea de clasificar las muestras se deja en manos de un clasificador externo, que es pre-entrenado bajo el enfoque de aprendizaje supervisado, haciendo posible combinar ambas estrategias para desarrollar un sistema de clasificación que optimiza el uso de los datos disponibles sin comprometer la exactitud de esta.

1.2 PLANTEAMIENTO DEL PROBLEMA

El proceso de detección y localización de fugas en redes de distribución de agua potable representa un reto significativo debido a la complejidad inherente de los sistemas de este tipo.

Los efectos y consecuencias de esta problemática son diversos, entre ellos se pueden mencionar las pérdidas económicas generadas a las empresas administradoras de las redes de tratamiento y distribución de agua, los problemas relacionados con el suministro del servicio a los usuarios de la red o el impacto negativo sobre el uso de los recursos hídricos.

En general, las fugas de agua se originan por gran cantidad de factores, en muchos casos, es difícil establecer una única causa. Sin embargo, según [2] se pueden mencionar factores físicos o estructurales, las condiciones geográficas en las que determinadas secciones de la red se encuentran instaladas (factores medio ambientales, tales como el nivel de salinidad), poco control sobre la presión en los elementos del sistema y problemas relacionados con el mantenimiento de la red.

Debido a la complejidad de los sistemas de distribución de agua potable, la detección y localización de fugas es una tarea no trivial. Especialmente cuando estas son de magnitud reducida, ya que el efecto que estas tienen sobre las variables en la red, tales como presión y flujo, son leves, y, por lo tanto, difíciles de detectar. Sin embargo, a pesar de no tener un efecto pronunciado, debido a la baja tasa de detección, generan pérdidas de agua considerables en el largo plazo.

Los métodos aplicados en el problema de detección de fugas consisten, generalmente, en sistemas de gestión de presiones de la red o la comparación del caudal generado con el registrado en los puntos de consumo final [11], junto con gran variedad de estrategias de inspección de la red, que se basan en la búsqueda activa de alteraciones en las adyacencias de los sistemas de tuberías, provocadas por las pérdidas de agua u otro tipo de anomalía, esto implica el desplazamiento de equipo y personal especializado a lo largo de toda la extensión geográfica de la red, generando altos costos para los encargados de la administración de la misma.

Aprovechando los avances en el área de la inteligencia artificial, se han creado métodos de detección y localización de fugas haciendo uso de modelos de redes hidráulicas y los algoritmos desarrollados en esta área. Una de las metodologías consiste en considerar el problema como una tarea de clasificación, donde los diferentes escenarios de fuga que se pueden generar en una red

hidráulica, son asociados con etiquetas de clase. En el presente trabajo de grado, se utiliza esta metodología de clasificación bajo un enfoque basado en aprendizaje por refuerzo, para detectar y localizar fugas en redes de distribución de agua potable.

1.3 JUSTIFICACIÓN

Una de las áreas que busca explotar las técnicas de aprendizaje por refuerzo, es el control de procesos industriales. En estos procesos se busca optimizar la utilización de los recursos disponibles, minimizando el costo energético y mejorando a la vez la calidad del producto de dicho proceso, por otro lado, se busca realizar las mejores acciones como respuesta ante fallas que interrumpirían la continuidad del proceso o incluso predecir dichas fallas, y ejecutar las acciones necesarias para evitar que ocurran. Para lograr esto, generalmente se necesita gran cantidad de personal capacitado para monitorear el gran número de variables que intervienen en los procesos, es ahí donde un sistema basado en inteligencia artificial resultaría de gran importancia, pudiendo llegar a tener mejores resultados que los operarios humanos.

En teoría, un agente de software basado en las técnicas de aprendizaje por refuerzo, sería capaz de controlar toda una planta industrial y optimizar los procesos involucrados. Por ejemplo, en [12] utilizan esta metodología para optimizar el proceso de irrigación de campos, obteniendo como resultado una disminución importante en el consumo de agua, en comparación con los métodos tradicionales de irrigación. También es notable el caso práctico en el que un agente es entrenado para controlar el flujo vehicular en una intersección [13], el objetivo es conseguir un tráfico más fluido, manteniendo la circulación en el sentido que más tráfico vehicular tenga. En [14] se implementa un sistema para la gestión de las operaciones de plantas purificadoras de agua, comprueban que el agente se puede desempeñar tan bien como los expertos, o incluso mejor, cuando el objetivo es mantener el sistema de distribución de agua potable en condiciones óptimas de operación, en este sentido, uno de los problemas que más afectan a la eficiencia en este tipo de redes se refiere a las pérdidas de agua debido a las fugas, en este caso, la detección y localización de estas averías es de suma importancia para garantizar que los niveles de pérdidas se mantengan al mínimo. En este trabajo se estudiarán técnicas basadas en aprendizaje por refuerzo para el desarrollo de algoritmos de detección y localización de fugas en las redes hidráulicas.

1.4 OBJETIVOS

1.4.1 General

- Implementar algoritmos basados en aprendizaje por refuerzo para la detección y localización de fugas en una red de distribución de agua potable.

1.4.2 Específicos

- Familiarizarse con los conceptos relacionados con redes hidráulicas, específicamente redes de distribución de agua potable.
- Familiarizarse con la paquetería WNTR siglas en inglés de, *Water Network Tool for Resilience* [15] y el programa EPANET, simuladores de redes hidráulicas.
- Estudiar los conceptos relacionados con aprendizaje automático con especial interés en aprendizaje por refuerzo.
- Crear el entorno o red hidráulica en WNTR en el que se desarrollarán las pruebas de algoritmos.
- Desarrollar la interfaz entre las variables del entorno simulado de WNTR y las librerías de aprendizaje automático en Python.
- Implementar técnicas basadas en aprendizaje por refuerzo para el control de la red hidráulica simulada.
- Comparar el sistema creado con respecto a las técnicas tradicionales usadas para el control de redes de distribución de agua potable.

1.5 METODOLOGÍA

Para abordar el problema de detección y localización de fugas, éste se considera como una tarea de clasificación, para ser más específicos, la idea principal es desarrollar un sistema capaz de identificar las diferentes situaciones en las que se puede encontrar la red hidráulica considerada, asignando a cada una de estas una etiqueta de clase. Por lo tanto, si se establece que cada una de estas etiquetas se corresponde con determinado evento de fuga en algún punto de la red, no

solo es posible detectar la fuga, sino que, además, se delimita la zona en la que ésta se encuentra.

Por lo tanto, se plantea el uso de algoritmos basados en aprendizaje por refuerzo para abordar el problema de clasificación considerado. El ambiente o entorno, genera una señal de recompensa que es entregada al agente por sus acciones, alentando a este a desarrollar determinado comportamiento, de tal manera que la recompensa que recibe sea maximizada. Aprovechando la flexibilidad de la metodología presentada, se puede establecer un modelo adecuado de agente, ambiente y recompensa acorde al problema de clasificación planteado.

Para el caso de estudio, el ambiente viene definido por todo el conjunto de variables inherentes al sistema hidráulico, que, en esta ocasión, se trata del modelo de una red hidráulica simulada con la librería WNTR. La metodología utilizada consiste en la utilización de un “Gimnasio” (en inglés *Gym*) para entrenar algoritmos basados en aprendizaje por refuerzo, Gym, es en sí un conjunto de herramientas que permite el entrenamiento de modelos bajo este paradigma del aprendizaje automático, en esencia, se encarga de modelar el ambiente o entorno con el que un agente puede interactuar. Así, siguiendo la estructura proporcionada, se implementan los métodos y rutinas necesarias para abordar el problema de detección de fugas como un problema de clasificación. En el Apéndice I se encuentran detalles sobre las clases y métodos implementados según la metodología de un Gym.

1.6 ALCANCE Y LIMITACIONES

1.6.1 Alcance

Este proyecto aborda el problema de detección y localización de fugas en redes de distribución de agua potable como un problema de clasificación, con un enfoque basado en aprendizaje por refuerzo para determinar la zona de la red en la que existe presencia de fugas. Se utilizará la librería WNTR de Python para modelar la red hidráulica considerada, junto con las herramientas disponibles en este lenguaje para aplicaciones de aprendizaje automático.

Se plantea la adaptación de un entorno de simulación que comprenda la interacción de un agente basado en aprendizaje por refuerzo, con las variables hidráulicas de una red de distribución de

agua potable, en otras palabras, se crea un Gym que representa la red hidráulica en cuestión.

1.6.2 Limitaciones

La principal limitación para el desarrollo de este trabajo de grado es el costo computacional involucrado para el entrenamiento de los algoritmos basados en aprendizaje por refuerzo. En este sentido, se debe agregar el costo por uso de hardware especializado para este tipo de aplicaciones, que se encuentran alojados en la nube, como un factor limitante.

1.7 ORGANIZACIÓN DEL DOCUMENTO

El presente trabajo de grado se encuentra organizado en cuatro capítulos, el presente capítulo que introduce aspectos preliminares al desarrollo del proyecto, el Capítulo 2 expone los conceptos principales relacionados al aprendizaje por refuerzo, tales como los Procesos de Decisión de Markov, las funciones de valoración y el algoritmo *Deep Q-Network* con las respectivas mejoras que se han desarrollado en torno a esta técnica, también se describen las principales métricas utilizadas para evaluar el desempeño en tareas de clasificación.

En el Capítulo 3 se da una breve exposición sobre los aspectos concernientes a las redes de distribución de agua potable, abarcando definiciones básicas, conceptos sobre las pérdidas de agua en las redes de este tipo, antecedentes sobre el uso de aprendizaje automático para la detección de fugas, y una breve descripción acerca de la manera en que WNTR modela las redes hidráulicas, en el Capítulo 4 se expone la metodología implementada para la detección de fugas en redes hidráulicas bajo el enfoque de aprendizaje por refuerzo, las simulaciones realizadas y los resultados obtenidos, Finalmente, se presentan las conclusiones obtenidas a partir de los resultados, y se comentan algunos puntos a mejorar para investigaciones futuras.

CAPÍTULO 2

MARCO TEÓRICO

Los algoritmos basados en aprendizaje por refuerzo pueden adaptarse para resolver gran cantidad de problemas. Por ejemplo, el sector de los vídeo juegos ha sido de gran utilidad para el desarrollo y prueba de estos modelos basados en inteligencia artificial, ya que los algoritmos creados pueden ser comparados directamente contra competidores humanos. Los ejemplos más renombrados se tienen con los programas desarrollados por DeepMind¹, creando en 2014 una inteligencia artificial capaz de jugar vídeo juegos Atari [16], y en 2015 crea AlphaGo, un programa capaz de jugar Go, clásico juego de estrategia Chino, logrando vencer a jugadores de talla profesional [17]. Incluso, dada su flexibilidad, permite abordar problemas relacionados con la clasificación y detección de anomalías [7], [8], [9], [10] temas de especial interés para el caso de estudio. El campo de conocimientos desarrollados en esta área es muy amplio, por lo tanto, en el presente capítulo se expone una descripción breve de la terminología y los conceptos esenciales de aprendizaje por refuerzo, haciendo énfasis en aquellos que se encuentran estrechamente relacionados con el problema abordado en este trabajo de grado. Se comienza con la definición formal de aprendizaje por refuerzo, describiendo los elementos que caracterizan esta rama del aprendizaje automático, se discute acerca de los Procesos de Decisión de Markov (MDP, del inglés *Markov Decision Processes*) y las funciones de valoración, hasta llegar a la definición de *Deep Q-Network* (DQN) y las diferentes técnicas que se implementan bajo este concepto.

2.1 APRENDIZAJE POR REFUERZO

Aprendizaje por refuerzo tiene como objetivo principal, descubrir secuencias de acciones que se deben realizar para maximizar una señal recompensa. En este sentido, el aprendiz o agente, no

¹Empresa Inglesa dedicada al desarrollo de inteligencia artificial, adquirida por Google en el 2014

posee conocimiento previo sobre las acciones que se deben tomar para lograr esto, en su lugar, debe explorar cuales son las acciones que generen la máxima recompensa para determinada situación, considerando que en la mayoría de las situaciones, las acciones tomadas tienen cierta influencia en las nuevas situaciones que se presenten en el entorno o ambiente, y por lo tanto en las recompensas que el agente podría tener a futuro.

La diferencia entre aprendizaje por refuerzo y otros tipos de aprendizajes que son comúnmente estudiados en el área de la inteligencia artificial, radica en la disyuntiva entre exploración y explotación. En contraste con otros tipos de aprendizaje, el aprendizaje por refuerzo se caracteriza por la existencia de una meta u objetivo, que consiste en maximizar la recompensa que se obtiene por la toma de acciones. Para lograr esto, un agente debe inclinarse por acciones que han generado recompensas altas en el pasado, sin embargo, para descubrir nuevas acciones que podrían generar mayor recompensa en el futuro, el agente debe tomar acciones que no han sido seleccionadas con anterioridad. En otras palabras, el agente debe “explotar” las experiencias previas, con la finalidad de recibir la señal de recompensa, pero además, debe “explorar” el campo de acción, de tal manera, que pueda elegir aquellas que generen mayor recompensa en el futuro [18].

Ya se han mencionado varios elementos claves que forman parte de un sistema de aprendizaje por refuerzo, sin embargo, más allá del agente, un ambiente y una señal de recompensa, podemos identificar un cuarto componente conocido como política. Una política define la forma en la que el agente aprende a tomar decisiones a lo largo del tiempo, formalmente, define una función de mapeo que transforma los estados percibidos desde el ambiente a una serie de acciones [18]. Establecer la política que seguirá un agente es de vital importancia, pues ésta, es suficiente para determinar por completo su comportamiento, una política puede estar basada en la búsqueda de valores en una tabla, donde cada posible estado representa una fila y cada acción posible se define como una columna o puede ser descrita por algoritmos más complejos.

2.1.1 Interacción del agente con su ambiente

El aprendizaje por refuerzo se basa en la interacción de un agente con su ambiente o entorno, esta interacción ocurre de forma discreta, es decir, en pasos de tiempo de la forma $t = 0, 1, 2, \dots, n$, [18]. En cada paso de tiempo, el agente recibe una observación del estado actual en el que

se encuentra el ambiente, brindando información parcial acerca de las diversas situaciones o escenarios en los que pudiera estar. Por cada observación recibida, el agente toma una acción cuya influencia sobre el ambiente lo lleva a un nuevo estado, generando una señal de recompensa que indica al agente que tan buena o mala fue la acción tomada, en el sentido de la cantidad de recompensa que podría recibir en el futuro.

2.1.2 Procesos de decisión de Markov

Los conceptos de aprendizaje por refuerzo se encuentran enmarcados en la teoría de procesos dinámicos [18], de forma más específica, en los procesos de decisión de Markov. Este tipo de procesos son generalmente de naturaleza estocástica, donde existe un número fijo de estados, que evolucionan a otros de forma aleatoria en cada paso de tiempo [4]. Estos procesos se caracterizan fundamentalmente por no tener memoria, o en otras palabras, la probabilidad de transición de un estado actual S_t a un estado futuro S_{t+1} depende sólo del par (S_t, S_{t+1}) y no de estados pasados [4].

Esta estructura fue evolucionando gradualmente hasta considerar, además, que existe un conjunto de acciones que el agente puede tomar en cada uno de los estados, afectando de esta forma, la probabilidad de transición entre estos. Finalmente, se incluye el concepto de recompensa, que se genera al transitar de un estado a otro siendo este valor positivo o negativo, de esta forma, se puede dotar de un objetivo al agente, hallar aquella política que maximice la recompensa recibida a lo largo del tiempo.

