UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA DE POSTGRADO Y EDUCACIÓN CONTINUA

ESTABILIZACIÓN DE LA OPERACIÓN DE MOLINO SAG MEDIANTE MODELO PRESCRIPTIVO UTILIZANDO TÉCNICAS DE APRENDIZAJE REFORZADO PROFUNDO FUERA DE LÍNEA

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIA DE DATOS

JOSÉ LUIS CÁDIZ SEJAS

PROFESOR GUÍA:
Marcos Orchard Concha

MIEMBROS DE LA COMISIÓN:
Carlos Orellana Sandoval
Javier Ruiz del Solar

Este trabajo ha sido financiado parcialmente por:
ANGLO AMERICAN

SANTIAGO DE CHILE
2024

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE MAGÍSTER EN CIENCIA
DE DATOS
POR: JOSÉ LUIS CÁDIZ SEJAS
FECHA: 2024
PROF. GUÍA: MARCOS ORCHARD CONCHA

ESTABILIZACIÓN DE LA OPERACIÓN DE MOLINO SAG MEDIANTE MODELO PRESCRIPTIVO UTILIZANDO TÉCNICAS DE APRENDIZAJE REFORZADO PROFUNDO FUERA DE LÍNEA

El objetivo de esta tesis es desarrollar un sistema que contribuya a la mejora de toma de decisiones de la operación de molinos SAG mediante el desarrollo de un modelo prescriptivo utilizando técnicas de Aprendizaje Reforzado Profundo Fuera de Línea. Para cumplir con el objetivo de la tesis se pretende demostrar que es posible la utilización de algoritmos avanzados de Aprendizaje Reforzado en aplicaciones prácticas dentro de la industria minera, evidenciando que el algoritmo es capaz de aprender políticas óptimas con un sentido operacional mediante su proceso de entrenamiento y que además es posible mejorar la calidad de las recomendaciones que se generan a través de los modelos actuales, los cuales son denominados como **modelo estadístico** (o modelo base) y **modelo estadístico mejorado**.

La metodología desarrollada consiste en consolidar y limpiar la fuente de datos que será utilizada para entrenar y testear la toma de decisiones del algoritmo, desarrollar una métrica de evaluación para los modelos, modelar el problema con un enfoque de Aprendizaje Reforzado, entrenar el modelo y finalmente, evaluar y analizar los resultados obtenidos.

En cuanto a los resultados obtenido al evaluar los modelos mediante la métrica desarrollada, el ranking en orden descendente de desempeño fue: **Modelo estadístico mejorado**, **Modelo RL** y **Modelo estadístico**.

Agradecimientos

A mi familia y amigos por siempre celebrar mis momentos de éxito y apoyarme en mis momentos más difíciles.

A todo el equipo de applied Intelligence de Accenture, por ser parte clave en el inicio de mi camino profesional, lo cual a su vez me permitió abrir todo un mundo de oportunidades.

A Carlos Orellana Data Scientist de Anglo American y actual amigo, por querer ser mi mentor y darme la oportunidad y el apoyo necesario para seguir creciendo profesionalmente.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción y objetivos

1.1. Introducción

La motivación de este tema nace a partir del proyecto "Tactical Recipe" entre Anglo American y Accenture. El objetivo del proyecto fue mejorar la producción de la planta concentradora de cobre Confluencia de la mina Los Bronces, a través de la reducción de las caídas de toneladas procesadas por hora (TPH). Para esto se desarrolló un modelo estadístico que recomienda el límite alto de carga del molino SAG. Esto se conoce como la recomendación de celda de carga.

El modelo estadístico logro ser una solución simple que permitió generar un entendimiento del fenómeno, sin embargo, existen enfoques del aprendizaje de máquinas que están enfocados en optimizar la toma de decisiones (encontrar la política óptima), este enfoque es conocido como **Aprendizaje reforzado** (RL) [1].

El enfoque tradicional del aprendizaje reforzado requiere experimentación para mejorar la toma decisiones, sin embargo, un molino SAG es un activo de alto costo de inversión para la industria minera, por esto, no es factible experimentar con modos operacionales que no son respaldados por la operación. Si se toma una mala decisión, este activo puede terminar dañado, incurriendo en altos costos de reparación y además generando cuellos de botella para la producción.

Recientemente, se ha estado explorando en la literatura un nuevo enfoque de RL llamado **Aprendizaje Reforzado Fuera de Línea** [2], este se caracteriza por aprender políticas óptimas únicamente a partir de datos históricos. Esta política aprendida posteriormente puede ser re-entrenada ($fine - tuning$) en un ambiente en línea para seguir mejorando su aprendizaje pero con un conocimiento previo del proceso [3]. En la figura 1.1 se esquematiza esta idea.

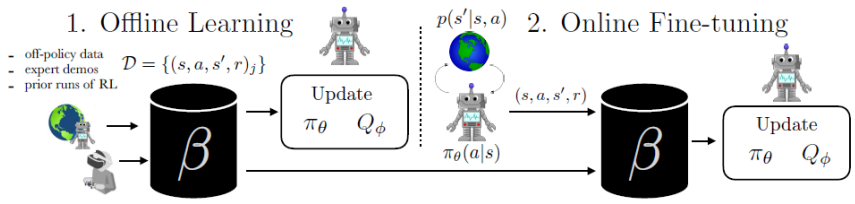

Figura 1.1: Esquema de entrenamiento fuera de línea y posterior re-entrenamiento con interacción en línea.

1.2. Hipótesis de investigación y objetivos

1.2.1. Hipótesis de investigación

Es posible la utilización de algoritmos avanzados de Aprendizaje Reforzado en aplicaciones prácticas dentro de la industria minera, evidenciando que el algoritmo es capaz de aprender políticas óptimas con un sentido operacional mediante su proceso de entrenamiento y que además es posible mejorar la calidad de las recomendaciones que se generan mediante los modelos actuales.

1.2.2. Objetivo general

Diseñar y desarrollar sistema prescriptivo basado en Aprendizaje Reforzado Profundo Fuera de Línea en molino SAG para la estabilización de la operación en función del límite superior de su celda de carga.

1.2.3. Objetivos específicos

- Diseñar y desarrollar una métrica de desempeño para comparar la calidad de las recomendaciones respecto a modelos ya existentes.
- Diseñar y desarrollar un modelo prescriptivo basado en Aprendizaje Reforzado Fuera de Línea.
- Evaluar los modelos desarrollados, comparar y analizar resultados.

Capítulo 2

Marco teórico

2.1. Molino SAG

2.1.1. Introducción [4]

La molienda autógena (AG) consiste en un molino cilíndrico, cuya característica física principal es que el diámetro es 2 a 3 veces su largo, del orden de 20 metros. La palabra autógena indica que la molienda ocurre debido a la propia acción de la caída de las colpas de mineral desde una altura cercana al diámetro del molino, es decir, no se emplea otro medio de molienda adicional que la roca misma. Por lo tanto, la carga de alimentación debe contener una fracción gruesa con la suficiente calidad y competencia como medio de molienda (dureza) para impactar y friccionar las fracciones de menor granulometría de la carga hasta reducir sus tamaños.

La molienda semi-autógena (SAG) es una variación del proceso de molienda autógena, es la más frecuente en la práctica y en ella se adicionan bolas de acero. El nivel volumétrico de llenado de bolas varia normalmente de 4 % a 14 % con respecto al volumen interno del molino. Para generalizar ambas alternativas generalmente se habla de molinos de cascada.

Las principales variables de interés para el contexto del problema son:

1. **TPH**: Indica las toneladas por hora procesadas y la velocidad de llenado del molino SAG.

2. **Setpoint TPH**: Es el objetivo (o setpoint) que define la operación, idealmente el TPH siempre debe estar en torno a este valor.

3. **Celda de carga**: Indica el peso del molino, considerando el peso del molino y la pulpa interna de este.

4. **Setpoint HH celda de carga**: Es el límite superior de la señal de celda de carga. Adicionalmente, para la celda de carga también existe el setpoint LL celda de carga, que define el límite inferior de esta. Idealmente la celda de carga siempre debe estar dentro de estos límites. Cabe destacar que el termino HH y LL se refiere al límite “alto alto” y “bajo bajo” de celda de carga respectivamente. Esto debido a que existen otras variables operacionales que adicionalmente tienen un límite “alto” y “bajo”, lo cual tiene como trasfondo el nivel de tolerancia a sobre pasar ciertos límites.

5. **Rpm**: Indica las revoluciones por minuto (rpm) del molino SAG.

6. **Porcentaje de sólidos**: Indica la proporción de material sólido respecto del contenido total dentro del molino (mineral, cal y agua).

7. **Agua**: Indica cuanta agua con cal esta ingresando al molino SAG.

8. **Granulometría**: Indica cual es el tamaño del mineral entrante al molino SAG, específicamente, se tiene la medición del porcentaje de mineral pasante dentro de una malla (0 % a 100 %), es decir, si se tiene un alto valor, significa que hay una mayor cantidad de mineral pasante, lo cual implica un mineral de menor tamaño. Por otro lado, si se tiene un bajo valor, implica que se tiene un mineral pasante de mayor tamaño.

9. **Dureza**: Indica cual es el nivel de dureza del mineral entrante al molino SAG, específicamente se posee la medición del SPI (SAG Power Index), este mide el tiempo en minutos que demora el 80 % de una roca en pasar de una granulometría de 12.7mm a una granulometría de 1.7mm.

10. **Desgaste del revestimiento del molino**: Los revestimiento del molino SAG están definidos principalmente por los dientes internos que permiten levantar y hacer girar el mineral. En la medida de que el nivel de procesamiento del molino se va acumulando, estos revestimientos se van desgastando y se deben adoptar estrategias operativas diferentes según el nivel de desgaste.

2.1.2. El problema a resolver

El problema a resolver busca encontrar una estrategia de toma de decisiones óptima que defina el límite alto de celda de carga con el que se debe operar el molino SAG, esto con el objetivo de que las toneladas por horas procesadas (**TPH**) se mantengan estables respecto de lo definido por la operación (**setpoint TPH**).

Las caídas de TPH identificadas se deben principalmente por dos motivos:

i) **Caídas por acción del sistema de control**: Esto ocurre en contextos donde el mineral tiene una alta granulometría o dureza. En general la decisión que se toma es disminuir el porcentaje de sólidos y aumentar las rpm, pero esto en algunas ocasiones genera que la celda de carga del molino supere o este muy cercano a su límite superior (**setpoint HH celda de carga**), lo cual activa el sistema de control reduciendo la cantidad de mineral que ingresa al molino, lo que a su vez genera caídas de TPH respecto de su objetivo (**setpoint TPH**). Lo mencionado anteriormente se esquematiza en la figura 2.1.

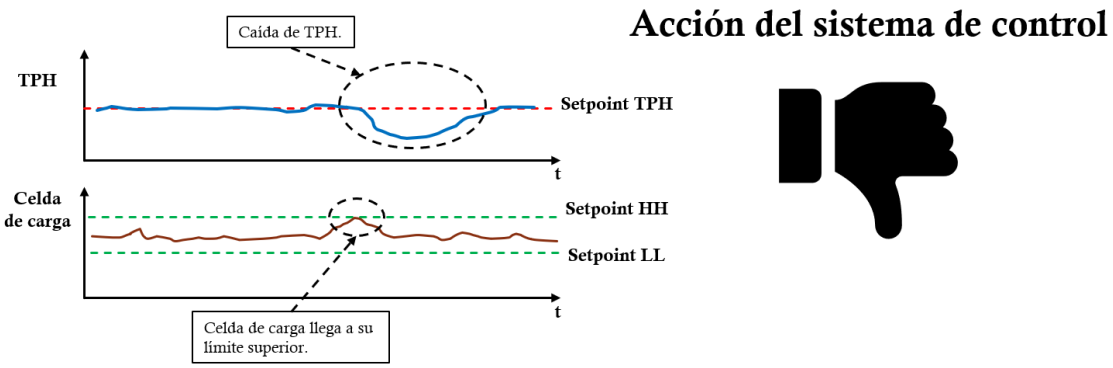

Figura 2.1: Esquema de caídas de TPH por acción del sistema de control.

En este contexto, la pregunta es si se podría haber subido el setpoint HH celda de carga del molino con el objetivo de lograr el procesamiento del mineral pero sin este tipo de caídas, las cuales se caracterizan por ser de alto impacto respecto de las caídas por embancamiento.

ii) **Caídas por embancamiento**: Esto ocurre en contextos en que el molino se encuentra demasiado lleno, es decir con un alto nivel de celda de carga. Es importante entender que este tipo de caídas de TPH no se deben a la acción del sistema de control, si no que es netamente por restricciones físicas del proceso. Lo mencionado anteriormente se esquematiza en la figura 2.2.

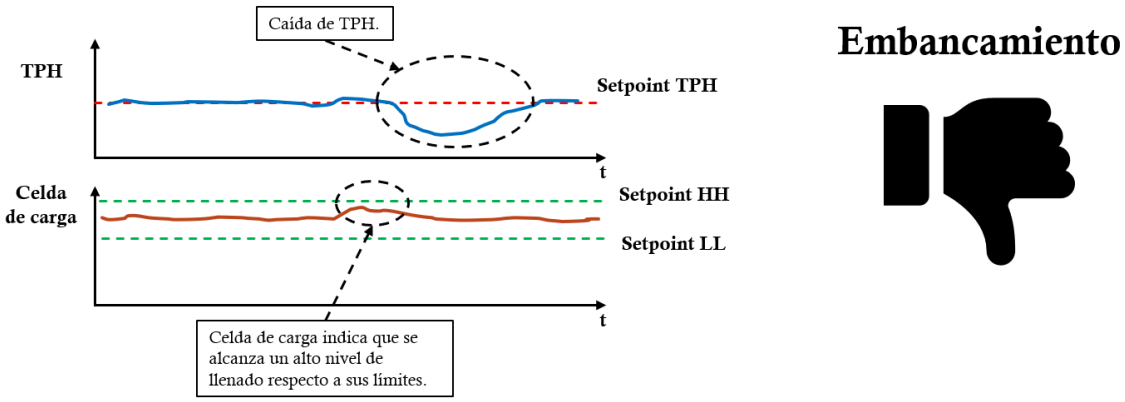

Figura 2.2: Esquema de caídas de TPH por embancamiento.

El comportamiento mencionado anteriormente se puede explicar teóricamente a partir de la denominada **curva de molienda** [4], la cual consiste en graficar todos los puntos operacionales del molino SAG a través de su TPH v/s celda de carga. Cabe destacar que esta curva nunca estará completa, es decir, no tendrá todo su dominio explorado, ya que solo se puede construir a partir de los puntos operacionales que efectivamente ocurrieron.

La figura 2.3 representa la curva teórica de molienda, la cual consiste en una parábola de concavidad negativa, en donde los puntos operacionales permitidos del molino

SAG se encuentran en el límite e interior de la curva. Es importante destacar que esta curva será estática siempre y cuando el contexto del molino sea el mismo, es decir, para un mismo nivel de granulometría, dureza y desgaste de sus revestimientos.

A través de la curva de molienda se visualiza la idea de como el TPH del molino SAG se puede ver afectado de manera negativa si es que se alcanzan niveles de llenado demasiado altos o bajos. Para el caso de niveles demasiado bajos, debido a que la cantidad de mineral entrante no es suficiente, no es posible alcanzar el TPH objetivo debido al bajo tonelaje procesado. Por otro lado, cuando el nivel de llenado es demasiado alto, no es posible una colisión eficiente entre partículas y bolas de acero, lo que provoca que los tiempos de residencia de las partículas aumente, lo que en consecuencia reduce la velocidad de salida del mineral, afectando negativamente el TPH del molino.

Por otro lado, la línea recta roja indica el setpoint TPH que define el operador sobre el molino, el cual en general esta bajo el óptimo de la curva de molienda debido a restricciones técnicas y operacionales.

Finalmente, es importante destacar que se aspira a operar en el punto estrellado que representa la intersección entre el setpoint TPH y el punto de la curva de molienda que permite tener el mayor nivel de llenado del molino. Esto último debido a que en niveles de llenado demasiado bajos el mineral dentro del molino podría llegar a golpear los revestimientos del molino, lo que puede provocar una reducción en la vida útil de estos.

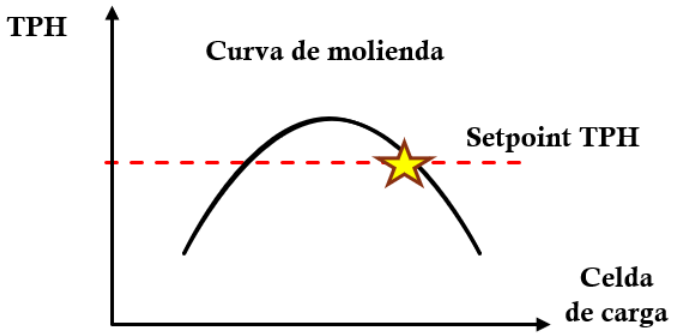

Figura 2.3: Curva de molienda.

Para diferenciar a que se deben las caídas de TPH, se debe observar la celda de carga minutos previos a la caída, con el objetivo de visualizar si la celda de carga estaba demasiado cerca de su setpoint HH. Si esto es así, la caída se debe a la acción del sistema de control, si no es así, se debe a una caída por embancamiento del molino. Adicionalmente en la figura 2.4, se observa el caso óptimo en que no existen caídas de TPH.

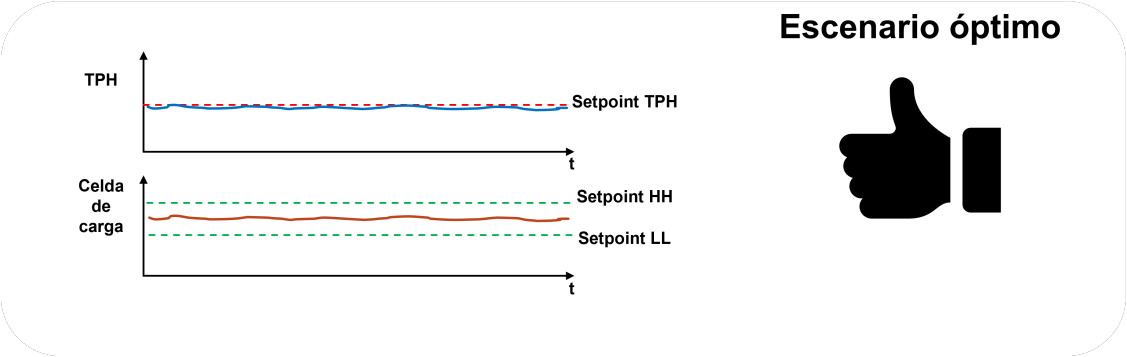

Figura 2.4: Escenario operativo óptimo.

El objetivo de la presente tesis se centrará netamente en evitar las caídas de TPH por embancamiento.

2.2. Modelo estadístico

El primer modelo que buscó encontrar una solución a la recomendación del setpoint HH celda de carga es el denominado **modelo estadístico** (o modelo base), el cual fue desarrollado por Felipe Contreras (Gerente Processing Analytics) y Carlos Orellana (Lead Data Scientst).

La dificultad de dar una recomendación de celda de carga radica en que generalmente se enfoca en maximizar TPH para un contexto dado, pero esto es difícil de operativizar debido a la variabilidad del contexto del molino. La solución propuesta del modelo estadístico se basa en dar una recomendación de celda de carga que mantenga el TPH constante respecto de su setpoint.

Este modelo utiliza la operación histórica del molino SAG. El modelo se basa en la clusterización de variables mineralógicas e identificación del nivel de desgaste de los revestimientos del molino SAG definido por 3 niveles (0, 1, 2), para luego acceder a ciertas curvas de recomendación en función de la granulometría entrante.

La recomendación del modelo esta definido principalmente por la dureza y granulometría del material de entrada, y del nivel de desgaste de los revestimientos del molino, siendo esto último definido como la edad del molino.