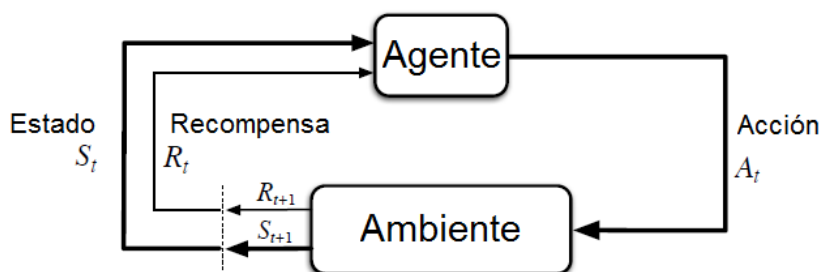


Figura 2.1. Interacción entre el agente y su ambiente en un MDP. (Imagen modificada de [18])

En la Fig. 2.1 se observa un esquema de la interacción entre el agente y su ambiente, para cada paso de tiempo, el agente recibe una observación del estado del ambiente $S_t \in S$ sobre la cual basa una acción $A_t \in A$, como consecuencia, un paso de tiempo después, el agente recibe una

recompensa numérica R_{t+1} y un nuevo estado S_{t+1} .

2.1.3 Funciones de valoración

En general, la mayoría de los algoritmos de aprendizaje por refuerzo se basan en la estimación de funciones de valor [18], que tratan de valorar “que tan bueno” es para el agente permanecer en un determinado estado, o en dado caso, tomar una acción en ese estado, aquí, el término “que tan bueno” se refiere a la futura recompensa que el agente puede obtener. La recompensa que el agente recibe depende de las acciones que éste tome, en otras palabras, depende de la política que sigue, por lo tanto, es importante destacar que las funciones de valoración se definen acorde a la política o forma de actuar del agente.

El valor de un estado, bajo cierta política, que se denota como $V(s)$, es la recompensa esperada cuando se comienza desde el estado s y se sigue dicha política, formalmente puede ser descrita de la siguiente manera.

$$V(s) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right] \quad \forall s \in S \quad (2.1)$$

Donde \mathbb{E} es el valor esperado de una variable aleatoria, t es el paso de tiempo y γ es un factor de determina la importancia que se le asigne a la recompensa futura. De forma análoga, se puede estimar el valor de tomar una acción a en un estado s , definiéndose la función $Q(s, a)$ como:

$$Q(s, a) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \mid A_t = a \right] \quad (2.2)$$

Las funciones de valoración V y Q se pueden estimar a partir de la experiencia, esto es, si un agente sigue determinada política y mantiene un promedio, para cada estado, de las recompensas obtenida en ese estado, el promedio convergerá al valor de estado $V(s)$ como el número de veces que el estado es encontrado a medida que el agente evolucione. De forma similar, manteniendo valores promedios separados para cada acción tomada en cada estado, se obtienen los valores de la función $Q(s, a)$.

Para obtener la máxima recompensa, un agente debe seleccionar la mejor acción en cada estado, sin embargo, tomar esta decisión no es trivial. Una forma de estimar el valor óptimo para cada estado, es decir, la suma de toda la recompensa que el agente puede recibir, en promedio, al

alcanzar cierto estado, es asumir que el agente actúa de forma óptima [4], de esta manera, el valor de un estado óptimo, puede ser encontrado según la ecuación de Bellman, que en su forma iterativa viene dada por.

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_k(s')] \quad (2.3)$$

Donde $T(s, a, s')$ es la probabilidad de transición del estado s al estado s' si el agente toma la acción a . $R(s, a, s')$ es la recompensa que el agente obtiene por el transito desde el estado s al estado s' tomando la acción a . De forma similar, se puede hallar una expresión para la función de valor $Q(s, a)$ como:

$$Q_{k+1}(s, a) \leftarrow \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_a Q_k(s', a)] \quad (2.4)$$

A partir de la Ec. (2.4) se puede seleccionar la mejor acción para cada uno de los estados, para ello, basta con tomar el valor Q más alto, de esta forma, asumiendo que los valores de la función $Q(s, a)$ son óptimos, se establece una política para la selección de una acción a en un estado s como:

$$\pi(s) = \operatorname{argmax}_a Q(s, a) \quad (2.5)$$

2.1.4 Aprendizaje por diferencias temporales y Q-Learning

Uno de los conceptos más importantes en aprendizaje por refuerzo trata sobre el aprendizaje por diferencias temporales, este concepto se origina de la combinación de los métodos de Monte Carlo, y las ideas de la programación dinámica. Al combinar estas dos técnicas, es posible garantizar el aprendizaje del agente sin necesidad de un conocimiento previo sobre la dinámica del ambiente, es decir, no es necesaria la existencia de la matriz de transición $T(s, a, s')$. Por lo tanto, asumiendo que los posibles estados y acciones son conocidos, el agente puede aprender directamente desde la experiencia, siguiendo cierta y determinada política, entre ellas una política de exploración aleatoria, por ejemplo.

Otro de los aspectos importantes del aprendizaje por diferencias temporales radica en una característica extraída de los métodos de programación dinámica, que consiste en la frecuencia

de actualización de los valores estimados, que, en este caso, se realizan sobre los valores previamente obtenidos, sin necesidad de esperar un resultado final de un episodio. Siguiendo esta metodología, la función de valor de estado se actualiza de la siguiente forma.

$$V_{k+1}(s) \leftarrow (1 - \alpha)V_k(s) + \alpha(r + \gamma V_k(s')) \quad (2.6)$$

Donde α es la tasa de aprendizaje. El algoritmo conocido como Q-Learning que relaciona cada estado con determinada acción, se puede obtener de forma análoga, utilizando aprendizaje por diferencias temporales, de la siguiente manera.

$$Q_{k+1}(s, a) \leftarrow (1 - \alpha)Q_k(s, a) + \alpha \left(r + \gamma \max_{a'} Q_k(s', a') \right) \quad (2.7)$$

De esta forma, la función de valor-acción, Q , convergerá directamente a una solución óptima, independientemente de la política que se siga, no obstante, esta sigue teniendo cierta influencia al determinar los pares estado-acción que son visitados, siendo el único requerimiento para asegurar la convergencia del algoritmo, que las actualizaciones del algoritmo se realicen de forma continua [18]. El algoritmo Q-Learning se muestra a continuación.

Algoritmo 2.1: Q-Learning Tabular. Adaptado de [18]

```

1 Inicializar  $Q(s, a)$ , para todo  $s \in S$   $a \in A(s)$  de forma aleatoria ;
2 for cada episodio do
3   Inicializar  $S$ ;
4   for cada paso del episodio do
5     Seleccionar una acción  $a$  desde  $S$  siguiendo una política dada por  $Q$ ;
6     Ejecuta la acción  $a$  y observa  $r, s'$  ;
7      $Q_{k+1}(s, a) \leftarrow (1 - \alpha)Q_k(s, a) + \alpha(r + \gamma \max_{a'} Q_k(s', a'))$ ;
8      $s \leftarrow s'$ ;

```

En el Algoritmo 2.1 se observa como se obtienen los valores de la función Q a lo largo de las iteraciones. Estos valores se pueden organizar de forma tabular, donde cada elemento (i, j) de la tabla representa el valor de tomar la acción a_j desde el estado s_i [19].

2.1.5 Aproximación de la función Q

El algoritmo Q-Learning actualiza en cada paso de tiempo los valores de la función Q , de esta forma, con suficientes iteraciones, se puede obtener una función Q óptima. Como se ha

mencionado, los datos que se van generando durante la ejecución del algoritmo, esto es, los posibles estados y el valor de las acciones, se organizan en una Tabla, conocida como Tabla Q [19], en ésta, cada fila representa un estado distinto y cada columna el valor de las acciones, convirtiéndose la Tabla Q en un mapa de los diversos estados en los que un agente puede transitar.

La desventaja de usar una tabla como función de mapeo, es que se vuelve impracticable con el aumento de los posibles estados y acciones. La solución para abordar este problema radica en el uso de una función de aproximación, de tal manera que los valores de la función Q puedan ser estimados con un número manejable de parámetros [4].

Existen muchas formas de aproximar la función Q , por ejemplo, combinaciones lineales de características, arboles de decisión, técnicas basadas en transformadas de Fourier, etc [20]. Sin embargo, con los recientes avances en aprendizaje profundo (*Deep Learning*) se ha demostrado que el uso de redes neuronales profundas (*Deep Neural Networks*) pueden funcionar mucho mejor que los métodos de aproximación ya mencionados. Así, al usar una red neuronal para aproximar la función Q , esta adquiere la denominación de *Deep Q-Network* (DQN), creando una nueva técnica de aprendizaje denominada *Deep Q-Learning*.

2.1.6 Redes neuronales como función de aproximación.

El objetivo principal de la mayoría de los algoritmos de aprendizaje por refuerzo, se basa en la estimación de los valores óptimos de la función Q , de esta forma, para cada estado presentado se puede obtener la acción que genera la mayor cantidad de recompensa. Además, considerando que para procesos con gran cantidad de estados, es necesario utilizar una función de aproximación para estimar los valores de dicha función. El algoritmo DQN, caracterizado por implementar redes neuronales artificiales² como función de mapeo, se presenta como herramienta principal para alcanzar este objetivo.

Las redes neuronales son estructuras que intentan imitar el comportamiento del sistema nervioso de los seres vivos, en este sentido, su componente principal, las neuronas, serían unidades de procesamiento idénticas, que se ordenan de acuerdo a una estructura jerárquica compuesta por

²De ahora en adelante se omite el término “artificial”, quedando simplemente como redes neuronales

capas, distinguiéndose tres niveles, la primera capa o capa de entrada, una capa intermedia o capa oculta y una capa de salida.

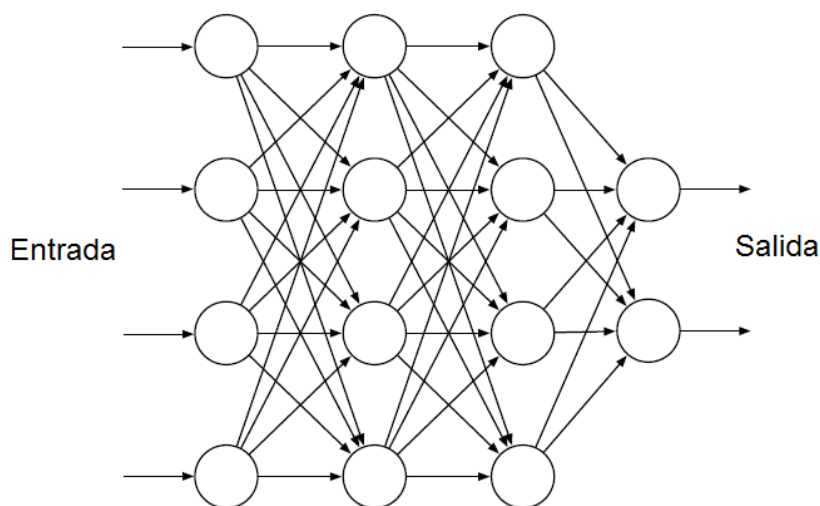


Figura 2.2. Arquitectura típica de una red neuronal. (Imagen modificada de [18])

En la Fig. 2.2 se puede observar la arquitectura básica de una red neuronal, las neuronas en cada una de las capas se encuentran conectadas a través de conexiones ponderadas, que sirven de camino para propagar los datos desde la capa de entrada hasta la capa de salida, pasando a través de las capas ocultas. Cada una de las neuronas se encuentran definidas por una función de activación de tipo no lineal³, que procesa la suma ponderada de los datos de entrada y generan una salida o activación, que servirá como entrada a las neuronas de la siguiente capa, o en dado caso, la salida propia de la red.

El proceso de aprendizaje o entrenamiento de una red neuronal, consiste en la obtención de los pesos asociados a las conexiones entre neuronas, de tal manera que se ajuste a los requerimientos de un problema determinado. Las tareas de aproximación generalmente se basan en la minimización de una función de error o de pérdida, que es la diferencia entre la salida obtenida de la red y el valor deseado, bajo el enfoque de aprendizaje por refuerzo, el entrenamiento de una red neuronal de parámetros θ que modela la función Q , se puede realizar de la siguiente manera.

$$L_i(\theta_i) = (y_i - Q(s, a; \theta_i)) \quad (2.8)$$

³Si es lineal, siempre se encontrará una red de una capa, equivalente a otra con cualquier número de capas.

Donde $y_i = r + \gamma \max_{a'} Q(s', a'; \theta_{i-1})$ es la ecuación de Bellman para la iteración i , en este caso la función objetivo. Los parámetros θ_{i-1} se fijan cuando se optimiza la función $L_i(\theta_i)$.

2.1.7 Mejoras del algoritmo DQN

A diferencia de las técnicas basadas en aprendizaje supervisado, donde los valores o función objetivo son conocidos de antemano, en la Ec. (2.8) se aprecia que la función objetivo depende de los parámetros que se intentan hallar, es decir, de los pesos de la red θ , esta particularidad produce cierta inestabilidad durante la etapa de aprendizaje del agente [16], para intentar solventar esta desventaja se han desarrollado ciertas técnicas que permiten un mejor desempeño del algoritmo DQN.

2.1.7.1. Uso de memoria y técnicas de muestreo

Una de las técnicas ampliamente utilizadas para acelerar y estabilizar el aprendizaje del agente utilizando el algoritmo DQN [16], consiste en utilizar una memoria de experiencias⁴ (en inglés, *Experiencie Replay*) donde se almacenan las diferentes transiciones (s, a, r, s') generadas por la interacción del agente con su entorno en cada paso de tiempo. De esta forma, al utilizar transiciones o experiencias pasadas junto con las experiencias más recientes se elimina cualquier correlación temporal que exista en los datos utilizados para el entrenamiento de la red. Así, durante un paso de entrenamiento los datos provendrían de diferentes episodios evitando que se produzcan sesgos durante la etapa de aprendizaje, reduciendo la posibilidad de caer en un mínimo local.

El entrenamiento o ajuste de los pesos de la red neuronal se realiza minimizando la Ec. (2.8), mediante el uso de un subconjunto de datos (*mini batch*) que se obtienen a través de un proceso de muestreo aleatorio de las transiciones almacenadas en la memoria del agente. De esta forma, se elimina la posibilidad de que se produzca algún sesgo durante el entrenamiento, pues al ser un muestreo aleatorio, todas las transiciones en la memoria tienen la misma probabilidad de ser seleccionadas, sin embargo, este proceso es poco eficiente, pues aquellas experiencias que brindan más información al proceso de aprendizaje pudieran usarse más esporádicamente [21].

Para optimizar el proceso de muestreo, en [22] proponen una metodología que consiste en

⁴También se conoce como memoria de reproducción (en inglés, *Memory Replay*)

utilizar aquellas transiciones que brinden más información durante el proceso de entrenamiento, de esta manera, existirán muestras prioritarias en el subconjunto de entrenamiento⁵. El criterio para establecer cuál transición brinda más información, se refiere a la magnitud del error de la Ec. (2.8), denotado como δ , en este caso, si la diferencia entre el valor objetivo y el valor esperado es relativamente grande, significa que el agente no se había adaptado a esta, por lo tanto se puede aprender más de ella. Formalmente, la probabilidad de tomar una muestra i desde la memoria del agente se define como:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \quad (2.9)$$

Donde p_i es la prioridad correspondiente a la transición i , e indica, según el criterio basado en la magnitud del error δ , que tan importante es ésta para el aprendizaje del agente. El exponente α determina la importancia de la priorización, al introducir cierto grado de aleatoriedad al proceso de muestreo, por ejemplo, con $\alpha = 0$ se tiene el caso uniforme. De forma contradictoria, el problema con esta técnica es que genera un sesgo en la selección de las transiciones, pues ahora las experiencias más priorizadas tendrán mayor probabilidad de ser seleccionadas. Para corregir este problema se utiliza una técnica que consiste en modificar el proceso de actualización de los pesos, reduciendo la magnitud de aquellos correspondientes a las muestras que se ven más frecuentemente, se conoce como *Importance sampling weights* [22]. Los pesos se actualizan según la siguiente expresión.

$$\omega_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)} \right)^\beta \quad (2.10)$$

Donde N es el tamaño de la memoria de almacenamiento y β un parámetro que controla la importancia de la actualización de los pesos que se hacen de esta manera.