El principio de funcionamiento es el siguiente, se clusterizan los datos mineralógicos en 4 clusters codificados de 0 a 3 (blando, medio blando, medio duro y duro) y la edad del molino se codifica de 0 a 2, estando definida según la última fecha de instalación de los revestimientos. El cluster y la edad del molino definen un **subcontexto** el cual permite acceder a una curva de recomendación de celda de carga en particular, que depende de la granulometría de entrada.

En resumen, para obtener una recomendación de celda de carga, se necesita un subcontexto (definido por el cluster y la edad del molino) y la granulometría de entrada, siendo ambas (subcontexto y granulometría) lo que permite obtener el contexto operacional del

molino para obtener la recomendación de celda de carga. Ver figura 2.5.

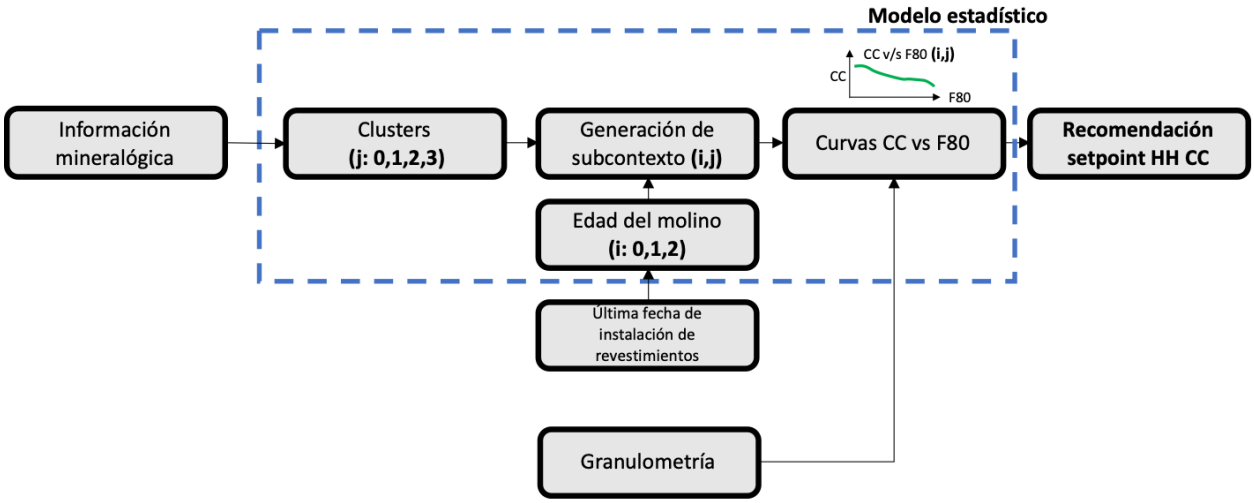

Figura 2.5: Esquema de funcionamiento modelo estadístico.

En la figura 2.6 se muestran las 12 curvas de recomendación (combinatoria de 4 clusters y 3 edades del molino) a las que accede el modelo para generar sus recomendaciones. Como se puede observar estas curvas poseen una alta variabilidad en la medida de que la granulometría cambia.

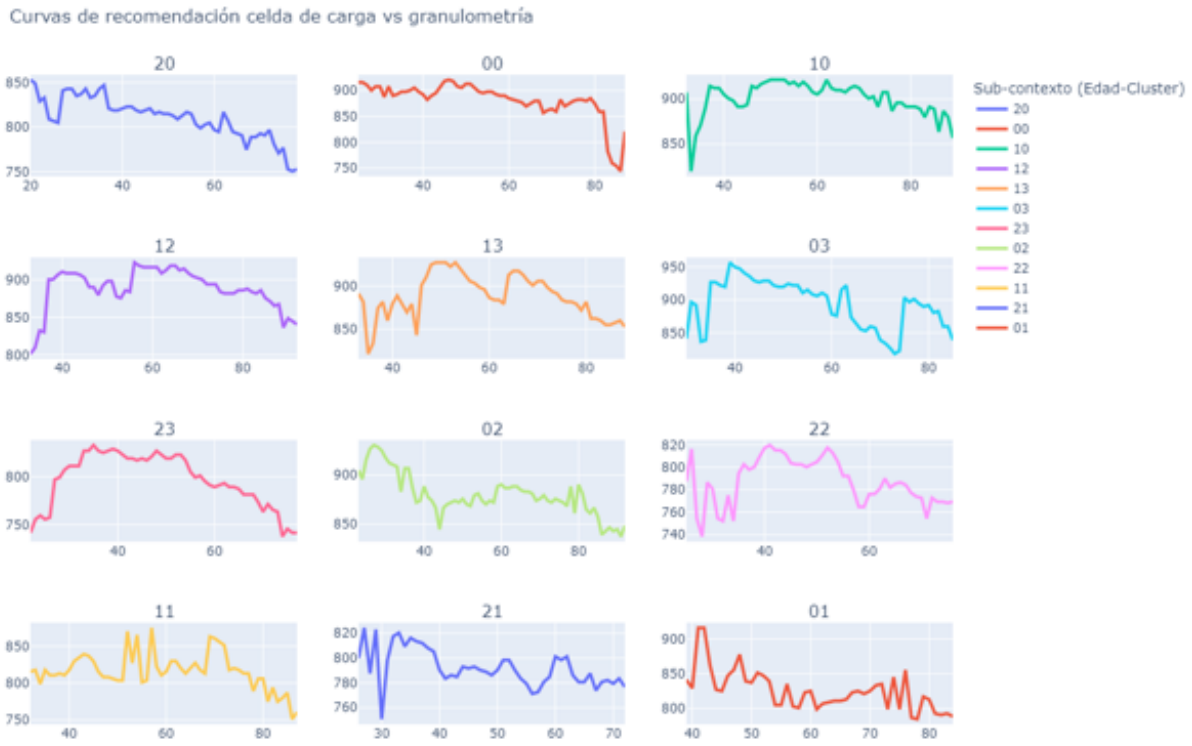

Figura 2.6: Curvas de recomendación en función de granulometría.

La lógica para obtener estas curvas es mediante la siguiente metodología:

1. División de los datos a través de las 12 combinaciones posibles de clusters y edad de molino.
2. Dividir nuevamente los datos iterando por cada rango granulométrico de ancho 1, es decir, rango del estilo [40,41), [41,42), [42,43) etc, hasta llegar al rango [99,100).
3. Para cada subdivisión generar gráfico de celda de carga v/s TPH (ver figura 2.7).
4. Calcular función envolvente mediante el calculo del promedio del percentil 95 de TPH para cada rango de celda de carga.
5. Calcular el valor de celda de carga en el cual se considera que la función envolvente de TPH comienza a caer. Este valor será la recomendación para el rango granulométrico particular. Este enfoque apunta hacia la **maximización de TPH** para cada nivel de celda de carga en particular.

En la figura 2.7 se muestra el principio constructivo de la curva 20 (edad del molino 2 y cluster 0), en donde se aprecia como se obtiene el valor de corte de celda de carga a recomendar para cada cúmulo de puntos asociado a cada rango granulométrico de la curva de recomendación en particular.

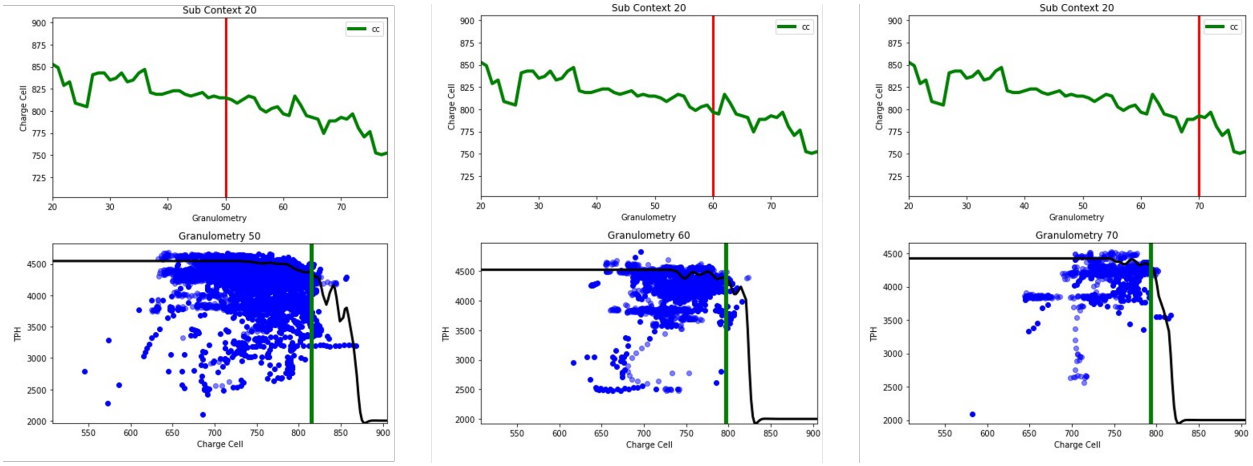

Figura 2.7: Método de obtención de curvas de recomendación de modelo estadístico.

Una vez construidas las 12 curvas de recomendación ("proceso de entrenamiento" del modelo estadístico), para obtener la recomendación de celda de carga en línea, minuto a minuto se clusterizan las propiedades mineralógicas del mineral de entrada, luego se accede a la edad del molino, lo cual permite acceder a 1 de las 12 curvas de recomendación, por ejemplo, la curva 20. Finalmente, para acceder a la recomendación también se debe consultar la granulometría de entrada en ese instante.

Para un subcontexto en particular, siguiendo con el ejemplo de la curva 20. Se observa que la curva sugiere que en la medida de que el mineral sea más grueso, es decir, disminuya el porcentaje pasante de mineral, se puede aumentar el límite alto de celda de carga. Esto

tiene sentido operacional, ya que en cambios de contextos en los cuales el mineral de entrada es más grueso, la estrategia operacional es ingresar más agua al molino SAG con el objetivo de moler de manera más eficiente el mineral, por lo que se le debe dar más espacio al molino para ser llenado. La figura 2.8 esquematiza este cambio de recomendación al ingresar mineral más grueso.

Si se sigue la recomendación del modelo para el subcontexto "20", se tiene la posibilidad de subir el límite alto de la celda de carga, lo que permite trabajar con el molino mas pesado, sin perder TPH. En el caso de no seguir la recomendación, es muy probable que al entrar mineral más grueso, el proceso de molienda se demore más, subiendo el peso del molino, el que al llegar al límite alto no recomendado, bajará TPH de manera automática, solo por una incorrecta definición del rango permitido de la celda de carga, no por razones operacionales ni de capacidad del molino.

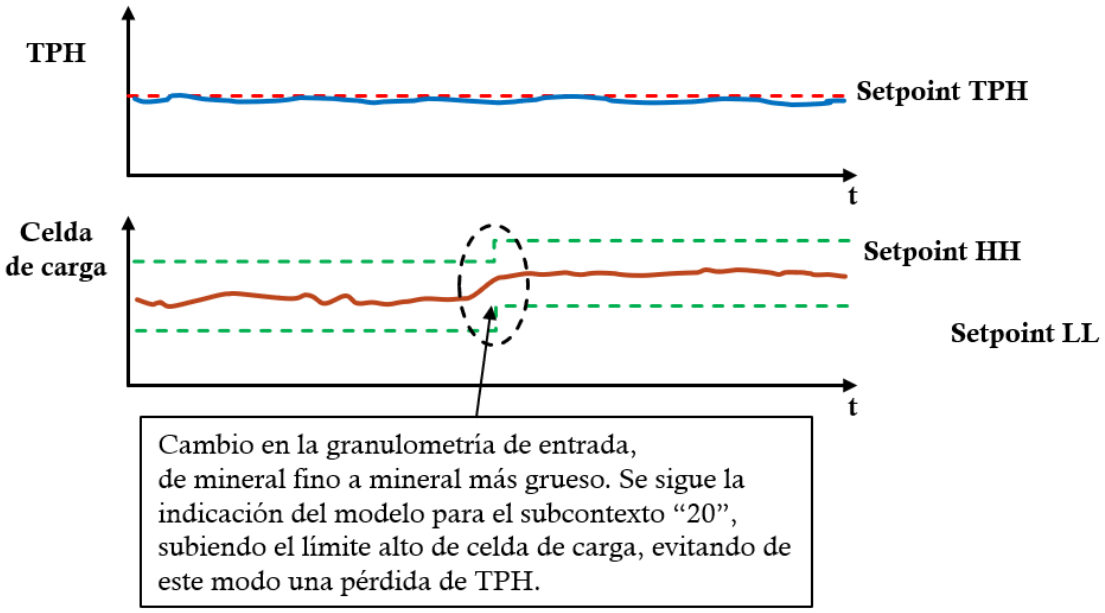

Figura 2.8: Cambio de recomendación debido a mineral entrante más grueso.

Sin embargo, es importante destacar que de igual forma, acorde al input del modelo, podría darse el caso en que si bien, el mineral entrante sea más grueso o más duro, no se pueda subir subir el nivel de celda de carga debido a las restricciones mismas del proceso de molienda.

Es importante entender que el subcontexto del molino (categorización cluster y edad) no debiese cambiar dentro de al menos una hora, sin embargo, la granulometría es un dato que varia en cada minuto. Debido a esto la recomendación cambiara en cada minuto y sumado a la variabilidad intrínseca de las curvas de recomendación, se alcanza una alta variabilidad en la recomendación para un periodo de tiempo en particular (por ejemplo 1 hora), lo cual dificulta la adherencia de la operación a la naturaleza de las indicaciones del modelo.

Dada la variabilidad de las recomendaciones que se podían llegar a generar dentro de un periodo de tiempo, Accenture hizo esfuerzos para mejorar el modelo estadístico siguiendo lógicas análogas, pero con la diferencias de que las 12 curvas de recomendación se construyen

a partir de pérdida de TPH v/s celda de carga, por lo que los puntos de corte nacen a partir de la **minimización de pérdidas** (las recomendaciones son la pre-imagen del mínimo de pérdidas para cada rango granulométrico), de este modo, logrando reducir la variabilidad de dichas curvas, obteniéndose así el denominado **modelo estadístico mejorado**. Ver figura 2.9.

Figura 2.9: Curvas suavizadas de recomendación en función de granulometría.

2.3. Aprendizaje Reforzado

2.3.1. Conceptos básicos [1]

1. **Agente**: Es una entidad (algoritmo) que interactúa con un entorno con el objetivo de aprender cómo tomar decisiones que maximicen alguna medida de recompensa a lo largo del tiempo. El agente ejecuta acciones de acuerdo a su **política**. La ejecución de estas acciones generan recompensas o castigos para el agente, y tienen consecuencias (a priori inciertas), dado que podrían afectar el estado del ambiente, y así, futuras acciones y recompensas.

2. **Política**: La política del agente es una estrategia o un conjunto de reglas que dicta cómo el agente debe seleccionar acciones en función de las observaciones del entorno. Puede ser determinista o estocástica, dependiendo de si la política siempre toma la misma acción en una situación dada o si tiene cierta incertidumbre. Matemáticamente, una política Π

es una distribución de acciones dado los estados:

$$\Pi(a|s) = P[A_t = a|S_t = s] \tag{2.1}$$

3. **Recompensa**: Una recompensa r_t es una señal de retroalimentación escalar que indica qué tan bien está despeñándose el agente en el paso t. La tarea del agente es maximizar el valor esperado de su **retorno** $J_{RL}(\pi)$ (o **suma descontada de recompensas**). Donde:

$$J_{RL}(\pi) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=1}^{T} \gamma^{t-1} r(s_t, a_t) \right] \tag{2.2}$$

4. **Factor de descuento**: $\gamma \in [0, 1]$, es un parámetro utilizado para determinar cómo se ponderan las recompensas futuras en relación con las recompensas inmediatas. En otras palabras, el factor de descuento determina cuánto valor se le da a las recompensas a largo plazo en comparación con las recompensas que se pueden obtener en el siguiente paso.

   a) Si $\gamma = 0$, el agente solo se preocupa por la recompensa inmediata en el siguiente paso y no considera las recompensas futuras. Esto se llama un enfoque de "descuento cero".

   b) Si $\gamma = 1$, el agente valora las recompensas futuras de la misma manera que las recompensas inmediatas. Esto implica una consideración completa de las recompensas a largo plazo.

5. **Estado**: El estado del ambiente s_t es la representación interna que tiene el agente del ambiente, es decir, es la representación del entorno observable para el agente y utiliza esta información para seleccionar la próxima acción a tomar. Los estados pueden ser discretos o continuos.

6. **Acción**: las acciones a_t son las decisiones que toma un agente en un entorno para interactuar y lograr sus objetivos. Las acciones representan las elecciones que el agente puede hacer en cada paso de tiempo para influir en el estado del entorno y, en última instancia, obtener recompensas. Las acciones pueden ser discretas o continuas.

7. **Hipótesis de recompensa**: Todo objetivo puede ser descrito a través de la maximización de la recompensa acumulada.

8. **Dilema de exploración y explotación**: Se refiere a la decisión que un agente debe tomar entre explorar nuevas opciones y acciones desconocidas o explotar las acciones que hasta ahora han demostrado ser efectivas en términos de recompensas.

2.3.2. Introducción al Aprendizaje Reforzado [1]

El Aprendizaje Reforzado se puede resumir como aprendizaje en base a prueba y error, y se basa en el marco de los **Procesos de Decisión de Márkov (**MDP**)**. Ver figura 2.10.

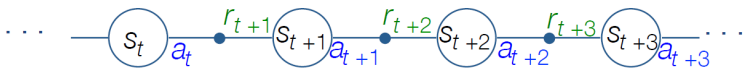

Figura 2.10: Ilustración de un Proceso de Decisión de Márkov (MDP).

Las políticas de los MDP dependen del estado actual, no de la historia, es decir, las políticas son estacionarias (independientes del tiempo), es decir, $A_t \sim \Pi(\cdot|S_t)$, $\forall t > 0$. Dichos estados se dice que poseen la **"propiedad de Márkov"**. **"Dado el presente, el futuro no depende del pasado"**.

Un MDP esta definido por la tupla $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$, donde:

1. $\mathcal{S}$: Conjunto de estados.
2. $\mathcal{A}$: Conjunto de acciones.
3. $\mathcal{T}$: Función de transición de estados.
4. $\mathcal{R}$: Función de recompensa.

La formalización clásica para modelar problemas de toma de decisiones secuenciales para $t = 1, ..., T$ es tal que:

1. El agente se encuentra en un estado $s_t \in \mathcal{S}$.
2. El agente elige y ejecuta una acción $a_t \in \mathcal{A}$ de acuerdo a su política $\Pi(a_t|s_t)$.
3. El ambiente evoluciona a nuevo estado $s_{t+1} \in \mathcal{S}$ de acuerdo a la función de transición $\mathcal{T}(s, a, s') = p(s_{t+1}|s_t, a_t)$ y el agente recibe una recompensa escalar r_t, de acuerdo a la función de recompensa $\mathcal{R}(s, a)$.

De un modo más formal, el Aprendizaje Reforzado aborda el problema de cómo un **agente** activo y autónomo aprende **políticas óptimas** mientras interactúa con un **entorno o ambiente** inicialmente desconocido. El objetivo de resolver un MDP es encontrar la política óptima.