2.1.7.2. DQN con red objetivo separada

Otra de las mejoras implementadas para estabilizar el proceso de aprendizaje y acelerar el entrenamiento del agente, consiste en la utilización de una red neuronal diferente para estimar el valor objetivo y_i de la Ec. (2.8). Esta red neuronal, de parámetros ϕ sigue a los pesos de la red Q_θ con cierto retraso, es decir, cada cierta cantidad de pasos los pesos de la red θ son copiados a la

⁵En inglés la técnica se conoce como *Prioritized Experience Replay* traducido como “Repetición de Experiencia Priorizada.”

red ϕ . La estimación de los valores objetivos con una versión anterior de la red neuronal que se utiliza para obtener los valores Q , provoca cierto retraso en el tiempo que evita las oscilaciones durante el aprendizaje y reduce el riesgo de divergencia del algoritmo [16], los valores objetivos se obtienen de la siguiente manera.

$$y_i = r + \max_a \gamma Q^\phi(s', a) \quad (2.11)$$

Donde Q^ϕ es la función Q modelada por la red auxiliar de parámetros ϕ , la frecuencia de actualización de los pesos de la red Q^ϕ depende de cada problema en particular.

2.1.7.3. Doble DQN

Una de las desventajas del algoritmo DQN es la sobre estimación de los valores de la función Q , como consecuencia de seleccionar siempre los valores máximos de esta función. Para reducir el sesgo producido por esta acción se utilizan dos estimadores para la función Q [23] obteniendo los valores de la función objetivo según la Ec. (2.12).

$$y = r + \gamma Q^\phi(s', \operatorname{argm\acute{a}x}_a Q^\theta(s', a)) \quad (2.12)$$

Al estimar el valor objetivo de esta forma, la acción se toma considerando el máximo valor de Q^θ , sin embargo, su valor se obtiene de la red Q^ϕ , por lo tanto, las acciones no siempre se corresponderán con el valor máximo de la función Q objetivo.

2.1.7.4. Dueling DQN

Esta técnica propuesta en [24] consiste en descomponer la función $Q(s, a)$ en el par de funciones $V(s)$ y $A(s, a)$, que representan el valor de estar en determinado estado, y la “ventaja” de tomar una acción en ese estado, respectivamente. La descomposición se realiza cambiando la arquitectura de la red neuronal que modela el clásico algoritmo DQN, creando dos caminos, cada uno de ellos caracterizado por diferentes parámetros. Finalmente, ambos caminos se combinan para obtener los valores de la función Q representada en la Ec. (2.13).

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(A(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta, \alpha) \right) \quad (2.13)$$

Donde θ hace referencia a los parámetros de la sección de red neuronal que ambas funciones

tienen en común, α a los parámetros del camino seguido por la función $A(s, a)$ y β los parámetros del camino correspondiente a la función de valor $V(s)$. La arquitectura presentada ha demostrado ser eficaz cuando el conjunto de posibles acciones que el agente puede tomar es amplio [24].

2.1.8 Evaluación de desempeño

Para evaluar el desempeño de un modelo clasificador es necesario establecer las métricas necesarias que aporten información suficiente para valorar el comportamiento del modelo, una de estas métricas es la exactitud (en inglés *Accuracy*), definida como la razón entre el número de instancias clasificadas correctamente y el total de instancias en el problema considerado. Sin embargo, al trabajar sobre conjuntos de datos no balanceados, donde el número de ejemplos de cada clase es diferente, la exactitud obtenida no sería un indicador confiable del desempeño del modelo, ya que no revelaría posibles sesgos hacia las clases mayoritarias.

Para problemas de clasificación multiclase, donde el número de instancias por clase es diferente, la mejor forma de valorar el desempeño de un modelo clasificador es a través de una matriz de confusión [4]. En ésta, se puede visualizar de forma clara las clases reales, representadas en cada una de las filas, y las clases que predice el sistema, que se corresponden con cada una de las columnas. Finalmente, la matriz de confusión, es de dimensión $n \times n$, donde n es el número de clases del problema considerado.

Desde la matriz de confusión se puede extraer gran cantidad de información, las métricas más comunes para evaluar problemas de clasificación multiclase son la *Precision*, *Recall* y *F_{score}*. La *Precision*, es la razón de casos positivos correctamente clasificados, indica la capacidad del modelo para no clasificar como positiva una clase que es negativa. El *Recall*, también conocido como sensibilidad, mide la proporción de casos que fueron correctamente identificados, formalmente estas métricas se definen de la siguiente manera.

$$Precision = \frac{TP}{TP + FP} \quad (2.14)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.15)$$

Donde TP, FP, FN son los tipos de errores (en inglés *True Positive*, *False Positive* y *False Negative*) verdaderos positivos, falsos positivos y falsos negativos. Para obtener una mejor

evaluación del desempeño del modelo, *Precision* y *Recall* se combinan para estimar el F_{score} , que es la media armónica entre estas dos métricas. Se define de la siguiente manera.

$$F_{score} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (2.16)$$

El valor F_{score} será un valor alto si ambas métricas, *Recall* y *Precision* son altas, en este caso, la valoración que recibe el modelo es positiva.

www.bdigital.ula.ve

CAPÍTULO 3

REDES DE DISTRIBUCIÓN DE AGUA POTABLE

Una red de distribución de agua potable es un sistema que tiene como objetivo transportar el agua desde los puntos de captación y tratamiento hasta los usuarios conectados a la red, manteniendo el agua en condiciones que permita su uso en un ambiente doméstico residencial, siendo también esencial en el mantenimiento de las actividades comerciales e industriales de cualquier ciudad [25]. Llevar a cabo esta tarea requiere de la operación conjunta de gran cantidad de elementos que componen este tipo de redes. Estos, pueden ser aquellos relacionados a la infraestructura, tales como tuberías, válvulas, bombas, etc. o elementos que cuantifican el grado de operatividad, indicadores de pérdidas técnicas y comerciales, calidad del agua e impacto ambiental. Uno de los problemas que más afecta la eficiencia de las redes de este tipo, son las pérdidas de agua que se generan a lo largo de todo el sistema hidráulico. Como se mencionó anteriormente, el presente trabajo aborda el problema de detección de fugas usando las herramientas desarrolladas en el área de la inteligencia artificial, específicamente aprendizaje por refuerzo.

3.1 SISTEMAS DE DISTRIBUCIÓN

Las redes de distribución de agua potable son una colección de tuberías de diferentes longitudes y diámetros que se encuentran conectadas entre sí mediante nodos o uniones a lo largo de toda la extensión de la red, sumado a elementos de maniobra y control que permiten el suministro de agua a los abonados de la red.

Los sistemas de distribución de agua potable, generalmente son suplidas por fuentes de agua ubicadas en una posición de mayor altura respecto a la ciudad, de esta forma, la presión necesaria para mantener el agua circulando por las tuberías principales, ya sea para uso doméstico o

de extinción de incendios, es generada gracias al efecto de la gravedad, de otra forma, sería necesario el uso de bombas para elevar la presión del sistema y distribuir el agua a lo largo de la ciudad.

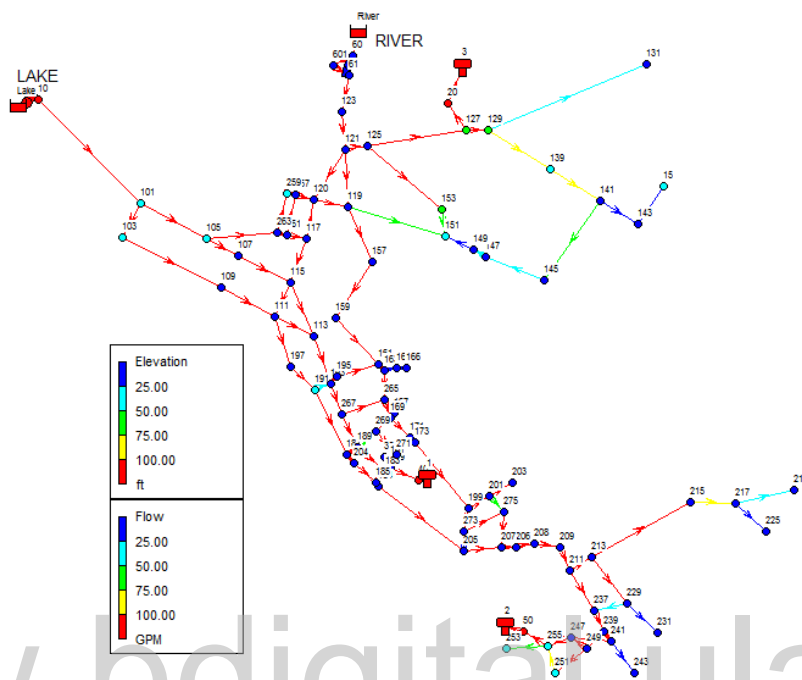


Figura 3.1. Red de prueba Net3 de EPANET.

En la Fig. 3.1 se aprecia una red de prueba extraída de la documentación de EPANET¹, en ella se puede observar la distribución de los elementos a lo largo de la red, los nodos o uniones (puntos en la figura) conectan a las tuberías que transportan el agua en distintas direcciones, se observa también, como la red está siendo alimentada por dos fuentes de agua, en este caso, un río y un lago.

3.1.1 Relación entre presión y caudal

En una red de distribución de agua potable las variables que caracterizan el comportamiento del sistema son la presión y el caudal de agua en las tuberías, la relación entre estas dos variables es muy importante para entender el funcionamiento de las redes hidráulicas, además, se encuentran estrechamente relacionadas con la generación de fugas en las tuberías. Cuando en una red de

¹Software para el análisis de redes de distribución de agua potable, desarrollados por la Agencia de Protección Ambiental de los Estados Unidos

distribución la demanda de agua se reduce al mínimo, por ejemplo, una demanda igual a cero², la presión en cualquier punto de la red sería simplemente la diferencia entre la altura de la superficie de la fuente de agua y el punto mismo, esta presión se conoce como presión estática [11].

No obstante, si existe una abertura en algún punto de la red, sea esta producto de una conexión de algún usuario o fuga en algún punto de la misma, la presión hará que el agua tienda a salir por la abertura, es de esta forma como los usuarios pueden obtener el vital líquido y a su vez, se generen pérdidas como resultado de las salidas no deseadas, es decir, las fugas. El contacto del agua en circulación con las paredes de la tubería genera una pérdida de energía producto de la fricción que se produce, estas pérdidas se traducen en caídas de presión en cualquier punto de la red [11], como consecuencia, se deduce que durante los periodos de máxima demanda la presión en el sistema es mínima, mientras que al disminuir, la presión es máxima. Por ejemplo, en horas de la madrugada, la demanda de agua se reduce al mínimo, como consecuencia, la presión en las tuberías tiende a incrementarse.

3.2 PÉRDIDAS DE AGUA EN LAS REDES DE DISTRIBUCIÓN

En las redes de distribución de agua potable, los problemas relacionados con las fugas de agua, son las principales causas que afectan la eficiencia del proceso de distribución del vital líquido. Las fugas, son pérdidas de agua que ocurren con mayor frecuencia en determinados puntos de la red, como uniones de tuberías, codos, roturas de conductos y válvulas [26], aunque igualmente, pueden ocurrir en secciones de tuberías ya sea por aplastamiento o corrosión de las mismas. Este tipo de problemas es frecuente en sistemas a presión, y las pérdidas pueden llegar a ser tan altas, que el efecto, en cuanto al costo económico, puede llegar a ser considerable.

Determinar la causa de una fuga en algún punto de la red puede llegar a ser una tarea imposible, sin embargo, según [27] se pueden encontrar diferentes categorías para agrupar los factores que aumentan la tasa de probabilidad de ocurrencia de fugas, en general, estos factores pueden ser de tipo físico, medio ambiental o relacionados con la operación del sistema.

Entre los factores físicos se encuentran características propias de las tuberías, tales como, el tipo

²Situación hipotética pues en una red de este tipo siempre existirán pérdidas.

de material, espesor de la pared, diámetros, longitud, tipo de uniones, revestimientos usados, entre otros. elementos tales como el tipo de suelo, agua subterránea, el clima, localización geográfica de la tubería y la actividad sísmica se corresponden con factores de tipo medio ambiental. Finalmente, la presión interna del agua, velocidad del flujo, calidad del agua, y en general, prácticas que comprometen la integridad estructural y la calidad del agua son considerados como factores operativos, todos estos factores aumentan la probabilidad de ocurrencia de fuga en las tuberías que se encuentran a lo largo de la red.

3.3 DETECCIÓN DE FUGAS

La detección temprana de fugas es de vital importancia para mantener la operatividad de la red en niveles óptimos, para esto, es fundamental establecer un control activo de pérdidas en toda la red. Según [11] se pueden definir dos tipos de pérdidas, las pérdidas reales, que serían las fugas propiamente dichas y las pérdidas aparentes, en este caso, se conforman por las producidas por consumidores no autorizados (conexiones ilegales) y las inexactitudes o errores de medición, no obstante, también se pueden originar por errores de tipo administrativo.

La detección de las pérdidas reales, es decir, de fugas en las redes de distribución, aumenta de complejidad según el tamaño o dimensión de ésta, en [28] clasifican las fugas según el caudal que se pierde y la duración de las mismas en tres categorías.

- Fugas reportadas o visibles.
- Fugas no reportadas u ocultas.
- Fugas de fondo o tipo goteo.

Las fugas reportadas son aquellas que se generan debido a explosiones repentinas o rupturas de uniones de tuberías de grandes diámetros, el agua que se pierde se observa rápidamente en forma de charcos, dependiendo de factores como el tipo de suelo, presión en la tubería, y la dimensión del orificio de la fuga. Si las fugas tienen caudales mayores a 250 l/h a 50 m de presión, pero que por diversas razones no son visibles desde la superficie, se denominan fugas ocultas, estas son detectadas observando comportamientos anómalos en los patrones de consumo dentro de una zona de suministro de agua.

Finalmente, las fugas que comprenden caudales menores a 250 l/h a 50 m de presión son consideradas fugas de fondo o de tipo goteo, tienden a generarse por filtraciones en uniones y válvulas. Este tipo de fugas tiene una baja tasa de detección por lo que a largo plazo generan pérdidas considerables, en este sentido, el tamaño de la fuga es una característica importante a tener cuenta en el proceso de detección, pues dependiendo de éste, el tiempo de localización y reparación será mayor o menor, teniendo impacto directo sobre la cantidad de agua que se pierde debido a la fuga. En el siguiente gráfico se observa la comparación entre dos eventos de fugas de mayor y menor magnitud.

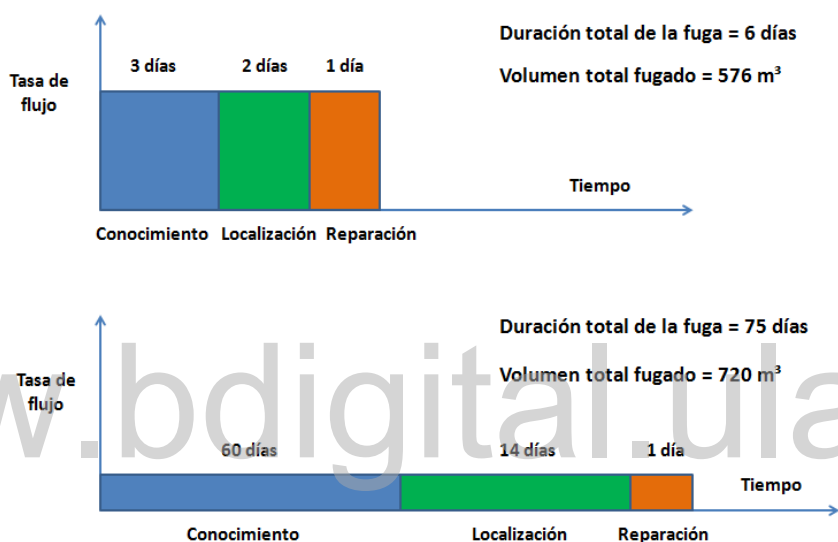


Figura 3.2. Relación entre el caudal perdido en una fuga en función de la duración de la misma [2])

En la Fig. 3.2 podemos apreciar como una fuga de mayor magnitud, a pesar de tener una tasa de flujo (caudal) considerable, desde la detección, es decir, desde el conocimiento de la existencia de la misma, hasta su reparación, genera un volumen de agua perdido menor que el volumen perdido a causa de una fuga con una tasa de flujo reducida, es decir, de menor magnitud. Esto es como consecuencia de los tiempos involucrados en el proceso de detección, localización y finalmente, reparación de la avería, ya que los efectos de las fugas de mayor magnitud son más evidentes, la rapidez con la que se puede actuar para frenar la pérdida es mayor que con fugas más reducidas, en estos casos los efectos no son tan pronunciados y el tiempo invertido para repararlas, cuando se detectan, es mayor.

3.4 APRENDIZAJE SUPERVISADO PARA DETECCIÓN DE FUGAS

La detección de fugas mediante el uso de algoritmos basados en aprendizaje automático, consiste, en la mayoría de los casos, en la implementación de clasificadores entrenados bajo la metodología de aprendizaje supervisado. En este tipo de algoritmos los datos o patrones de entrada se encuentran acompañados con la salida o respuesta deseada, en otras palabras, se trata de datos etiquetados [4], de esta forma, el ajuste de los parámetros del clasificador se realiza de tal manera que la diferencia entre la entrada y la salida deseada sea la menor posible.

La metodología más común para implementar este tipo de algoritmos para la detección de fugas, consiste en utilizar un modelo de la red hidráulica en cuestión, para generar, mediante simulaciones de diferentes escenarios de falla, una base de datos lo suficientemente representativa del comportamiento de la red ante diversas situaciones de fuga, cada instancia de la base de datos generada se acompaña con su etiqueta de clase, de esta manera, se entrena un clasificador que distinga entre los diferentes escenarios presentados, y además, tenga suficiente capacidad de generalización para operar con datos que no fueron presentados durante la etapa de entrenamiento.

Uno de los clasificadores de uso más extendido consiste en arquitecturas basadas en redes neuronales, éstas han ganado popularidad en los últimos años y tienen aplicación en gran diversidad de áreas, por ejemplo, en [6] comparan el desempeño de diferentes arquitecturas de redes neuronales en la tarea de detección de fugas, en [5], se compara el desempeño de diferentes algoritmos característicos del aprendizaje automático, demostrando excelentes resultados en cuanto a exactitud en la clasificación.

3.5 MODELO DE REDES HIDRÁULICAS

Disponer de un modelo representativo de las redes de distribución de agua potable, permiten realizar gran cantidad de estudios y análisis sobre el comportamiento de la red en cuestión ante gran diversidad de escenarios. En este sentido, existen herramientas de simulación que permiten crear, simular y analizar redes de este tipo de forma relativamente sencilla, incluyendo elementos tales como uniones, tanques, reservorios, tuberías, bombas, y válvulas, de tal manera que los modelos generados son lo suficientemente aproximados de las redes reales. Para ello es necesario

resolver las ecuaciones hidráulicas que rigen el sistema, estas, determinan el comportamiento de todas las variables en el modelo hidráulico.

3.5.1 Pérdidas de presión en las tuberías

Las pérdidas de presión en la tuberías son producidas por la fricción que se genera entre las paredes de la tubería y el flujo de agua, WNTR calcula estas pérdidas según la fórmula Hazen-Williams, que se define de la siguiente manera.

$$H_{n_j} - H_{n_i} = h_L = 10,667 C^{-1,852} d^{-4,871} L q^{1,852} \quad (3.1)$$

Donde h_L es la pérdida de presión en metros, C es el coeficiente de rugosidad de la tubería (sin dimensiones), d es el diámetro de la tubería en metros, L es la longitud de la tubería en metros, q es la tasa de flujo en (m^3/s), H_{n_j} y H_{n_i} son las presiones en los nodos extremos de la tubería en metros.

3.5.2 Modelo de fugas

Las fugas de agua pueden ser modeladas en tuberías y nodos, el caudal que se pierde en una fuga es proporcional al área del orificio (considerado circular), y a la presión en el punto de fuga. WNTR utiliza el modelo propuesto en [29] donde el flujo a través del agujero viene dado por.

$$Q_{leak} = C_d A p^\alpha \sqrt{\frac{2}{\rho}} \quad (3.2)$$

Donde Q_{leak} es la demanda de la fuga en (m^3/s), C_d es el coeficiente de descarga (sin dimensiones) y se encuentra comprendido entre 0 y 1, A es el área del orificio en (m^2), p es la presión en el interior de la tubería en Pascales, α es el exponente de fugas, su valor es de 0,5, y ρ es la densidad del fluido.

En la Fig. 3.3 se puede apreciar la demanda de fugas de diferentes dimensiones en función de la presión, como se observa el aumento de la presión y del área del orificio tienen como consecuencia un aumento en el agua que se pierde a través de la fuga.

Al disponer de un modelo matemático del comportamiento de las redes hidráulicas, y que de

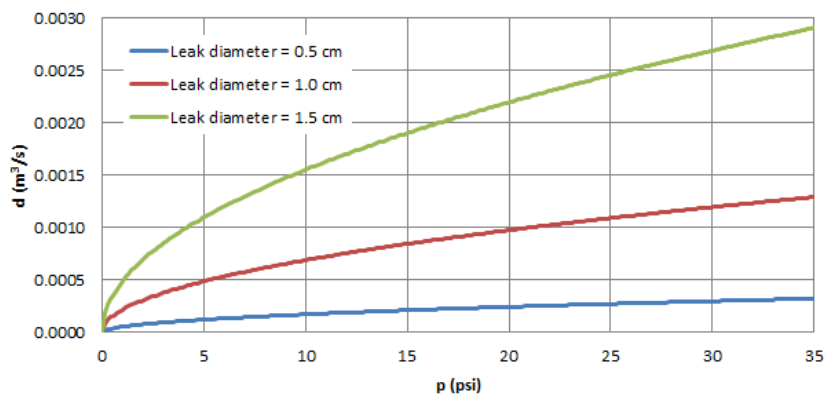


Figura 3.3. Relación entre la presión y la demanda de la fuga. (Imagen extraída de [15])

forma más específica, describe los efectos de las fallas que pueden suceder en estas. Como por ejemplo, fugas en tuberías o nodos, y que además, permiten cierto grado de ajuste sobre los parámetros del modelo. Es posible estudiar mecanismos de control de la red sobre un ambiente simulado, sin que estos pierdan generalidad para ser aplicados en situaciones reales. Por ejemplo, mecanismos para detectar fugas en redes hidráulicas desarrollados a partir de estos modelos, son, con las respectivas consideraciones, formas viables de abordar el problema en una red real.

CAPÍTULO 4

DETECCIÓN Y LOCALIZACIÓN DE FUGAS

En este capítulo se describe la metodología implementada para detectar y localizar fugas en redes de distribución de agua potable, con un enfoque basado en aprendizaje por refuerzo. Se plantea el problema de detección y localización de fugas como un problema de clasificación, en este sentido, para reducir el número de clases, la red es sectorizada en zonas de fuga, para ello, la red hidráulica es considerada como un grafo conectado. Se presentan los detalles referentes al ambiente de simulación de la red hidráulica con el que un agente basado en aprendizaje por refuerzo interactúa. Finalmente, se estudia el desempeño del agente cuando la tarea de clasificación implica, además, un proceso de selección de características.

4.1 DETECCIÓN Y LOCALIZACIÓN DE FUGAS COMO UN PROBLEMA DE CLASIFICACIÓN

Para implementar un sistema capaz de detectar y localizar fugas en redes de distribución de agua potable este se plantea como un problema de clasificación, donde cada situación o evento de fuga en algún punto de la red represente una clase categórica en el problema considerado.

4.1.1 Residuos de presión

La lectura regular de variables físicas tales como presión y flujo, además de ser herramienta para la calibración de los modelos matemáticos de las redes hidráulicas, sirve también para detectar cualquier anomalía en el comportamiento de la red. Sin embargo, debido a la complejidad de estos sistemas, con patrones de consumo que varían a lo largo del tiempo, y por lo tanto, valores de presión y flujo cambiando constantemente, establecer si determinado comportamiento es anómalo o no, a partir de las mediciones realizadas directamente de la red, es una tarea complicada.

Para determinar si los cambios que ocurren en las variables de una red hidráulica son productos de fallas, por ejemplo, fugas de agua, es necesario comparar las mediciones con un estado que se pudiera considerar base. En otras palabras, al disponer de un histórico de datos de la red en cuestión, se puede establecer si existe o no anomalías en la red mediante la comparación entre este histórico y las mediciones que se obtienen del monitoreo de la red.

En este sentido, partiendo del modelo computacional de la red hidráulica, se establece que cada instancia o ejemplo que representa una situación particular en la que se encuentra la red, se corresponde con la diferencia de presión en los nodos, entre una situación de operación normal, donde se considera que no hay fuga, y otra en la que se simula una fuga en algún nodo de la red. Además, uno de los efectos más remarcados en los sistemas hidráulicos que producen las pérdidas de agua, son las caídas de presión en las cercanías de la abertura u orificio de la fuga. Por lo tanto, al monitorear el comportamiento de la presión en los nodos de la red, se puede detectar comportamientos anómalos correspondientes con eventos de fuga que ocurren a lo largo de la red.

4.2 SECTORIZACIÓN

La sectorización de redes hidráulicas consiste en la creación de porciones de red, delimitadas geográficamente dentro de la misma. En general, se realiza para tener mayor control sobre aspectos como consumo y calidad del agua, detección de fugas, entre otros [30].

En el problema de detección y localización de fugas, manejar la red por sectores permite la reducción significativa de clases concernientes al problema de clasificación, en este sentido, al considerar un área delimitada dentro de la red, como zona en la cual se desea conocer si existe o no una fuga, se abarca gran cantidad de elementos, es decir, de nodos o tuberías en los que una falla puede ocurrir. De esta forma, se establece que cada sector en la red se corresponde con una posible zona de fuga.

4.2.1 Red como un grafo

Existe gran variedad de criterios en los que se puede basar la sectorización de la red, por ejemplo, en función de las características físicas que comparten los elementos de la red, tales como el material, diámetro, y longitud de las tuberías o factores geográficos, como la elevación o altura respecto al nivel del mar, aspecto que tiene relación directa con la presión del sistema, otro criterio sería basarse en la analogía de la red hidráulica con un grafo, aprovechando todas las propiedades y características de estos para establecer zonas de fuga bien definidas.

Un grafo, denotado como $G(V, E)$, es una colección de puntos denominados vértices V unidos por líneas o conexiones llamadas aristas E , en analogía con una red hidráulica, los nodos o uniones de estas se corresponden con los vértices, mientras que las aristas serían las tuberías que interconectan a los nodos.

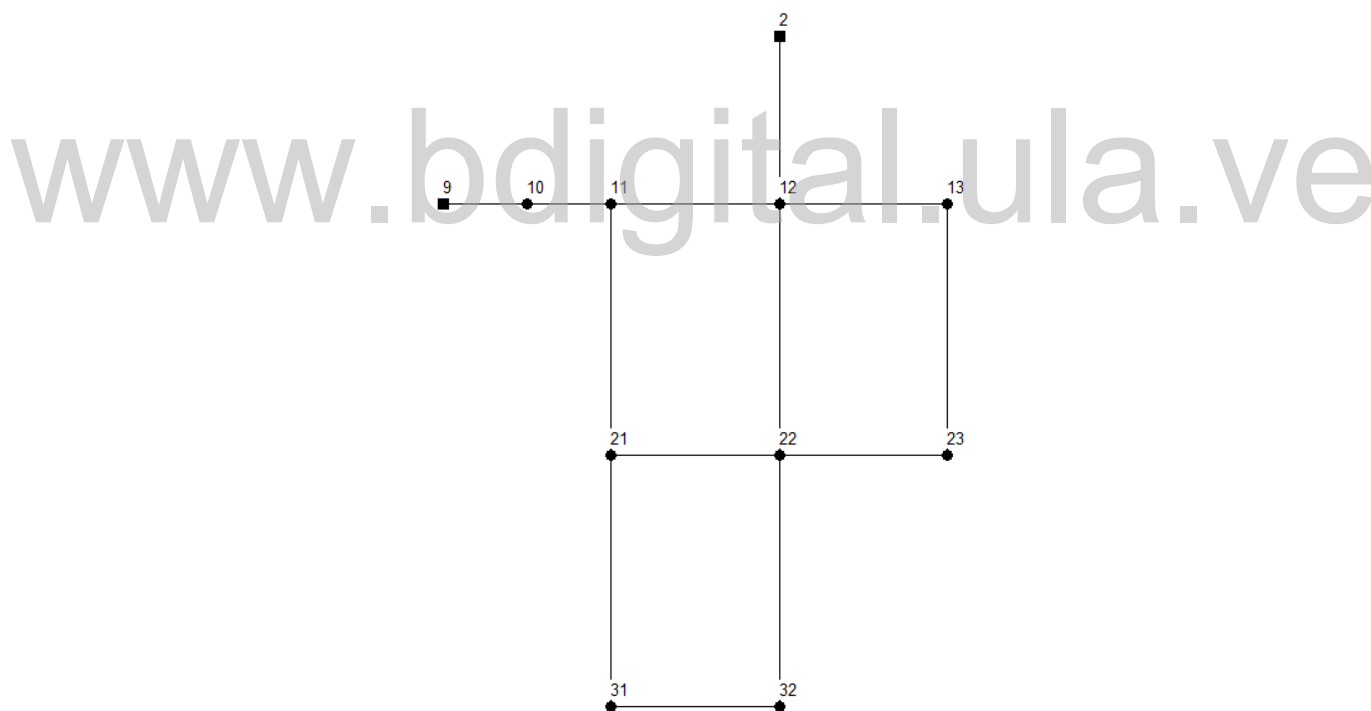


Figura 4.1. Red de prueba Net1 de EPANET.

En la Figura 4.1 podemos observar la topología de una de las redes experimentales proporcionados en la documentación de EPANET, se ha omitido la simbología de los elementos para facilitar su comparación con un grafo. La red está formada por 11 vértices representando los nodos o uniones y 13 aristas, que modelan las tuberías y bombas en la red.

4.2.1.1. Propiedades de los grafos.

Los grafos poseen ciertas características y propiedades que se pueden utilizar para crear modelos más representativos de las redes hidráulicas. Para modelar una red de tuberías acorde con las características físicas referentes a la longitud de estas, es necesario emplear el concepto de grafo ponderado, que se define a continuación.

Definición 1 *Un grafo, se denomina grafo ponderado, denotado como $G(V, E, W)$, si a cada arista que conecta el par de vértices $\{v_i, v_j\}$ de G se asigna un número no negativo $\omega_{ij} \in W$ denominado peso o longitud. [5]*

Así, las conexiones entre los vértices pueden relacionarse con un peso, costo o longitud. Para el caso específico de las redes hidráulicas, se asocia con la longitud de las tuberías, es decir la distancia existente entre dos nodos adyacentes. Establecido este valor, se puede hablar entonces del concepto de vecinos más cercanos, formalmente se define de la siguiente manera.

Definición 2 *Dado un grafo ponderado $G(V, E, W)$ donde V representa el conjunto de vértices en el grafo, E el conjunto de aristas y $\omega_{ij} \in W$ la distancia entre el par de nodos adyacentes $\{v_i, v_j\}$, se dice que v_j es el vecino más cercano de v_i si y sólo si $\omega_{ij} < \omega_{ik}$ con $v_k \in V$ [5]*

Finalmente, otra propiedad de la que se puede hablar trata sobre los conjuntos dominantes en un grafo, cuya definición se da a continuación.