Conceptos claves para el modelamiento de un MDP lo son la **Función de Valor V**, la **Función Q** y las **Ecuaciones de Bellman**:

1. **Función de valor** $V^\pi(s)$: La función de valor V asigna un valor numérico a cada estado en el espacio de estados del entorno, indicando cuán deseable o beneficioso es estar en ese estado. Esta función refleja la cantidad de recompensa acumulada que un agente puede esperar recibir en el futuro a partir de un estado dado, siguiendo una determinada política.

$$V^\pi(s) = \mathbb{E}_{\tau_t \sim p_\pi(\tau_t)} \left[\sum_{k=t}^{T} \gamma^{k-t} r(s_k, a_k) \middle| s_t = s \right] \tag{2.3}$$

2. **Función** $Q^\pi(s, a)$: La función Q es utilizada para medir el valor esperado acumulado que un agente puede obtener al tomar una determinada acción en un estado particular

y luego seguir una política específica para el resto de la secuencia de acciones. El uso de funciones de valor estado-acción elimina la necesidad de almacenar políticas de manera explícita o de conocer el modelo (probabilidades de transición de estado).

$$Q^{\pi}(s,a) = \mathbb{E}_{\tau_t \sim p_{\pi}(\tau_t)} \left[\sum_{k=t}^{T} \gamma^{k-t} r(s_k, a_k) \middle| s_t = s, a_t = a \right] \tag{2.4}$$

3. **Ecuaciones de Bellman**: Para resolver un problema de Aprendizaje Reforzado, se debe considerar el retorno que el agente podría recibir de acuerdo a las acciones que este ejecuta. Notando que $J_{RL}(\pi) = \mathbb{E}_{s_1 \sim p(s)}[V^{\pi}(s_1)]$, buscamos una política que tenga asociada una función de valor óptima:

$$V^{*}(s) = \max_{\pi} V^{\pi}(s) \tag{2.5}$$

Existe una relación directa entre $V^{\pi}(s)$ y $Q^{\pi}(s,a)$:

$$V^{\pi}(s) = \mathbb{E}_{a \sim \pi(a|s)}[Q^{\pi}(s,a)] \tag{2.6}$$

$$Q^{\pi}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}[V^{\pi}(s')] \tag{2.7}$$

Por lo que también es posible resolver el problema de Aprendizaje Reforzado a través de:

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{2.8}$$

De lo anterior, se pueden establecer relaciones de recurrencia para $V^{\pi}(s)$ y $Q^{\pi}(s,a)$, las cuales se conocen como **"Ecuaciones de Bellman"**:

$$V^{\pi}(s) = \mathbb{E}_{a \sim \pi(a|s)} \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}[V^{\pi}(s')] \right] \tag{2.9}$$

$$Q^{\pi}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} \left[\mathbb{E}_{a' \sim \pi(a'|s')}[Q^{\pi}(s',a')] \right] \tag{2.10}$$

Tanto $V^{*}(s)$ como $Q^{*}(s,a)$ satisfacen su respectiva ecuación de Bellman, no obstante, en este caso pueden ser escritas de manera especial considerando la definición de las funciones de valor óptimas. Estas ecuaciones se conocen como **"ecuaciones de optimalidad de Bellman"**:

$$V^{*}(s) = \max_{a \in A} \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}[V^{*}(s')] \right] \tag{2.11}$$

$$Q^{*}(s,a) = r(s,a) + \gamma \max_{a' \in A} \left[\mathbb{E}_{s' \sim p(s'|s,a)}[Q^{*}(s',a')] \right] \tag{2.12}$$

Métodos matemáticos clásicos para lograr aproximar estas funciones de manera eficiente lo son la **programación dinámica** 2.11 y los **métodos de Monte Carlo** 2.12:

Inicializar π
Inicializar umbral $\epsilon > 0$ y $\Delta v \leftarrow 0$
Inicializar $V^{\pi}(s)$ (aleatoriamente), pero con $V^{\pi}(s_{\text{terminal}}) \leftarrow 0$
while $\Delta v < \epsilon$ **do**
    $\Delta v \leftarrow 0$
    foreach $s \in \mathcal{S}$ **do**
        $v_s \leftarrow V^{\pi}(s)$
        $V^{\pi}(s) = \max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} V^{\pi}(s') \right]$
        $\Delta v \leftarrow \max\{\Delta v, |v_s - V^{\pi}(s)|\}$
    end
end

Figura 2.11: Ejemplo de algoritmo de programación dinámica.

Inicializar una política $\pi(s)$
Inicializar $Q^{\pi}(s,a)$ aleatoriamente
Inicializar $S(s,a) \leftarrow 0$ y $N(s,a) \leftarrow 0$ para cada $s \in \mathcal{S}, a \in \mathcal{A}$
for *episodio = 1, N* **do**
    Seleccionar aleatoriamente (s_1, a_1)
    Generar *rollout* siguiendo política π, desde (s_1, a_1): $s_1, a_1, r_1, s_2, ...s_{T-1}, a_{T-1}, r_{T-1}, s_T$
    $R \leftarrow 0$
    for *t=T-1, 1* **do**
        $R \leftarrow r_t + \gamma R$
        if $(s_t, a_t) \notin \{s_1, a_1, ..., s_{t-1}, a_{t-1}\}$ **then**
            $N(s_t, a_t) \leftarrow N(s_t, a_t) + 1$
            $S(s_t, a_t) \leftarrow S(s_t, a_t) + R$
        end
    end
    $Q^{\pi}(s,a) \leftarrow \frac{S(s,a)}{N(s,a)}$ para todo $s \in \mathcal{S}, a \in \mathcal{A}$
    $\pi(s_t) \leftarrow \arg\max_{a \in \mathcal{A}} Q^{\pi}(s_t, a)$
end

Figura 2.12: Ejemplo de algoritmo de Monte Carlo.

2.3.3. Taxonomía de algoritmos de Aprendizaje Reforzado [1]

Los algoritmos de RL pueden ser clasificados dentro de 4 categorías, siendo no siempre excluyentes una de la otra.

1. **Value-Based**: Aproximan $V^{\star}(s)$ o $Q^{\star}(s,a)$ para derivar una política.
2. **Policy-Based**: Buscan $\Pi(a|s)$ a través de la optimización directa de $J_{RL}(\pi)$.
3. **Actor-Critic**: Aproximan conjuntamente $V^{\star}(s)$ o $Q^{\star}(s,a)$ y una política $\Pi(a|s)$.
4. **Model-Based & Model-Free**: Hacen o no uso de un modelo del ambiente.

Por ejemplo, **Q-Learning** y **SARSA** son algoritmos **Value-based & Model-Free**, ya que aproximan funciones Q y no necesitan un modelo del entorno. **REINFORCE** [5] es un ejemplo de algoritmo **Policy-Based & Model-Free**, ya que representa su política $\Pi_{\theta}(a|s)$ explícitamente a través una red neuronal optimizando sus parámetros θ mediante gradiente ascendente según $\nabla J_{RL}(\pi_{\theta})$. Por otro lado, el algoritmo **Actor-Critic**, es de la familia que indica su nombre, esto debido a que posee una red neuronal para aproximar funciones Q, llamada **"Critic"** y otra red neuronal que aprende políticas óptimas llamada **"Actor"**. Ver figura 2.13 y 2.14 para facilitar el entendimiento de la taxonomía de estos algoritmos.

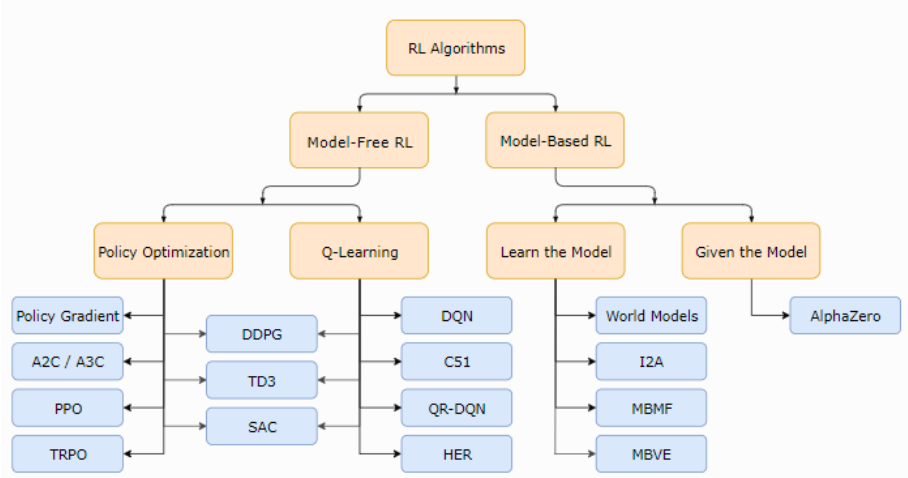

Figura 2.13: Taxonomía de algoritmos de Reinforcement Learning (OpenAI).

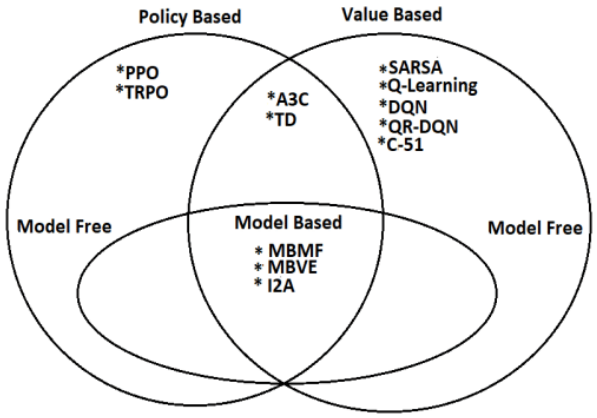

Figura 2.14: Taxonomía de algoritmos de Reinforcement Learning en diagrama de Venn.

Capítulo 3

Estado del arte

3.1. Aprendizaje reforzado

3.1.1. Algoritmos de Aprendizaje Reforzado [1]

1. **Value-Based**: Dentro de los algoritmos Value-Based tenemos los algoritmos por aprendizaje temporal conocidos como **Temporal-Difference Learning (TD-Learning)**, estos son una combinación de ideas entre programación dinámica y métodos de Monte Carlo. **Q-Learning** y **SARSA** son ejemplos de algoritmos **TD-Learning**, en la figura 3.1 e 3.2 se pueden ver sus algoritmos respectivos.

Inicializar $Q^{\pi}(s, a)$ aleatoriamente
Inicializar parámetro $\alpha \in (0, 1]$ y $\epsilon > 0$
for *episodio = 1, N* **do**
  Obtener s_1
  for *t=1, T* **do**
    Con probabilidad ϵ, elegir a_t aleatoriamente, si no, $a_t = \arg\max_{a \in \mathcal{A}} Q_\theta(s_t, a)$
    Ejecutar acción a_t, observar r_t y s_{t+1}
    $Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha(r_t + \gamma \max_{a'} Q^{\pi}(s_{t+1}, a') - Q^{\pi}(s_t, a_t))$
  end
end

Figura 3.1: Algoritmo Q-Learning.