Definición 3 *Un conjunto dominante de un grafo $G(V, E)$ es un subconjunto $V' \in V$ tal que los vértices que no pertenecen a V' son adyacentes a al menos un vértice de V' [5]*

En [5] proponen una metodología que consiste en la sectorización de la red mediante el uso del concepto de conjunto dominante de un grafo. El objetivo es tratar de ubicar aquellos nodos que forman un conjunto dominante para la colocación de sensores, y de esta forma establecer zonas de fuga en la red. El algoritmo implementado es el siguiente.

Algoritmo 4.1: Sectorización de la Red. Adaptado de [5].

```

1 Entrada: Grafo ponderado  $G(V, E, W)$ ;
2 Salida: Conjunto de nodos que conforman las diferentes zonas de fuga.;
3 for  $l = 1 \dots |V|$  do
4   | Selecciona un vértice  $v_l \in V$ ;
5   | Encontrar los vecinos más cercanos de  $v_l$  y agregar a un vector  $n$ .;
6 Si la frecuencia de ocurrencia de cada elemento en  $n$  es  $\geq 2$  agregar al conjunto  $A$ ;
7 Encontrar vecinos de cada elemento de  $A$  y almacenar en un conjunto  $P$ ;
8 Construir un conjunto de nodos excluidos  $L$  tal que  $L = V \cap P$ ;
9 Construir un nuevo conjunto  $S$  tal que  $S = L \cap A$ ;
10 for  $i = 1 \dots |S|$  do
11   | for  $s_i \in S$  do
12   |   | Hallar el vecino en  $S - \{s_i\}$  y agregarlo al conjunto  $R$ ;
13   |   | if  $s_i \in R$  then
14   |   |   | if Un vecino de  $s_i \in R$  then
15   |   |   |   | Remover  $s_i$  de  $S$ ;
16   |   | else
17   |   |   |  $i \leftarrow i + 1$ 
18 Se asigna cada nodo sin sensor al nodo con sensor más cercano y se incluye en el
    conjunto  $M$ ;
19 return  $S, M$ ;

```

Del Algoritmo 4.1 se obtienen dos conjuntos S y M , el primero contiene los nodos que conforman un conjunto dominante en la red hidráulica, mientras que el segundo contiene los nodos más cercanos a los nodos del primer conjunto, de esta manera, se define como zona de fuga, cada grupo de nodos formados por un nodo con sensor y sus vecinos más cercanos.

4.2.2 Red hidráulica considerada

Para desarrollar el esquema de detección y localización de fugas se utiliza la red hidráulica número tres de la documentación de EPANET, esta es lo suficientemente representativa de una red real, por tanto, es posible obtener conclusiones acerca de la aplicabilidad del sistema de detección y localización de fugas abordado en este trabajo, en sistemas de distribución de agua potable reales, las características más relevantes en cuanto a la cantidad de elementos de la red hidráulica son las siguientes.

- Número de nodos o uniones: 92
- Reservorios o fuentes de agua: 2

- Número de tanques: 3
- Número de tubería: 117
- Número de bombas: 2

Además, la red tiene diferentes patrones de demanda, específicamente, la red cuenta con cinco patrones de tiempo que modelan la variación de la demanda en los nodos. Estos patrones de tiempo son un conjunto de factores que pueden ser aplicados a una cantidad para modelar su variación a lo largo del tiempo [31].

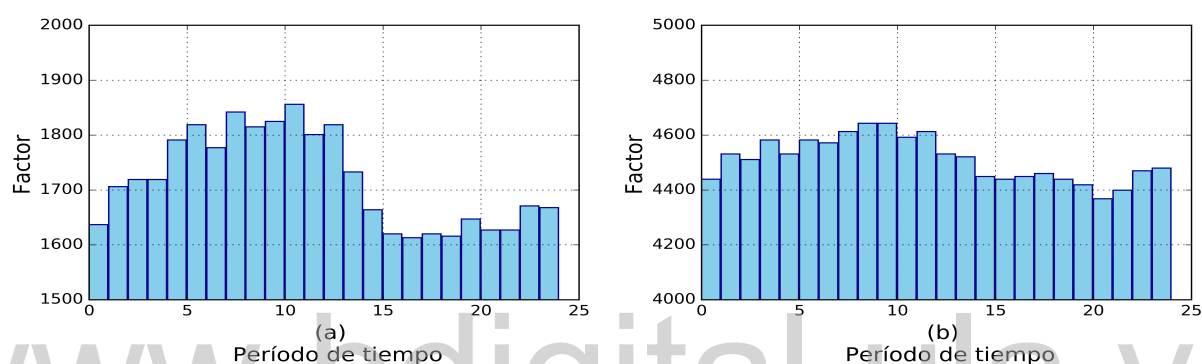


Figura 4.2. Patrones de tiempo para representar la variación de la demanda en los nodos de la red. (a) Patrón de tiempo para modelar la demanda en el nodo 35. (b) Patrón de tiempo para modelar la demanda en el nodo 203

En la Fig. 4.2 se puede apreciar dos patrones de tiempo para modelar la demanda en dos nodos particulares para un período de tiempo de una hora. Cada uno de los nodos de la red tiene definido una demanda base, de tal manera que al multiplicar por los factores de los patrones de tiempo, se puede obtener el consumo de agua para un periodo de 24 horas. Por otro lado, la red opera con diferentes acciones de control asociadas con el nivel de los tanques, estas acciones o maniobras definen el estado en el que se encuentran las bombas existentes en el sistema hidráulico. Finalmente, las características físicas de las tuberías tales como factor de rugosidad, diámetros y longitudes varían a lo largo de toda la extensión de la red.

4.2.2.1. Zonas de fuga

Siguiendo el método descrito anteriormente, la red se divide en zonas de fuga, donde cada una de estas abarca diferente cantidad de nodos, que están conectados entre sí con tuberías de diferentes características.

Zonas de fuga en la red 3

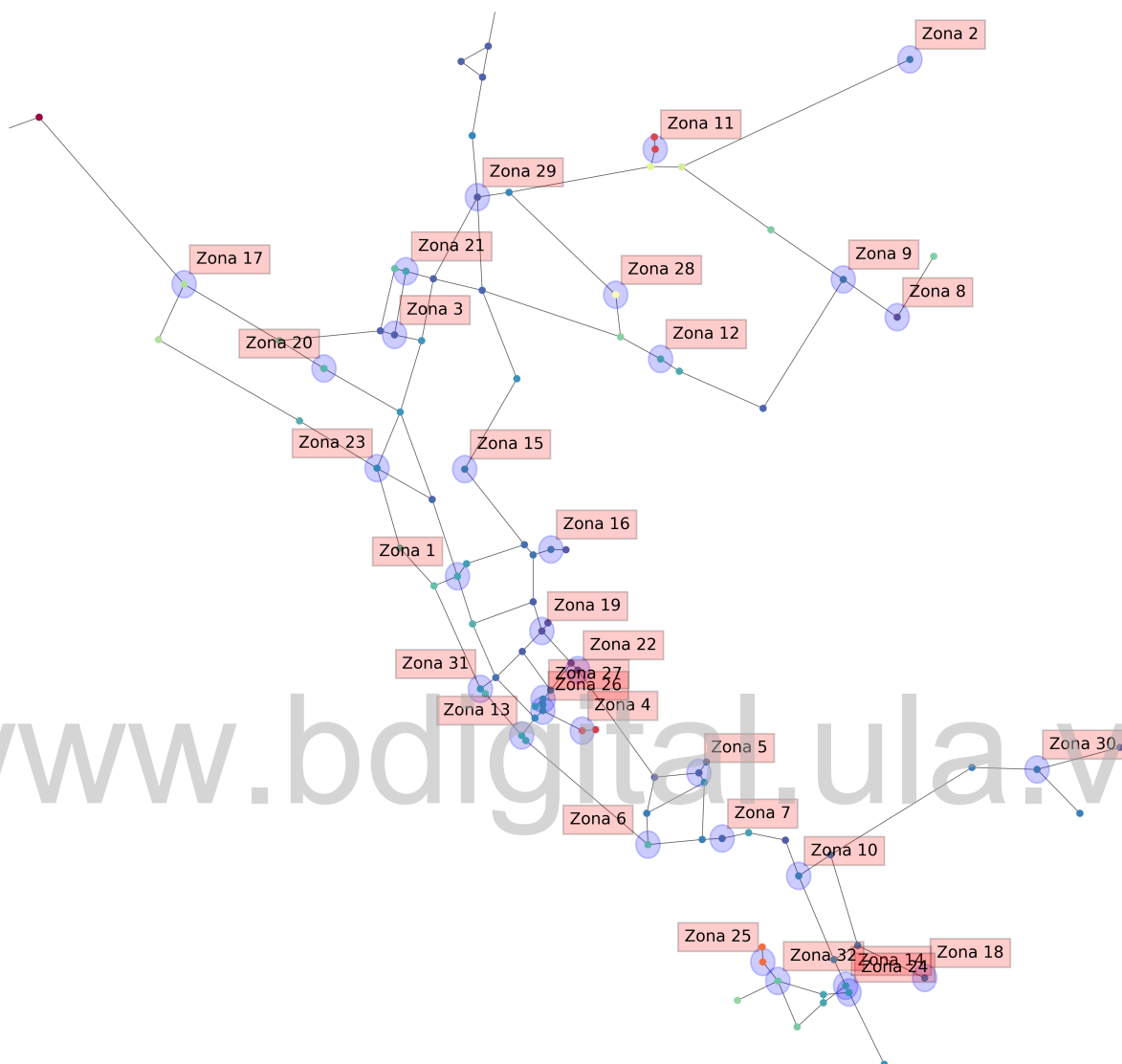


Figura 4.3. Sectores de la red de prueba Net3 de EPANET.

En la Fig. 4.3 se observa la red tres de la documentación de EPANET sectorizada, cada nodo encerrado en un círculo azul, junto con sus nodos adyacentes, conforman una zona de fuga. Es importante destacar que los nodos cercanos a las bombas, junto con tanques y reservorios, que son modelados como nodos, no se consideran para las zonas de fugas, esto es, se asume que no ocurren fugas en dichos nodos.

Esta consideración se realiza sólo para reducir la complejidad del sistema implementado, pues

conduce a una reducción significativa del número de zonas de fugas, sin embargo, en las redes reales, las fugas si pueden ocurrir en estos elementos¹, no obstante, este tipo de zonas en redes hidráulicas cuentan con sistemas de monitoreo ya implementados.

Finalmente, al aplicar el algoritmo se obtienen 32 zonas de fuga, en las que se encuentran distribuidos 88 nodos, en los cuales, se considera que puede ocurrir una fuga. Por tanto, el problema de clasificación considerado constará con 33 clases, 32 de ellas correspondientes a fugas en nodos pertenecientes a alguna de las zonas de fuga establecidas, y una clase extra para la condición en la que se asume que no existe fuga.

Tabla 4.1. Zonas de fuga en la red de prueba Net3.

Zona	Nodos	Zona	Nodos
1	193, 267, 113, 191, 195	17	101, 103
2	131, 129	18	231, 229
3	261, 263, 117	19	169, 265, 269, 167
4	40	20	107, 105
5	201, 275, 199, 203	21	257, 259, 120
6	205, 273	22	173, 171
7	206, 207, 208	23	111, 109, 115, 197
8	143, 15	24	241, 243, 247
9	141, 139, 145	25	50
10	211, 209, 213	26	179, 177, 183
11	20, 127	27	181, 271, 35
12	149, 147, 151	28	153
13	185, 184	29	121, 119, 125, 123
14	239, 237, 249	30	217, 215, 219, 225
15	159, 157, 161	31	187, 189, 204
16	164, 163, 166	32	255, 251, 253

En la Tabla 4.1 se puede observar la cantidad de nodos que hay en cada zona de fuga, como se aprecia, cada zona puede estar formada por un único nodo o hasta por cinco nodos².

4.2.3 Selección de datos

Los procesos de clasificación implican necesariamente el uso de datos, de tal manera que los clasificadores puedan establecer reglas de asociación con las diferentes categorías en un

¹Generalmente, cualquier evento de fuga en las cercanías de estos elementos puede detectarse de forma visual.

²Como resultado de las consideraciones previas, donde los tanques, reservorios y zonas adyacentes a las bombas no son considerados como puntos probables de falla.

problema dado, sin embargo, la información, en general, es un recurso costoso con el que no siempre se puede disponer, por ello es de vital importancia optimizar su uso, de tal manera que una muestra se pueda clasificar con la menor cantidad de información o características de esta. Por otro lado, la reducción de la cantidad de datos que se necesitan para clasificar una instancia acelera el proceso de entrenamiento de los algoritmos basados en aprendizaje automático, como consecuencia, los modelos desarrollados con estas consideraciones son mucho más rápidos, al requerir de menos datos para, finalmente, clasificar.

La aceleración del proceso de clasificación trae como consecuencia que las decisiones que se deban tomar en base a esta predicción de clase se hagan de forma rápida y óptima, por ejemplo, en medicina, es de suma importancia realizar un diagnóstico con la menor cantidad de datos, que se traduce en menor número de exámenes y por tanto una reducción del costo incurrido y, además, el diagnóstico debe hacerse a la brevedad posible, pues de esto dependerá la aplicación del correcto tratamiento y la probabilidad de recuperación del paciente. Situaciones como la descrita anteriormente se pueden hallar en diversidad de las áreas, por lo tanto, implementar un sistema de clasificación que optimice el uso de los datos y el tiempo asociado al proceso mismo, es de suma importancia.

4.3 APRENDIZAJE POR REFUERZO

Para abordar cualquier problema con los algoritmos desarrollados en el área del aprendizaje por refuerzo, es necesario contar con una serie de elementos, que modelan los problemas a tratar como un proceso de secuencial de observación y toma de acciones, esto como consecuencia de la estrecha relación con los procesos de decisión de Markov. En general, es necesario definir la función de recompensa, un conjunto de acciones, las observaciones que describen el estado del ambiente y, además, la naturaleza del ambiente, esto es, las reglas que determinan la transición de un estado a otro.

4.3.1 Observaciones

Para el problema de clasificación considerado es necesario disponer de una serie de datos con etiquetas de cada una de las clases existentes, en este caso, un evento de fuga en alguna de las zonas de la red. De esta manera, las muestras de un conjunto de datos \mathbf{D} son de la

forma $(\mathbf{x}, y) \in \mathbf{D}$ donde $\mathbf{x} \in \mathbf{X}$ contiene los valores asociados con las características de cada instancia, $y \in \mathbf{Y}$ es una etiqueta de clase en el problema en cuestión, en general, las clases son $Y = \{y_1, y_2, y_3, \dots, y_n\}$ donde n es el número de clases.

Para detectar y localizar las fugas en la red hidráulica considerada, cada instancia se define como un vector de características correspondientes al estado de la red en cada instante de tiempo, que viene definido por el paso de tiempo de la simulación hidráulica. Además, considerando el proceso de sectorización y el uso óptimo de los datos, cada característica se corresponde con los residuos de presión de los nodos que pertenecen a determinada zona.

4.3.2 Acciones

Partiendo de la metodología de la selección de características para el proceso de clasificación, las acciones que se establecen se corresponden a las acciones de selección de características \mathbf{A}_f y a las acciones de clasificación \mathbf{A}_c , de esta manera, el conjunto de acciones para el problema considerado se define como $\mathbf{A} = \mathbf{A}_f \cup \mathbf{A}_c$.

Para la detección y localización de fugas en redes hidráulicas, las acciones de selección de características revelan el valor de los residuos de presión de los nodos que pertenecen a la zona seleccionada, por consiguiente, habrán tantas acciones de selección de características como zonas de fuga establecidas en la red, en este caso 32.