Inicializar $Q^{\pi}(s, a)$ aleatoriamente
Inicializar parámetro $\alpha \in (0, 1]$ y $\epsilon > 0$
for *episodio = 1, N* **do**
  Obtener s_1
  Con probabilidad ϵ, elegir a_1 aleatoriamente, si no, $a_1 = \arg\max_{a \in \mathcal{A}} Q_\theta(s_1, a)$
  for *t=1, T* **do**
    Ejecutar acción a_t, observar r_t y s_{t+1}
    Con probabilidad ϵ, elegir a_{t+1} aleatoriamente, si no, $a_{t+1} = \arg\max_{a \in \mathcal{A}} Q_\theta(s_{t+1}, a)$
    $Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha(r_t + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi}(s_t, a_t))$
  end
end

Figura 3.2: Algoritmo SARSA.

Es importante mencionar el concepto de **"Aprendizaje Reforzado Profundo"** (DRL), este surge de la combinación entre el **Aprendizaje Profundo** y el **Aprendizaje Re-**

forzado. El DRL se caracteriza por el uso de redes neuronales artificiales como aproximadores funcionales. El primer trabajo que popularizó el uso de redes neuronales en RL fue **Deep Q-Network (DQN)** [6]. En la figura 3.3 se muestra una ilustración que plasma la diferencia de ideas entre Q-Learning y DQN. Por otro lado, en la figura 3.4 se puede apreciar el detalle de su algoritmo.

DQN emplea una red neuronal $Q_\theta(s, a)$ para aproximar $Q^*(s, a)$. Esta red neuronal, $Q_\theta(s, a)$, es entrenada empleando una variante de Q-Learning que posee dos elementos fundamentales, un **Experience Replay** y el uso de **Target Networks**.

Las experiencias (s_t, a_t, r_t, s_{t+1}) son almacenadas en un buffer finito $\mathcal{D}$. Batches de experiencias son muestreados uniformemente de $\mathcal{D}$ para actualizar los parámetros de $Q_\theta(s, a)$, según:

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim U(D)} \left[(y - Q_{\theta_i}(s, a)) \nabla_{\theta_i} Q_{\theta_i}(s, a) \right] \tag{3.1}$$

Donde:

$$y = r(s, a) + \gamma \max_{a' \in A} \mathbb{E}_{s' \sim p(s'|s,a)} Q_{\theta_{i-k}}(s', a') \tag{3.2}$$

Lo anterior permite la reutilización de experiencias, y también romper la correlación que existe entre ellas.

Se utiliza una copia $Q_{\bar{\theta}}(s, a)$ de $Q_\theta(s, a)$ para el cómputo de y. Esta copia es actualizada copiando los parámetros de $Q_\theta(s, a)$ cada k > 1 actualizaciones. Este mecanismo reduce la posibilidad de oscilaciones (o divergencia) en los parámetros θ, al introducir un "delay" en el efecto que tienen los cambios en $Q_\theta(s, a)$ en el computo de y.

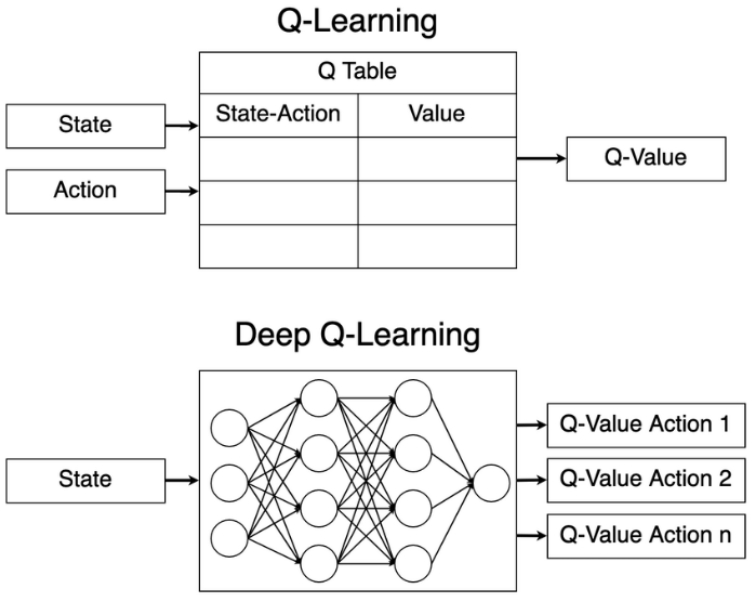

Figura 3.3: Diagrama Q-Learning vs Deep Q-Learning.

Inicializar $Q_\theta(s, a)$ con parámetros θ
Inicializar $Q_{\overline{\theta}}(s, a)$ con parámetros $\overline{\theta} \leftarrow \theta$
Inicializar *replay memory* D
for *episodio = 1, M* **do**
  Obtener s_1
  for *t=1, T* **do**
    Con probabilidad ϵ, elegir a_t aleatoriamente, si no, $a_t = \arg\max_{a \in \mathcal{A}} Q_\theta(s_t, a)$
    Ejecutar acción a_t, observar r_t y s_{t+1}
    guardar transición (s_t, a_t, r_t, s_{t+1}) en D
    Muestrear un *minibatch* de N transiciones (s_j, a_j, r_j, s_{j+1}) de D
    Calcular $y_j = \begin{cases} r_j & \text{si } s_{j+1} \text{ es un estado terminal,} \\ r_j + \gamma \max_{a'} Q_{\overline{\theta}}(s_{j+1}, a') & \text{si no.} \end{cases}$
    Actualizar $Q_\theta(s, a)$ minimizando el costo $L(\theta) = \frac{1}{N} \sum_{j=1}^{N} (y_j - Q_\theta(s_j, a_j))^2$
    Cada C actualizaciones de $Q_\theta(s, a)$, $\overline{\theta} \leftarrow \theta$
  end
end

Figura 3.4: Algoritmo DQN.

2. **Policy-Based**: Como se mencionó anteriormente, los algoritmos de esta familia se caracterizan por aprender distribuciones de probabilidad de acciones dados los estados a través de una red neuronal, mejorando su política mediante gradiente ascendente. En la figura 3.5 se plasma la idea de que la salida de la red es una distribución de probabilidad. Yendo más al detalle, la red entrega una media y desviación estándar que son parámetros de algún tipo de distribución, por ejemplo, una Gaussiana. En la figura 3.6 se muestra el algoritmo **REINFORCE** [5] .

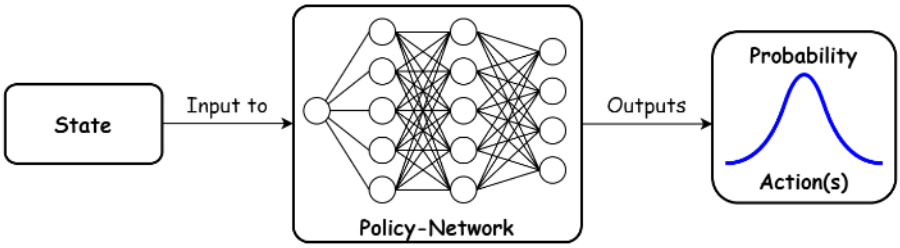

Figura 3.5: Diagrama de acciones generadas en algoritmos Policy-Based.

Inicializar $\pi_\theta(a|s)$ con parámetros θ
for *i=1, M* **do**
  for *k=1, N* **do**
    Obtener s_1
    for *t=1, T-1* **do**
      Ejecutar acción $a_t \sim \pi_\theta(a_t|s_t)$, observar r_t y s_{t+1}
      Guardar transición (s_t, a_t, r_t) en $\tau^{(k)}$
    end
  end
  Calcular $\nabla_\theta J_{\text{RL}} \approx \frac{1}{N} \sum_{k=1}^{N} \left[\left(\sum_{t=1}^{T} \nabla_\theta \log \left(\pi_\theta \left(a_t^{(k)} | s_t^{(k)} \right) \right) \right) \left(\sum_{t=1}^{T} \gamma^{t-1} r \left(s_t^{(k)}, a_t^{(k)} \right) \right) \right]$
  Actualizar $\pi_\theta(a|s)$ con gradiente ascendente según $\nabla_\theta J_{\text{RL}}(\pi_\theta)$
end

Figura 3.6: Algoritmo REINFORCE.

3. **Actor-Critic**: Esta familia de algoritmos es una mejora de los algoritmos Policy-Based

al reducir la alta varianza de la estimaciones del gradiente mediante la incorporación del termino **"Advantage Function"**:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{3.3}$$

$$A^{\pi}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} V^{\pi}(s') - V^{\pi}(s) \tag{3.4}$$

$$A^{\pi}(s,a) \approx r(s,a) + \gamma V^{\pi}(s') - V^{\pi}(s) \tag{3.5}$$

con esto:

$$\nabla_{\theta} J_{RL}(\pi_{\theta}) \approx \frac{1}{N} \sum_{k=1}^{N} \left(\sum_{t=1}^{T} \nabla_{\theta} \log \left(\pi_{\theta} \left(a_t^{(k)} | s_t^{(k)} \right) \right) \underbrace{\left(Q^{\pi} \left(s_t^{(k)}, a_t^{(k)} \right) - V^{\pi} \left(s_t^{(k)} \right) \right)}_{A^{\pi}(s^{(k)}, a^{(k)})} \right) \tag{3.6}$$

Advantage Function es una medida que cuantifica cuán favorable es tomar una acción en comparación con las acciones promedio disponibles en un estado específico bajo una política dada. Según su signo tiene la siguiente interpretación:

a) **Acciones más favorables**: Si $A^{\pi}(s,a) > 0$, significa que la acción a es mejor de lo que se esperaría de las acciones promedio en ese estado bajo la política Π. En este caso, tomar la acción a es más ventajoso que tomar las acciones promedio.

b) **Acciones menos favorables**: $A^{\pi}(s,a) < 0$, la acción a es peor de lo que se esperaría de las acciones promedio en ese estado bajo la política Π. En este caso, tomar la acción a es menos ventajoso que las acciones promedio.

Bajo este concepto surge la necesidad de aproximar $V^{\pi}(s)$ mediante $V_{\phi}(s)$, esto a través de una red neuronal denominada **"Critic"**, por otro lado, la red neuronal encargada de aprender la distribución de probabilidad de acciones dado los estados, es denominada **"Actor"**. En la figura 3.7 se muestra un ejemplo sencillo del algoritmo Actor-Critic básico.