Por otro lado, las acciones de clasificación se corresponden con las clases en el problema considerado, ya que se busca identificar en cual zona existe una fuga, el número de estas acciones viene determinado también por el número de zonas de fugas, considerando además, una clase extra, correspondiente a un estado de operación normal donde se considera que no existe fuga.

4.3.3 Función de recompensa

Para alentar un comportamiento en el agente que optimice el uso de las características y además desarrolle una alta exactitud en el problema de clasificación, se establece la función de recompensa de la siguiente manera.

$$R(a, l_m) = \begin{cases} \beta & \text{si } a \in \mathbf{A}_c \text{ y } a = l_m \\ -1 & \text{si } a \in \mathbf{A}_c \text{ y } a \neq l_m \\ -\lambda n_f & \text{si } a \in \mathbf{A}_f \end{cases} \quad (4.1)$$

En la Ec. (4.1) se define la función de recompensa, donde a es la acción tomada por el agente en cada paso de tiempo y l_m es la etiqueta de clase. Si la acción es de clasificación, y esta es correcta, el agente recibe una recompensa β que depende de la clase en cuestión. Si la acción consiste en seleccionar características, el agente recibe una penalización proporcional al número de características n_f seleccionadas anteriormente, en este caso, el parámetro λ se utiliza para asignar el grado de importancia que tiene la acción de seleccionar características antes de clasificar. Valores altos de λ alentarán al agente a realizar la acción de clasificación, mientras que, con valores más bajos, el agente tenderá a seleccionar más características antes de realizar una acción de clasificación.

Para reducir posibles sesgos que se pueda desarrollar hacia la clase con mayor número de muestras, en este caso, la clase correspondiente al estado de “no fuga”, el parámetro β toma un valor igual a 0,5 cuando la clasificación de las muestras pertenecientes a esta clase es correcta, mientras que para el resto de las clases $\beta = 1,0$. De esta forma, se espera desarrollar mayor sensibilidad a las clases con menor número de muestras.

4.3.4 Episodio

El problema de clasificación considerado se puede modelar como un problema secuencial, en el cual, en cada paso de tiempo, un agente puede decidir si tomar una característica de la instancia presentada o realizar una predicción de clase. Para el problema de detección y localización de fugas en una red de distribución de agua potable, cada observación se corresponde con un vector que contiene los residuos de presión para cada paso de tiempo de la simulación hidráulica.

Bajo el enfoque considerado, una observación consiste en un vector enmascarado, que contiene el valor de los residuos de presión para las zonas que se han seleccionado y cero en las posiciones correspondientes al resto de las zonas, qué zonas son reveladas y cuales no, depende de las acciones de selección de características. De esta forma, queda en manos del agente realizar tanto la selección de características de cada instancia como la clasificación de la misma, formalmente,

cada vector de observación se define como sigue.

$$\mathbf{O} = (\mathbf{P} \cdot \mathbf{m}_1, \mathbf{m}_2) \quad (4.2)$$

En la Ec. (4.2) \mathbf{P} es el vector con los residuos de presión para los nodos de la red, \mathbf{m}_1 es una máscara binaria que tiene 1 en las posiciones correspondientes a los nodos pertenecientes a las zonas de fuga que se han seleccionado previamente, y 0 para el resto de las posiciones, \mathbf{m}_2 es una máscara binaria que posee 1 en las posiciones concernientes a las zonas de fuga que se han seleccionado y 0 para el resto de las posiciones. Por lo tanto, el vector \mathbf{m}_1 contiene tantos elementos como nodos en la red, mientras que \mathbf{m}_2 tendrá un número de elementos igual al número de zonas de fuga establecidas.

4.3.5 Ambiente

En la mayoría de los problemas que se abordan con aprendizaje por refuerzo, el ambiente o entorno es influido de alguna manera por las acciones que un agente toma, no obstante, ya que el problema abordado obedece un esquema de clasificación, y las acciones, en este caso, se refieren a las etiquetas de las instancias, similar a como fueron atacados los problemas de clasificación en [7], [8], [9], el ambiente, considerado este como la red hidráulica en cuestión, no se ve alterada por las acciones que el agente tome. El proceso de transición entre los diferentes estados, esto es, los diferentes comportamientos de la red a lo largo del día, se desarrolla como se muestra en el siguiente algoritmo.

Algoritmo 4.2: Simulación del Ambiente

```

1 Paso de tiempo: timestep  $\leftarrow$  15 min ;
2 Duración de la simulación: n timestep  $\leftarrow$  96 (24 horas);
3 for  $n = 1, 2, 3 \dots total\ de\ episodios$  do
4   Seleccionar una zona de fuga  $z_l$  de forma aleatoria;
5   Seleccionar un nodo perteneciente a la zona de fuga  $z_l$ ;
6   Agregar fuga al nodo seleccionado;
7   Iniciar nueva simulación hidráulica;
8   while  $n\ timestep \leq 96$  do
9     Se crea el vector de residuos de presión para el paso de tiempo  $n\ timestep\ P_i$ ;
10    if  $a \in A$  then
11      if  $a \in A_c$  then
12         $R = \begin{cases} \beta & \text{si } a = l_m, \\ -1 & \text{si } a \neq l_m, \end{cases}$ 
13        n timestep  $\leftarrow$  n timestep + 1;
14         $c = 0$ ;
15        La máscara  $m_1$  se inicializa en cero;
16        La máscara  $m_2$  se inicializa en cero ;
17      else
18         $c \leftarrow c + 1$ ;
19         $R \leftarrow -\lambda \cdot c$ ;
20        En la máscara  $m_1$  se coloca 1 en la posición de los nodos de la zona  $a$ ;
21        En la máscara  $m_2$  se coloca 1 en la posición de la zona  $a$ ;
22      end
23    end
24    return  $O_i = (P \cdot m_1, m_2), R, T = Falso$ ;
25  end
26  n timestep = 0 ;
27  En la máscara  $m_1$  se coloca 1 en la posición de los nodos de la zona  $a$ ;
28  En la máscara  $m_2$  se coloca 1 en la posición de la zona  $a$ ;
29  return  $O_i = (P \cdot m_1, m_2), R, T = Verdadero$ ;

```

En el Algoritmo 4.2 se aprecia los pasos que describen el comportamiento del ambiente a lo largo de la simulación hidráulica, cada periodo de simulación tiene una duración de 24 horas con un paso de tiempo de 15 minutos. Para aproximar más el caso de estudio al comportamiento real de las redes de distribución de agua potable, la probabilidad de ocurrencia de fuga es mayor al comienzo de la simulación, que se corresponde con las horas de la madrugada. Durante estas horas, la demanda de agua por parte de usuarios es mínima, por lo tanto, la presión en el sistema es máxima, situación que incrementa la probabilidad de ocurrencia de fuga en el sistema.

Es importante mencionar que la dimensión de las fugas, diámetro del agujero considerado circular, se fija en 10,16 mm, este valor se obtiene considerando el 5 % del diámetro de una tubería de 200 mm de diámetro, que se corresponde con la tubería de menor tamaño en la red considerada, de esta manera, se logra tomar en cuenta las fugas de tipo goteo, que debido a su baja tasa de detección implican pérdidas de agua considerables.

4.4 ARQUITECTURA Y ENTRENAMIENTO DEL AGENTE

La técnica usada se basa en el algoritmo DQN, en este, la función Q es modelada por una red neuronal de tres capas completamente conectadas y 128 neuronas en cada capa. La dimensión del vector de entrada se corresponde con la dimensión del vector de observación, que resulta de la concatenación del vector de características con los residuos de presión enmascarados y la máscara binaria que indica las zonas que han sido seleccionadas. Siguiendo la metodología de [8], la máscara se incluye para que el agente pueda diferenciar entre un valor de cero y un valor que no ha sido observado.

La salida de la red neuronal se corresponde con los valores estimados de la función Q para los dos conjuntos de acciones de forma conjunta, esto es, las acciones correspondientes a la selección de características, en este caso, las zonas de fuga, y las referentes a la clasificación de la instancia presentada.

4.4.1 Entrenamiento

Par acelerar y estabilizar el aprendizaje del agente se implementaron varias de las técnicas desarrolladas en el área de aprendizaje por refuerzo, que incluye, la utilización de una red objetivo separada para estimar los valores de la función Q [16], esta red se actualiza cada 50 pasos. Se implementa la metodología *Dueling DQN* [24] y la técnica de muestro descrita anteriormente, *Prioritized Experience Replay* [22]. Los valores de los hiperparámetros fijados durante el entrenamiento se muestra a continuación.

Tabla 4.2. Parámetros para el entrenamiento del agente.

γ	α	ϵ_{frac}	ϵ_{end}	Batch Size	Memoria
0,95	0,0005	0,1	0,02	32	50000

En la Tabla 4.2 se aprecian los parámetros fijados durante el entrenamiento, ϵ_{frac} es la fracción

de todo el entrenamiento en el cual se mantiene una política de exploración activa, esto es, las acciones que se toman tienen una alta probabilidad de ser aleatoria, esta probabilidad se reduce gradualmente hasta alcanzar el valor de ϵ_{end} . El tamaño del conjunto de datos que se utiliza para el entrenamiento del agente es de 32, y se crea la memoria con un tamaño de 50000 transiciones.

Es importante destacar que el entrenamiento comienza sólo hasta que se hayan almacenado 50000 transiciones en la memoria del agente, de esta forma, el proceso de aprendizaje es acelerado. Se realizan entrenamientos para diferentes valores de λ , este valor se encarga de penalizar en mayor o menor medida las acciones que consisten en seleccionar características. Finalmente, para el entrenamiento de los modelos se usó la librería *stable-baselines*, donde se encuentran disponibles gran variedad de algoritmos desarrollados en el área de aprendizaje por refuerzo, compatibles con la estructura de un Gym, la librerías se encuentra disponible en <https://stable-baselines.readthedocs.io/en/master/>.

4.4.2 Curvas de Aprendizaje.

La curva de aprendizaje muestra el desempeño del agente a lo largo del entrenamiento, esto es, la cantidad de recompensa que el agente obtiene en cada uno de los episodios.

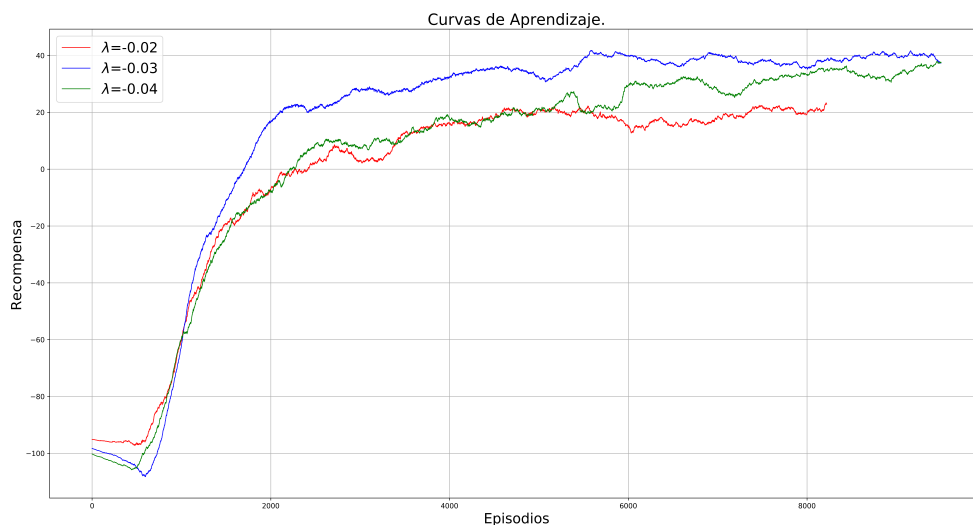


Figura 4.4. Curvas de aprendizaje obtenidas para diferentes valores de λ .

En la Fig. 4.4 se puede observar el desempeño del agente durante la etapa de entrenamiento,

para diferentes valores de λ . Se puede apreciar que el agente obtiene más recompensa, cuando este valor es igual a 0,03.

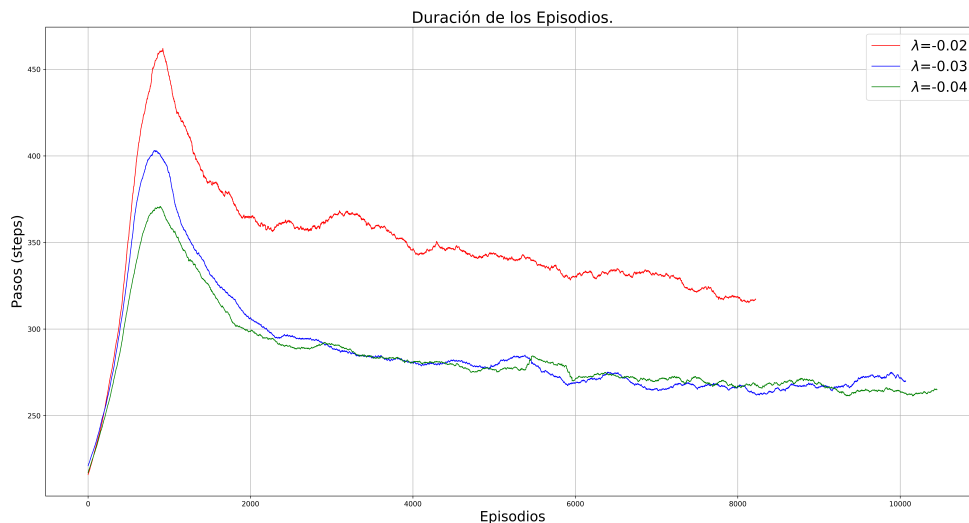


Figura 4.5. Duración de los episodios durante la etapa de entrenamiento.

En la Fig. 4.5 se puede apreciar la duración de los episodios medidos en pasos, para los diferentes valores de λ durante el entrenamiento. A medida que los episodios avanzan, estos se van haciendo más cortos, indicando que el agente reduce gradualmente el número de características que necesita para realizar una clasificación. Un pequeño valor para λ tiene como consecuencia episodios más largos, por otro lado, a medida que este valor aumenta los episodios tienden a ser más cortos, en la figura, valores de $\lambda = 0,03$ y $\lambda = 0,02$ tienden a tener un comportamiento similar.

4.5 RESULTADOS

Para probar los modelos ya entrenados se presenta un ambiente hidráulico similar al usado en el entrenamiento. En este caso, los parámetros del modelos se mantienen fijos y no se asigna recompensa alguna por las acciones que tome el agente, siendo la única función de éste decidir si tomar una característica de la muestra presentada o asignar una etiqueta de clase. De esta forma, para cada instancia presentada, seleccionará características hasta que decida realizar una predicción de clase. Si la acción es de clasificación un nuevo paso de tiempo de la simulación hidráulica es presentada, reiniciando así el proceso. El episodio termina sólo cuando se han

presentado 96 pasos de tiempo, correspondientes a un periodo de simulación de un día.

4.5.1 Métricas del clasificador

Para evaluar el desempeño de los modelos generados, se realiza una prueba de 500 episodios, que se corresponden con 500 días de simulación hidráulica, en los que deberá predecir una etiqueta de clase para los diferentes escenarios en los que se encuentre la red. Las métricas obtenidas se muestran en la siguiente tabla.

Tabla 4.3. Métricas de desempeño obtenidas durante la puesta en producción de los modelos generados con diversos valores de λ .

λ	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-score (%)</i>
0,02	80,56	77,68	76,15
0,03	88,89	88,25	88,11
0,04	78,79	76,73	75,77

En la Tabla 4.3 se aprecia el desempeño del agente para distintos valores de λ . Se observa que el mejor desempeño se obtiene cuando este valor es igual a 0,03. Las métricas altas implican que el agente puede discriminar relativamente bien entre las diversas clases del problema considerado.

4.5.2 Matriz de confusión

Para visualizar cuales son las etiquetas de clase que el agente tiende clasificar incorrectamente, se obtiene la matriz de confusión para el mejor modelo generado.

En la Fig. 4.6 se aprecia la matriz de confusión para la etapa de producción del agente, se observa que logra identificar todas las clases correspondientes a fugas en las respectivas zonas de la red. En la matriz se puede advertir que el mayor error lo comete al identificar fugas en la zona 24 como fugas en la zona 14 y viceversa. No obstante, observando la Fig. 4.3 se puede apreciar que estas dos zonas son vecinas, cuyos nodos se encuentran en condiciones geográficas similares. Como consecuencia, se esperaría que los efectos de una fuga sobre la presión del sistema, en algunos de estos nodos, sean similares, y por tanto, el clasificador tienda a confundir las clases correspondientes a fugas en estos nodos.

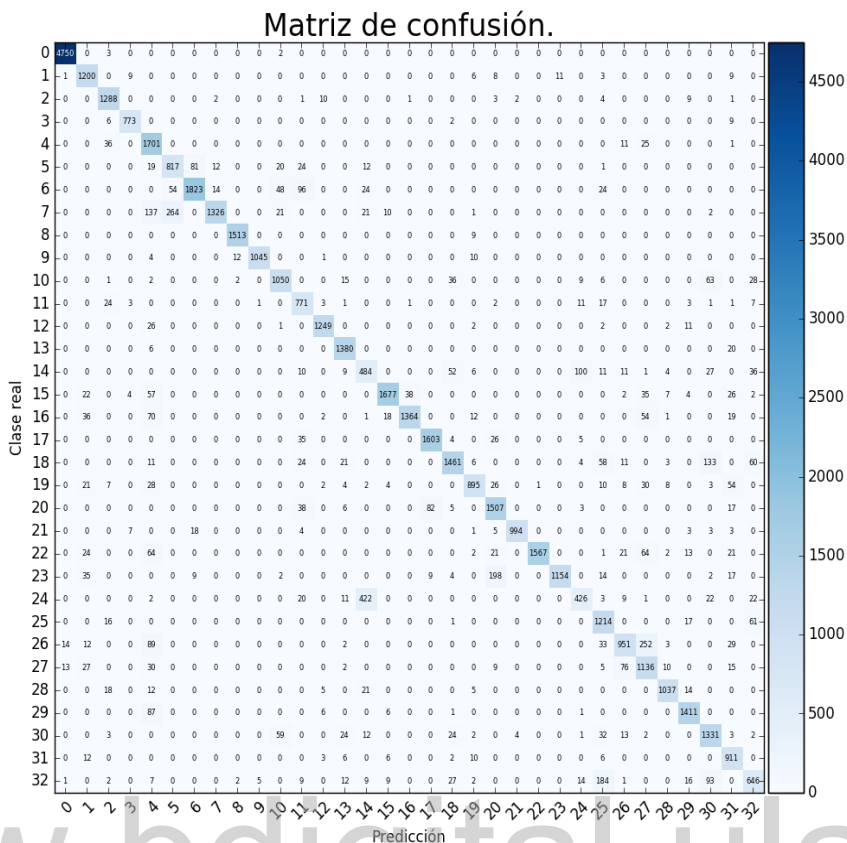


Figura 4.6. Matriz de confusión para el modelo entrenado con $\lambda = 0,3$.

4.5.3 Evaluación por zona

En la Fig. 4.7 puede observarse que las métricas obtenidas para cada una de las clases supera el 80 % en la mayoría de los casos. Nuevamente, se aprecia que el agente tiene mayor dificultad para diferenciar entre las instancias pertenecientes a las clases 14 y 24, correspondientes con fugas en los nodos de la zona 14 y en la zona 24 respectivamente. El número de instancias para la clase correspondiente al estado de operación normal, o de “no fuga”, es considerablemente mayor, esto es debido a la presencia de esta clase en todos los episodios, pues es el estado previo a un evento de fuga. En general, todas las clases en el problema considerado presentan distintos números de instancias o muestras, lo que se corresponde con un problema de clasificación con clases no balanceadas.

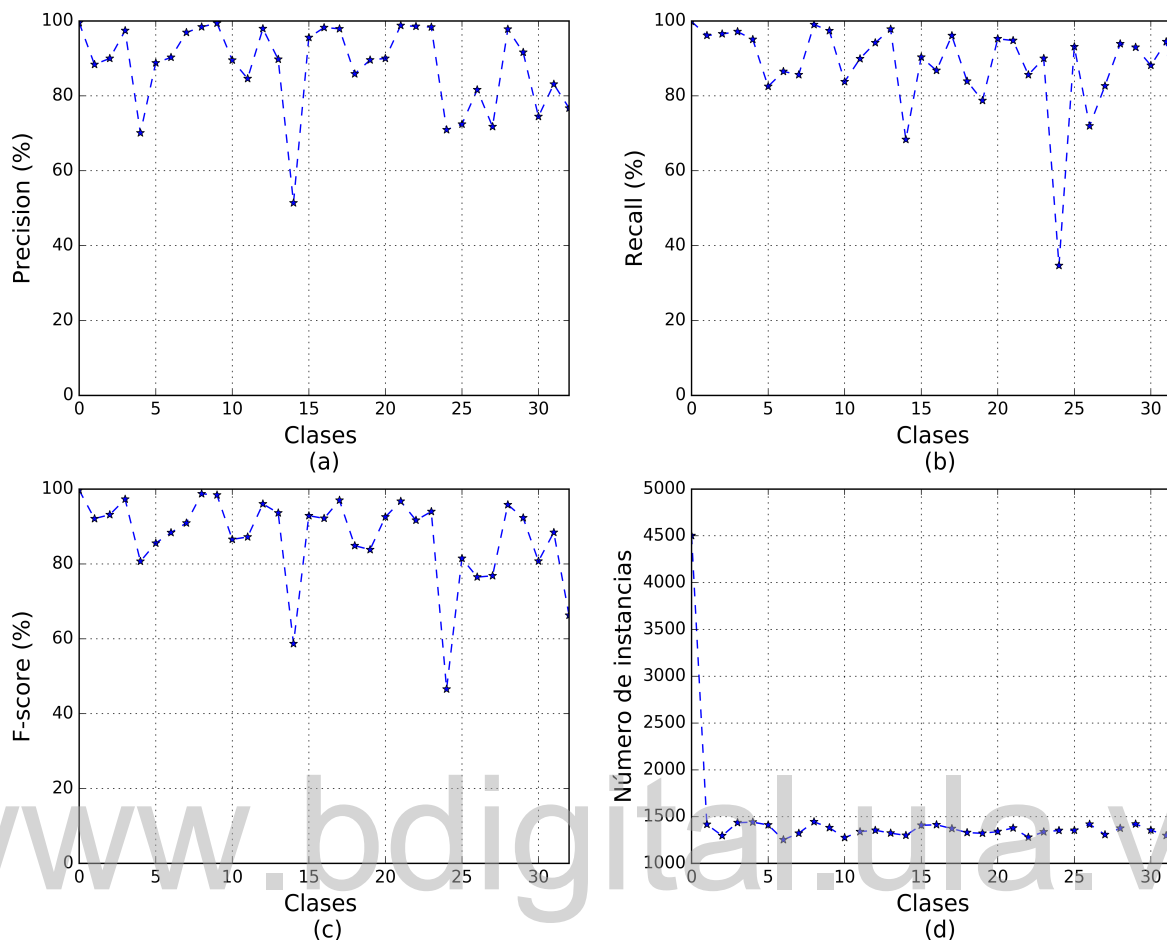


Figura 4.7. Métricas de desempeño considerando las clases individualmente. (a) Precision. (b) Recall. (c) F-score. (d) Número de instancias por cada una de las clases.

4.5.4 Características usadas para la clasificación

La característica principal de la metodología implementada para detectar y localizar fugas en algunas de las zonas de la red, radica en el uso eficiente de las características de cada una de las instancias que se muestran. Ya que las características de cada instancia se corresponden con residuos de presión en las zonas de la red, el uso selectivo de estas para la clasificación, implica que sólo se supervisan determinadas zonas de fuga en función del lugar o la zona donde ocurre el evento de fuga. En promedio, el agente monitorea dos zonas de fugas antes de predecir una etiqueta de clase. A continuación se muestra cuales son las zonas más usadas para el proceso de clasificación.

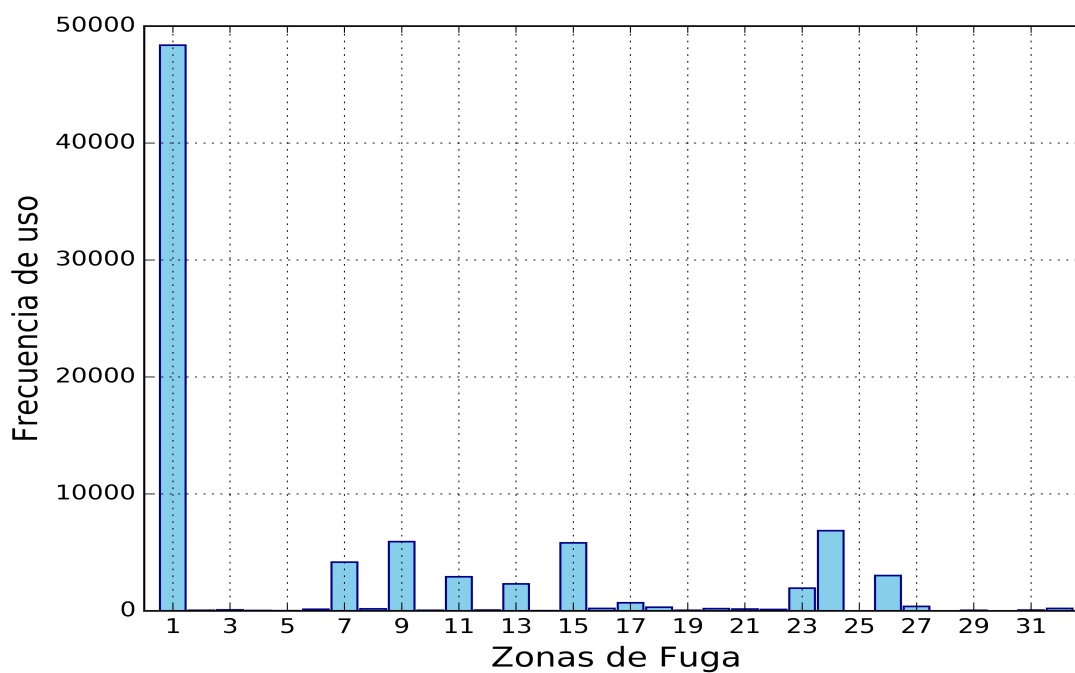


Figura 4.8. Frecuencia de las zonas monitoreadas por el agente antes de realizar una predicción de clase

En la Fig. 4.8 se puede apreciar el número de veces que cada zona es utilizada durante la prueba de 500 episodios, puede observarse que el algoritmo tiende a utilizar, de forma más significativa, los datos provenientes de la zona número 1.

CONCLUSIONES

- Aprendizaje por refuerzo demostró ser una alternativa viable para abordar el problema de detección y localización de fugas en sistemas de distribución de agua potable. Si bien, el esquema desarrollado enfoca el problema como una tarea de clasificación, el sistema implementado fue más allá de las capacidades de un clasificador básico, ya que permitió discriminar entre el total de características que conforma una muestra, seleccionar sólo aquellas que son estrictamente necesarias para la identificación de la clase a la que ésta pertenece. Por otro lado, la exactitud desarrollada por el algoritmo es dependiente de la función de recompensa, específicamente del valor del parámetro λ , que penaliza el uso de características de cada muestra antes de realizar una predicción de clase, las variaciones de este parámetro determinan, en general, el número de características que el agente explora antes de realizar una predicción de clase, siendo el valor de 0,03 con el que se obtuvo los mejores resultados.
- Debido al gran número de nodos en la red hidráulica utilizada durante las pruebas, detectar y localizar las fugas en cada nodo particular de la red implica la consideración de un gran número de clases, en este sentido, el proceso de sectorización realizado sobre la red de prueba contribuyó a la reducción del número de clases concernientes al problema de clasificación, reduciendo así la complejidad del problema considerado. En general, un proceso de sectorización se hace indispensable para abordar el problema de detección y localización de fugas en los sistemas de distribución de agua potable, pues en la práctica, estas infraestructuras se encuentran formadas por un gran número de elementos en los cuales puede presentarse una fuga.
- El esquema desarrollado se basa en la suposición de que se dispone de una base de datos con la cual se puedan comparar los valores obtenidos mediante simulación, y así, detectar anomalías que puedan indicar la presencia de una fuga en algún punto de la red. Por lo

tanto, la efectividad del método para detectar y localizar fugas en sistemas de distribución de agua potable depende de la existencia de un histórico del comportamiento de la red a lo largo del tiempo. Por otro lado, durante las pruebas se consideró un estado inicial en el que no existen eventos de fugas. Esta suposición se encuentra alejada de la realidad, ya que en este tipo de redes siempre existirán pérdidas de agua. Por lo tanto, los registros con los que se cuentan tendrán información sobre fallas en las redes para el momento de la adquisición, que son pasadas por alto al considerar que se trata de una condición libre de fugas.

- Si bien, las redes neuronales utilizadas para modelar la función Q del algoritmo DQN no están basadas en topologías con gran número de capas, los entrenamientos de estas requieren del ajuste de un elevado número de parámetros. La cantidad de posibles combinaciones a tomar, para asignar valores a cada uno de los parámetros involucrados en el entrenamiento de los modelos, es infinita. Entre estos parámetros tenemos a los concernientes a la actualización de la función Q a través de la ecuación de Bellman, es decir, la tasa de aprendizaje α y el factor de descuento de recompensa β , también tenemos los ajustes referentes al entrenamiento de la red neuronal, como, por ejemplo, la frecuencia con la que se actualizan redes neuronales auxiliares o el tamaño del conjunto de entrenamiento (*mini batch*) que se utiliza para actualizar los parámetros de la redes neuronales, por mencionar algunos. Pese a que se siguieron ciertos criterios para asignar valores a la mayoría de parámetros existentes, hallar el mejor modelo que sirva como solución a determinado problema no es una tarea trivial.

RECOMENDACIONES

El uso de aprendizaje por refuerzo para la detección y localización de fugas en redes de distribución de agua potable, se presenta como una metodología novedosa para resolver uno de los problemas que más afectan a estas infraestructuras. Si bien, la estrategia utilizada permite realizar el proceso de detección y localización de un evento de fuga en algún punto de la red, considerando el uso selectivo de características y, a su vez, manteniendo niveles de exactitud aceptables, debido a la gran cantidad de parámetros que se pueden ajustar sumado a la complejidad inherente de las redes de distribución de agua se hacen las siguientes recomendaciones a tener en cuenta como trabajo a futuro.

- Implementar una estrategia de sectorización que considere las características físicas de las tuberías y las condiciones geográficas en las que estas se encuentren en la red hidráulica, y así, generar zonas de fuga que agrupen elementos con características similares. De esta manera, se esperaría que el efecto de una fuga en alguna de estas zonas produzca un comportamiento característico de la ubicación y sea más sencillo para un agente basado en aprendizaje por refuerzo discriminar los eventos de falla en las distintas zonas.
- Considerar mediciones de caudal en las tuberías como otra variable en el vector de observación que indique el estado en el que se encuentra la red hidráulica, de esta manera, un agente basado en aprendizaje por refuerzo podrá identificar de manera más fácil posibles escenarios de fuga en algún punto de la red.
- La función de recompensa utilizada, establece una penalización por utilizar características de cada una de las instancias que varía de forma lineal con el número de zonas monitoreadas. No obstante, según [7], utilizar variaciones de forma potencial en la función de recompensa para penalizar el uso de características, demostró brindar excelentes resultados. En este sentido, se propone explorar una función de recompensa que varíe

de forma potencial con el número de características que se utilizan antes de obtener una predicción de clase.