Inicializar $\pi_{\theta}(a|s)$ con parámetros θ
Inicializar $V_{\phi}(s)$ con parámetros ϕ
for *i=1, M* **do**
  for *k=1, N* **do**
    Obtener s_1
    for *t=1, T-1* **do**
      Ejecutar acción $a_t \sim \pi_{\theta}(a_t|s_t)$, observar r_t y s_{t+1}
      Guardar transición (s_t, a_t, r_t) en $\tau^{(k)}$
    end
  end
  Ajustar ϕ minimizando $L(\phi) = \frac{1}{TN} \sum_{k=1}^{N} \sum_{t=1}^{T} \left(V_{\phi} \left(s_t^{(k)} \right) - \sum_{t'=t}^{T} \gamma^{t'-t} r \left(s_{t'}^{(k)}, a_{t'}^{(k)} \right) \right)^2$
  Calcular $\nabla_{\theta} J_{\text{RL}} \approx \frac{1}{N} \sum_{k=1}^{N} \left[\sum_{t=1}^{T} \nabla_{\theta} \log \left(\pi_{\theta} \left(a_t^{(k)} | s_t^{(k)} \right) \right) A^{\pi} \left(s_t^{(k)}, a_t^{(k)} \right) \right]$
  Actualizar $\pi_{\theta}(a|s)$ con gradiente ascendente según $\nabla_{\theta} J_{\text{RL}}(\pi_{\theta})$
end

Figura 3.7: Algoritmo Actor-Critic.

3.1.2. Clasificación de algoritmos de Aprendizaje Reforzado según temporalidad de aprendizaje

Los algoritmos **On-Policy** aprenden de las decisiones tomadas por la política actual que están siguiendo, mientras que los algoritmos **Off-Policy** aprenden de decisiones tomadas por una política diferente a la que están evaluando y mejorando [1].

Por ejemplo, Q-Learning es Off-Policy pues actualiza Q siguiendo una politica greedy, pero la política que genera las transiciones es epsilon-greedy. En contraste, SARSA es On-Policy pues es consistente con la estrategia epsilon-greedy al momento de realizar actualizaciones.

En la figura 3.8 (a) se observa que las experiencias generadas a través de interacciones agente-ambiente son utilizadas para actualizar la política. Tras actualizar Π_k se deben generar nuevos datos para poder realizar una nueva actualización. Dichas actualizaciones son generadas por Π_k.

En la figura 3.8 (b) las interacciones agente-ambiente son guardadas en un replay buffer $\mathcal{D}$. Las experiencias de $\mathcal{D}$ son muestreadas para realizar actualizaciones de la política. De este modo $\mathcal{D}$ presenta experiencias generadas por Π_0, ...Π_k.

Por otro lado, los algoritmos **Offline** (**Aprendizaje Reforzado Fuera de Línea**) [2] hacen referencia a aprender políticas sin interacciones agente-ambiente, las experiencias provienen de un dataset fijo $\mathcal{D}$. En principio cualquier algoritmo Off-Policy puede ser transformado a un algoritmo Offline, por ejemplo, aplicando Q-Learning a un dataset fijo (buffer).

Los desafíos del Aprendizaje Reforzado Fuera de Línea, son la exploración. El dataset $\mathcal{D}$ es fijo, por lo que la exploración queda fuera de lo que un algoritmo Offline RL puede abordar. Además, obtener un dataset $\mathcal{D}$ que sea representativo puede ser muy complejo. En la figura 3.8 (c) las experiencias para aprender una política son muestreadas de un buffer $\mathcal{D}$. Las experiencias en $\mathcal{D}$ fueron generadas por una política Π_β (que puede ser desconocida). No existe interacción agente-ambiente durante el aprendizaje de Π.

En Aprendizaje Reforzado Fuera de Línea se requiere formular y responder preguntas del tipo "¿Qué hubiese ocurrido si...?" pues se quiere obtener una política Π que sea mejor que Π_β, es decir, queremos aprender algo mejor que lo que se observa en $\mathcal{D}$. Lo anterior se traduce en un problema de cambio de distribución entre la política que recopiló los datos y la política aprendida (**Distributional shift**).

Distributional shift se refiere a la discrepancia entre las distribuciones de datos que se utilizan para entrenar el modelo y las distribuciones de datos que se encuentran en el ambiente en el que el modelo se desplegará. En muchos casos, los datos históricos pueden provenir de un entorno o una política pasada que difiere significativamente del entorno actual. Esta discrepancia entre las distribuciones de datos puede resultar en una mala generalización y un rendimiento deficiente cuando se despliega el modelo en el mundo real.

El problema de distributional shift puede llevar a que el modelo tenga dificultades para adaptarse a las nuevas situaciones y tomar decisiones precisas en el entorno actual. Esto se

debe a que el modelo no ha visto suficientes ejemplos de las situaciones actuales durante su entrenamiento, lo que puede resultar en comportamientos inesperados o subóptimos.

Para abordar este problema, es importante desarrollar métodos de Aprendizaje Reforzado Fuera de Línea que sean más resistentes al cambio en la distribución y que puedan generalizar de manera efectiva a situaciones no vistas previamente.

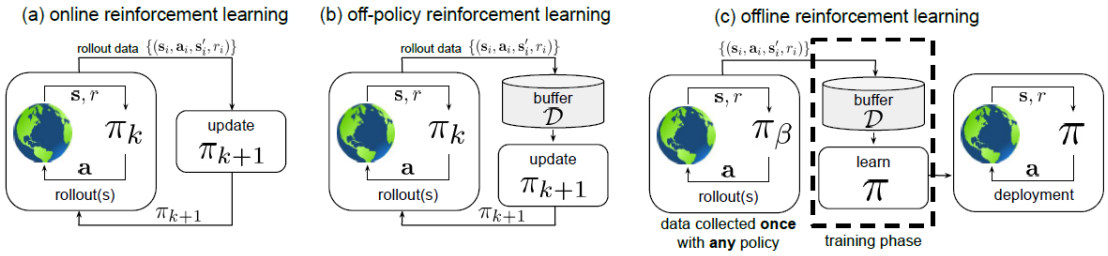

Figura 3.8: Ilustración de Aprendizaje Reforzado On-Policy (a), Aprendizaje Reforzado Off-Policy (b) y Aprendizaje Reforzado Fuera de Línea (c) [2].

3.1.3. Formulación del problema asociado a Aprendizaje Reforzado fuera de Línea

A continuación se explican los algoritmos **Conservative Q-learning (CQL)** y **Advantage-Weighted-Actor-Critic (AWAC)** con el objetivo de mejorar el entendimiento del enfoque del Aprendizaje Reforzado Fuera de Línea.

1. **Conservative Q-Learning (CQL)** [7]: CQL es la adaptación de DQN en un ambiente Offline, esto debido a que utilizar directamente algoritmos de RL Value-Based Off-Policy existentes en un entorno Offline generalmente resulta en un rendimiento deficiente, producto de que se generan problemas relacionados con el inicio de acciones fuera de la distribución y el overfitting. Esto suele manifestarse como estimaciones de la función de valor erróneamente optimistas. Si, en cambio, se pudiese aprender una estimación conservadora de la función de valor, que proporcione un límite inferior sobre los valores reales, este problema de sobre-estimación podría resolverse.

   Dado un batch de transiciones $\beta = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^{N}$ definiendo la regla de actualización de la función Q como:

$$L(\beta, \theta) = \sum_{i=1}^{N} \left(r_i + \gamma \max_{a' \in A} Q_\theta(s'_i, a') - Q_\theta(s_i, a_i) \right)^2 \tag{3.7}$$

   En CQL se busca modificar este objetivo agregando una penalización $\mathcal{C}(B, \theta)$ para obtener un nuevo objetivo de minimización, dado por:

$$\tilde{L}(\beta, \theta) = L(\beta, \theta) + \alpha \mathcal{C}(\beta, \theta) \tag{3.8}$$

   Ejemplo de penalización:

$$\mathcal{C}(\beta, \theta) = \mathbb{E}_{s \sim \beta} \left[\mathbb{E}_{a \sim \mu(a|s)} \left[Q_\theta(s, a) \right] \right] \tag{3.9}$$

Con una correcta elección de $\mu(a|s)$, seria posible reducir Q-values altos, y mantener aproximaciones razonables para aquellos asociados a acciones que están dentro de la distribución.

La elección $\mu(a|s) \propto \exp(Q_\theta(s,a))$ permite que la estimación regularizada de la función Q sea una cota inferior de la función Q real.

En la práctica la elección anterior de $\mathcal{C}(\beta,\theta)$ puede resultar en sub-estimaciones muy grandes, por lo que otra opción es definirla de manera diferente:

$$\mathcal{C}(\beta,\theta) = \mathbb{E}_{s\sim\beta}\left[\mathbb{E}_{a\sim\mu(a|s)}\left[Q_\theta(s,a)\right]\right] - \mathbb{E}_{(s,a)\sim\beta}\left[Q_\theta(s,a)\right] \tag{3.10}$$

Esto permite que valores Q altos puedan ser asignados a pares (s,a) que estén dentro de la distribución.

2. **Advantage Weighted Actor Critic (AWAC)** [3]: AWAC es de la familia Actor-Critic basado en TD3. Este algoritmo nace a partir de la necesidad de tener procesos de entrenamiento más eficientes en ambientes complejos.

   Para esto el agente se entrena preliminarmente con un dataset $\mathcal{D}$ Offline, de modo que el modelo pueda posteriormente ser desplegado y ser sometido a $fine-tuning$, pero esta vez interactuando con el ambiente de una manera Online.

   La política óptima se aprende a partir de la optimización de la siguiente función objetivo:

$$J(\phi) = \mathbb{E}_{s_t,a_t\sim\mathcal{D}}\left[\log \pi_\phi(a_t|s_t)\exp\left(\frac{1}{\lambda}A^\pi(s_t,a_t)\right)\right] \tag{3.11}$$

   La ecuación (3.11) representa una estimación del retorno del agente en función de los parámetros de una red neuronal, lo que permite obtener la política $\Pi_\phi(a|s)$ explícitamente a través de la optimización de sus parámetros ϕ mediante gradiente ascendente según $\nabla J_{RL}(\pi_\phi)$.