- El problema de detección y localización de fugas, enfocado como una tarea de clasificación, presenta problemas de desbalance en el número de instancias en cada una de las clases consideradas. Como consecuencia, el modelo clasificador puede generar sesgos hacia aquellas clases con mayor número de muestras. En este sentido, se propone utilizar métodos de muestreo estratificado para asegurar que el subconjunto de datos (*mini batch*) que se utiliza para entrenar la red neuronal que aproxima la función Q del agente, se encuentre equilibrado.
- Debido al gran número de parámetros ajustables en la metodología implementada, se propone realizar un barrido en rejilla de aquellos parámetros más importantes para el entrenamiento del agente. De esta manera, al realizar un ajuste más fino sobre estos parámetros se podrá tener un modelo clasificador que brinde resultados más exactos en cuanto a la detección y localización de las fugas.

www.bdigital.ula.ve

APÉNDICE A. GYM HIDRÁULICO

En este apéndice se describe la estructura seguida para la elaboración del entorno de simulación hidráulico, usado para entrenar el agente basado en aprendizaje por refuerzo, con el objetivo de detectar y localizar fugas en redes de distribución de agua potable. Para crear el entorno de simulación se sigue una estructura ya definida por [32], esta se muestra a continuación.

Programa 4.1. Estructura de Gym

```
import gym
from gym import error, spaces, utils
from gym.utils import seeding

class Environment(gym.Env):
    metadata = {'render.modes': ['human']}

    def __init__(self):
        ...
    def step(self, action):
        ...
    def reset(self):
        ...
    def render(self, mode='human'):
        ...
    def close(self):
        ...
```

Los métodos más importantes para la simulación del ambiente son el constructor, el método `step()` y `reset()`. El constructor se encarga de inicializar los parámetros necesarios para simular el ambiente, de definir el espacio de observación y de acción. El método `step()` ejecuta una acción en cada paso de tiempo, el parámetro de entrada es la acción y tiene como salida un vector con la

observación obtenida del ambiente, la recompensa que se otorga al agente por la acción realizada y la condición del episodio, esto es, si es terminal o no, en caso de ser terminal, el método `reset()` es llamado, de esta forma, todos los valores vuelven al estado inicial, y un nuevo evento de fuga es presentado, dando inicio a un nuevo episodio. Finalmente, el método `render()` se utiliza para obtener una representación gráfica del proceso que se está simulando. A continuación se muestra el código implementado para la simulación de la red hidráulica.

Programa 4.2. Gym Hidráulico

```
def __init__(self, *args, **kwargs):
    self.inp_file='Net3.inp'
    self.viewer=None
    self.fig_created=False
    self.scaler=MinMaxScaler(feature_range = (0,1))
    self.wn=wntr.network.WaterNetworkModel(self.inp_file)
    self.timestep=900
    self.wn.options.time.duration=97*900
    self.wn.options.time.hydraulic_timestep=900
    #sectors
    with open('sectors','rb') as f:
        self.sectors=pickle.load(f)
    #nodos en los que ocurre fuga
    self.nodes_monitor=[]
    self.nodes_leak=[]
    for sector in self.sectors:
        self.nodes_monitor.append(sector[0])
        for node in sector:
            self.nodes_leak.append(node)
    #Pressures in normal operating condition
    sim_r=wntr.sim.WNTRSimulator(self.wn,mode='PDD')
    self.results=sim_r.run_sim()
    self.results2=self.results.node['pressure'][self.nodes_leak]
    self.wn=wntr.network.WaterNetworkModel(self.inp_file)
    self.wn.options.time.hydraulic_timestep=900
    self.wn.options.time.duration=self.timestep

    self.situations=["No Leak"]
    self.leak_zones_id=[]
```

```

for y,_ in enumerate(self.sectors):
    self.situations.append("Leak at zone "+str(y+1))
    self.leak_zones_id.append(y+1)
#select zone
self.leak_zone=np.random.choice(self.leak_zones_id)
#select node to fail
self.scenario=np.random.choice(self.sectors[self.leak_zone-1])
#add leak
self.pro=np.exp(-(np.linspace(0,86400,97)/15000)**2)/\
sum(np.exp(-(np.linspace(0,86400,97)/15000)**2))
self.leak_size=np.round(np.random.uniform(0.005,0.05),3)
self.start_time=np.round(\
np.random.choice(np.linspace(0,86400,97),p=self.pro))

self.end_time=86400-self.start_time
self.node_to_fail=self.scenario
self.leak_diameter=0.0254*8.0*self.leak_size
self.leak_area=np.pi*(self.leak_diameter/2)**2
self.leak_node=self.wn.get_node(self.node_to_fail)
self.leak_node.add_leak(self.wn, area=self.leak_area,
                        start_time=self.start_time,
                        end_time=(self.start_time+\
self.end_time))

#mensaje
print("Leak at Zone {} in Node {} Leak Size {} %\
      Start Time {}".format(self.leak_zone,
                             self.node_to_fail,np.round((self.leak_size*100),3),
                             str(timedelta(seconds=(self.start_time))))))
self.action_space=spaces.Discrete(65)
self.observation_space = spaces.Box(np.zeros(120),
                                     np.ones(120), dtype=np.float32)

self.lambda_=-0.03
self.count=0
self.nstep=0
self.nstep2=0
self.total_reward=0
self.mask=np.zeros(len(self.nodes_leak))
self.mask_zone=np.zeros(len(self.nodes_monitor))

```

```

def step(self, action):

    assert self.action_space.contains(action), \
        "%r (%s) invalid" %(action, type(action))

    if self.nstep*900 < self.start_time:
        self.stat2 = 'No Leak'
    else:
        self.stat2 = 'Leak at zone '+str(self.leak_zone)

    self.predict = action
    reward = self.get_reward(action)

    if action >= len(self.nodes_monitor) or self.nstep2 >=32:
        self.nstep += 1
        self.nstep2 = 0
        self.mask = np.zeros(len(self.nodes_leak))
        self.mask_zone = np.zeros(len(self.nodes_monitor))
        print("Time: {} State: {} Prediction: {} "
            .format(str(timedelta(seconds = self.nstep*900)),
                self.stat2, self.situations[action-len\
                    (self.nodes_monitor)]))
    else: #feature selection
        self.nstep2 +=1
        for node in self.sectors[action]:
            self.mask[self.nodes_leak.index(node)]=1
        self.mask_zone[action] = 1
        print("Time: {} State: {} Feature Selection [Nodes]:{} "
            .format(str(timedelta(seconds = self.nstep*900)),
                self.stat2, [self.nodes_leak[i] for i in np.where\
                    (self.mask==1)[0]]))

    if self.nstep >= 96:# end of simulation
        done = True
    else:
        done = False

    state = np.squeeze(self.scaler.fit_transform\
        (np.array(self.Pres.iloc[self.nstep])\
            .reshape(-1,1)))
    state = np.concatenate((state*self.mask, self.mask_zone))
    self.total_reward += reward

```

```
return state, reward, done, {}
```

```
def get_reward(self, action):
    if action >= len(self.nodes_monitor):
        if self.situations[action-len(self.nodes_monitor)]\
            == self.stat2:
            if self.stat2 == 'No Leak':
                return 0.5
            else:
                return 1
        else:
            return -1
    if action < len(self.nodes_monitor):
        return self.nstep2*self.lambda_

def reset(self):
    print("RESET TOTAL REWARD {}".format(self.total_reward))
    self.nstep = 0
    self.nstep2 = 0
    self.total_reward = 0
    self.mask = np.zeros(len(self.nodes_leak))
    self.mask_zone = np.zeros(len(self.nodes_monitor))
    #new lake scenario
    self.wn = wntr.network.WaterNetworkModel(self.inp_file)
    self.wn.options.time.duration = self.timestep*97
    self.wn.options.time.hydraulic_timestep = 900
    self.wn.options.time.report_timestep = 900
    if self.count == 0:
        np.random.shuffle(self.leak_zones_id)
    if self.count == len(self.leak_zones_id):
        self.count = 0
    ##select zone
    self.leak_zone = self.leak_zones_id[self.count]
    self.count += 1
    #select node to fail
    self.scenario = np.random.choice\
        (self.sectors[self.leak_zone-1])
    #add leak
    #self.leak_size = np.round(np.random.uniform(0.005,0.05),3)
    self.leak_size = 0.000001
    self.start_time = np.round(np.random.choice\
        (np.linspace(0,86400,97), p=self.pro))
    self.end_time = 86400-self.start_time
    self.node_to_fail = self.scenario
```

```
self.leak_diameter = 0.0254*8.0*self.leak_size
self.leak_area = np.pi*(self.leak_diameter/2)**2
```

```
self.leak_node = self.wn.get_node(self.node_to_fail)

self.leak_node.add_leak(self.wn, area=self.leak_area,
                        start_time=self.start_time,
                        end_time=(self.start_time+\
self.end_time))

#run simulation
sim = wntr.sim.WNTRSimulator(self.wn,mode = 'PDD')
self.Ps = sim.run_sim()
self.Pleak = self.Ps.node['pressure'][self.nodes_leak]
#residuos
self.Pres = self.results2 - self.Pleak
#"normalization"
if self.nstep*900 < self.start_time:
    self.stat2 = 'No Leak'
else:
    self.stat2 = 'Leak at zone '+str(self.leak_zone)
#message
print("LEAK AT ZONE: {}  NODE: {} LEAK SIZE:\
{} % START TIME: {} ".format(self.leak_zone,
self.node_to_fail,np.round((self.leak_size*100),3),
str(timedelta(seconds=(self.start_time))))))

state = np.squeeze(self.scaler.fit_transform\
(np.array(self.Pres.iloc[self.nstep]).reshape(-1,1)))
state = np.concatenate((state*self.mask,self.mask_zone))
return state
```

REFERENCIAS

- [1] Programa Mundial de las Naciones Unidas de Evaluación de los Recursos Hídricos, “Informe Mundial de las Naciones Unidas sobre el Desarrollo de los Recursos Hídricos 2018: Soluciones basadas en la naturaleza para la gestión del agua”, UNESCO, París, inf. téc.
- [2] D. J. Mogollón, *Estudio de la Probabilidad de Rotura en Abastecimientos Urbanos con la Ayuda de Modelos Estadísticos e Hidráulicos. Estrategias de Renovación de Activos*, Trabajo de Fin de Máster, Dep. de Ingeniería Hidráulica y Medio Ambiente, Universidad Politécnica de Valencia, Valencia, España, 2017.
- [3] I. G. y P. Isasi, *Redes de Neuronas Artificiales. Un enfoque práctico*. Madrid, España: Pearson Educación. S.A, 2004, pp. 197-200.
- [4] G. Aurélien, *Hands-On Machine Learning with Scikit Learn and Tensorflow*. EE.UU: O’Reily Media, 2017, pp. 106-123, 557-582.
- [5] G. Gupta, *Monitoring Water Distribution Network using Machine Learning*, Degree Project in Electrical Engineering, KTH Royal Institute of Technology School of Electrical Engineering, Stockholm, Sweden, 2017.
- [6] I. Rojek and J. Studzinski, “Detection and Localization of Water Leaks in Water Nets Supported by an ICT System with Artificial Intelligence Methods as a Way Forward for Smart Cities”, *Sustainability*, vol. 11, n.º 518, Jan. de 2019.
- [7] M. Coralie, G. Perrin, E. Ramasso and M. Rombaut, “A deep reinforcement learning approach for early classification of time series”, *Presented at 26th European Signal Processing Conference*, Rome, Italy, Sep. de 2018.
- [8] J. Janisch, T. Pevný and V. Lisý, “Classification with Costly Features using Deep Reinforcement Learning”, *Adaptive Learning Agents Workshop*, 2019.
- [9] E. Lin, Q. Chen and Q. Xiaoming, “*Deep Reinforcement Learning for Imbalanced Classification*”, submitted for publication.

- [10] D. Zhao, Y. Chen and L. Lv, “Deep Reinforcement Learning With Visual Attention for Vehicle Classification”, *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, n.º 4, Dec. de 2017.
- [11] J. E. Van Ly, *Introduction to Operation and Maintenance of Water Distribution Systems*. Pretoria, Republic of South Africa: Water Research Commission, 2014, pp. 121-131.
- [12] S. Irukula, *Reinforcement Learning Based Controller for Precision Irrigation*, Dissertation for obtaining the degree of Master of Science, Texas AM University, Dec. de 2015.
- [13] X. Liang, X. Du, G. Wang and Z. Han, “Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks”, *IEEE Transactions on Vehicular Technology*, vol. 20, n.º 20, Mar. de 2018.
- [14] P. Nguyen and T. Endo, “Applying Reinforcement Learning in Operations of Water Purification Plants”, *Presented at IWA World Water Congress Exhibitions*, Tokyo, Japan, Sep. de 2018.
- [15] K. A. Klise, R. Murray and T. Haxton, *Water Network Tools Resilience, WNTR (0.1.7)*. Available:<https://wntr.readthedocs.io/en/latest/>, 2018.
- [16] V. Mnih, K. Kavukcuoglu and D. Silver, “Human Level control through deep reinforcement learning”, *Nature*, vol. 518, n.º 529, Feb. de 2015.
- [17] DeepMind. (2019). AlphaGo, [online] Available:<https://deepmind.com>.
- [18] R. Sutton and A. Barto, *Reinforcement Learning an Introduction*. MIT Press, 2017, pp. 37-54, 97-112, 161-187.
- [19] F. Domínguez, *Entrenando bots para juegos mediante aprendizaje por refuerzo basado en casos*, Doble Grado en Ingeniería Informática y Matemáticas, Universidad Complutense de Madrid, Madrid, España, 2017.
- [20] D. Silver. (2015). UCL Course on RL [lecture notes], [online] Available:<http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html>.
- [21] T. Simonini. (2018). Improvements in Deep Q Learning: Dueling Double DQN, Prioritized Experience Replay, and fixed Q-targets, [online] Available:<https://www.freecodecamp.org>.
- [22] T. Schaul, J. Quan, I. Antonoglou and D. Silver, “Prioritized Experience Replay”, *Presented at International Conference on Learning Representations*, Feb. de 2016.

- [23] H. Van Hasselt, A. Guez and D. Silver, “Deep Reinforcement Learning with Double Q-Learning”, pp. 2094-2100, *Presented at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [24] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot and N. de Freitas, “Dueling Network Architectures for Deep Reinforcement Learning”, *Presented at 33 rd International Conference on Machine Learning*, vol. 48, New York, USA, 2016.
- [25] H. E. Hickey, *Water Supply Systems and Evaluation Methods*. U.S. Fire Administration, 2008, pp. 1-6.
- [26] O. Fuentes, A. Palma y K. Rodríguez, “Estimación y localización de fugas en una red de tuberías de agua potable usando algoritmos genéticos”, *Ingeniería Investigación y Tecnología*, vol. 12, n.º 2, 2011.
- [27] F. K. Seyed, *Failure Rate Prediction Models of Water Distribution Network*, A Thesis for Degree of Master of Applied Science (Civil Engineering), Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, Quebec, Canada, 2015.
- [28] E. Rondán, *Estado del arte de la calibración de modelos hidráulicos. Modelado de fugas con Epanet*, Trabajo de fin de grado, Dep. Ingeniería Energética, Universidad de Sevilla, Sevilla, España, 2016.
- [29] D. A. Crowl y J. Louvar., *Chemical Process Safety: Fundamentals with Applications*. Prentice Hall, 2002.
- [30] E. Campbell, *Propuesta Para una Metodología de Sectorización de Redes de Abastecimiento de Agua Potable*, Trabajo de Fin de Máster, Universidad Politécnica de Valencia, Valencia, España, 2013.
- [31] *EPANET 2 MANUAL DE USUARIO*, United States Environmental Protection Agency, pp. 36-37.
- [32] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, *OpenAI Gym*, 2016.