   En la figura 3.9 se puede ver en detalle su algoritmo.

Dataset $\mathcal{D} = \{(\mathbf{s},\mathbf{a},\mathbf{s}',r)_j\}$
Initialize buffer $\beta = \mathcal{D}$
Initialize π_θ, Q_ϕ
for iteration $i = 1, 2, \ldots$ **do**
  Sample batch $(\mathbf{s},\mathbf{a},\mathbf{s}',r) \sim \beta$
  $y = r(\mathbf{s},\mathbf{a}) + \gamma\mathbb{E}_{\mathbf{s}',\mathbf{a}'}[Q_{\phi_{k-1}}(\mathbf{s}',\mathbf{a}')]$
  $\phi \leftarrow \arg\min_\phi \ \mathbb{E}_\mathcal{D}[(Q_\phi(\mathbf{s},\mathbf{a}) - y)^2]$
  $\theta \leftarrow \arg\max_\theta \ \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\beta}\left[\log \pi_\theta(\mathbf{a}|\mathbf{s})\exp\left(\frac{1}{\lambda}A^{\pi_k}(\mathbf{s},\mathbf{a})\right)\right]$
  if $i >$ num_offline_steps **then**
    $\tau_1, \ldots, \tau_K \sim p_{\pi_\theta}(\tau)$
    $\beta \leftarrow \beta \cup \{\tau_1, \ldots, \tau_K\}$
  end if
end for

Figura 3.9: Algoritmo AWAC [3].

3.2. Sistemas prescriptivos

Según el libro "**Recommender Systems**" [8] existen ciertas metodologías clásicas para el desarrollo de sistemas prescriptivos, tales como "**Collaborative Filtering Models**", "**Knowledge-Based Recommender Systems**" e "**Hybrid and Ensemble-Based Recommender Systems**". Estos conceptos son más fáciles de entender basandose en la idea de que se busca generar recomendaciones de contenido o venta de productos para usuarios de cierta plataforma, lo cual puede ser análogo a recomendaciones de celda de carga a un operador de Molino SAG en el contexto de la presente tesis.

Los **modelos de filtrado colaborativo** consisten en modelos que utilizan el poder colaborativo de las calificaciones proporcionadas por múltiples usuarios para hacer recomendaciones. El principal desafío en el diseño de métodos de filtrado colaborativo es que las matrices de calificaciones podrían contener alta variabilidad o baja representatividad. Existen dos tipos comunes de métodos utilizados en el filtrado colaborativo: los **métodos basados en memoria** los cuales se basan en en criterios de similitud, por ejemplo algoritmos de clustering y los **métodos basados en modelos** los cuales utilizan métodos de aprendizaje automático y minería de datos para crear modelos predictivos. Cabe destacar que el modelo estadístico de la presente tesis podría ser clasificado como un modelo de filtrado colaborativo basado en memoria.

Los **modelos de recomendación basados en conocimiento** son particularmente útiles en el contexto de elementos o eventos que no ocurren con alta frecuencia, dado que estos elementos ocurren raramente y de forma muy especifica, estos se basan en información explícita sobre los requerimientos y preferencias del usuario, en lugar de depender únicamente de datos históricos o calificaciones. Este tipo de modelo podría ser análogo a una especie de sistema experto en el caso en el que número de opciones sea limitado y abordable. En el contexto del problema que se busca resolver esto no es factible, esto debido a la alta variabilidad del contexto operacional de un molino SAG.

Finalmente, los **sistemas híbridos basados en ensamblaje de modelos** pretende utilizar la flexibilidad de diferentes tipos de sistemas de recomendación para un objetivo común cuando existen diversos tipos de fuentes de información. En tales casos, existen muchas oportunidades para la hibridación, donde se combinan los aspectos de diferentes tipos de sistemas para aprovechar las ventajas de cada sistema. Por ejemplo, un sistema de recomendación híbrido podría buscar combinar el poder de múltiples algoritmos de aprendizaje automático para crear un modelo más robusto.

Otros métodos más avanzados para sistemas recomendadores son el **Aprendizaje Reforzado**. En el artículo "**Reinforcement Learning based Recommender Systems: A Survey**" [9] se muestra el diseño de un sistema recomendador basado en RL en el contexto de páginas web que interactúan con usuarios para recomendar artículos, por ejemplo, YouTube, páginas de retail, Netflix etc. Además se explica como el problema de recomendación se consideraba un problema de clasificación o predicción (clasificación o índice de preferencia), pero ahora se acepta ampliamente para ser formulado como un problema de decisión secuencial que se refleja mejor como una interacción usuario-sistema (análogamente Agente-Ambiente). Por lo tanto, el problema puede ser formulado como un **Proceso de Decisión de Markov**

(MDP) y ser resuelto mediante **algoritmos de aprendizaje reforzado**.

A diferencia de los métodos tradicionales de recomendación, RL es capaz de manejar la interacción secuencial y dinámica entre el usuario y el sistema, teniendo en cuenta el compromiso a largo plazo del usuario. Aunque la idea de utilizar RL para la recomendación no es nueva y ha estado presente durante aproximadamente dos décadas, no era práctica, principalmente debido a problemas de escalabilidad de los algoritmos tradicionales de RL. Sin embargo, ha surgido una nueva tendencia en el campo desde la introducción del aprendizaje profundo por refuerzo (DRL), lo que ha hecho posible aplicar RL al problema de la recomendación con grandes espacios de estados y acciones.

Como se ha mencionado en secciones anteriores, la posibilidad de experimentación en línea en un molino SAG queda descartada debido a las graves consecuencias que podría tener tomar una mala decisión en el proceso de aprendizaje de mejorar la toma de decisiones. En el artículo "**Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems**" [2] se expone como es posible que algoritmos de aprendizaje por refuerzo utilicen datos históricos previamente recopilados, sin necesidad de recopilación adicional de datos en línea mediante la interacción con el ambiente. Se menciona que los algoritmos de aprendizaje por refuerzo fuera de línea ofrecen un gran potencial para convertir grandes conjuntos de datos en motores de toma de decisiones poderosos.

Las aplicaciones mostradas en el artículo son escenarios en los cuales no es posible la experimentación, por ejemplo en el área de la salud, en el proceso de diagnóstico y tratamiento de un paciente, donde las acciones corresponden a varias intervenciones disponibles (por ejemplo, pruebas diagnósticas y tratamientos), las observaciones corresponden a los síntomas del paciente y los resultados de las pruebas diagnósticas. Otro ejemplo lo es la manipulación robótica, la cual se utiliza para transmitir el conocimiento de movimientos básicos a otro tipo de aplicaciones como lo son robots de cocina.

Por otro lado, los desafíos relacionados al Aprendizaje Reforzado Fuera de Línea son debido a que los datos de entrenamiento son fijos y no se generan a partir de una interacción con el ambiente, por lo que no hay oportunidad de mejorar mediante exploración y además si existe un cambio en la distribución de los datos, el agente que fue entrenado de manera Offline no será de capaz de captar este cambio en la distribución, tomando acciones a partir de solo los datos históricos con los que fue entrenado.

El artículo "**AWAC: Accelerating Online Reinforcement Learning with Offline Datasets**" [3] intenta abordar el desafió asociado al Aprendizaje Reforzado Fuera de Línea, presentando el algoritmo denominado "**Advantage Weighted Actor Critic**" (AWAC), el cual ofrece un marco simple y efectivo capaz de aprovechar grandes cantidades de datos históricos. Posteriormente se busca realizar un ajuste en línea de sus políticas aprendidas en la modalidad fuera de línea. El estudio demuestra que AWAC permite un aprendizaje rápido de habilidades al combinar datos de demostración previos con experiencia en línea. La validación se lleva a cabo en diversos entornos robóticos simulados y del mundo real, como la manipulación de objetos con una mano robótica multifuncional, la apertura de cajones con un brazo robótico y la rotación de una válvula. Los resultados indican que la incorporación de datos previos puede reducir significativamente el tiempo necesario para aprender diversas

habilidades robóticas en escalas de tiempo prácticas.

Un ejemplo de Utilización del Aprendizaje Reforzado Fuera de Línea en la industria lo es "**DeepThermal: Combustion Optimization for Thermal Power Generating Units Using Offline Reinforcement Learning**" [10]. En el artículo se explica el sistema de IA DeepThermal que utiliza Offline Reinforcement Learning basado en modelos MORE (Model-based Offline RL with Restrictive Exploration) para optimizar la estrategia de control de la combustión de las unidades de generación de energía térmica (TPGU). Se comenta además que DeepThermal ya se ha implantado en cuatro grandes centrales térmicas de carbón en China y ha demostrado una mayor eficiencia de la combustión.

En cuanto al MDP que modela el problema, la función de recompensa de DeepThermal se modela como una combinación ponderada de la eficiencia de la combustión y la reducción de las emisiones, con restricciones de seguridad modeladas como costes. Los estados incluyen las propiedades químicas del carbón y los datos pertinentes de los sensores, como la temperatura, la presión, el viento, el volumen de agua y otras lecturas en diferentes etapas del proceso de combustión. Todas las acciones son continuas, estas son variables de control clave que afectan al proceso de combustión en una TPGU, como el ajuste de válvulas y deflectores.

En cuanto a la aplicación del Aprendizaje de Máquinas en Molinos SAG en el artículo "**Neural networks and support vector machine models applied to energy consumption optimization in semiautogeneous grinding**" [11] se desarrolla un modelo predictivo del nivel de llenado de un molino SAG en función de sus presiones para luego usar esta estimación en la optimización del consumo energético. Por otro lado, en el articulo "**Real-time optimization of sag mills using genetic algorithms**" [12] se busca la optimización del TPH en molinos SAG mediante el uso de algoritmos genéticos junto con ecuaciones físicas que describen el fenómeno en función de las restricciones operativas del proceso, principalmente la velocidad máxima de alimentación de mineral, el consumo de energía y el nivel de llenado de este.

Finalmente, debido a la naturaleza del problema que se busca resolver en la presente tesis, luego del entrenamiento del agente no se podrá evaluar empíricamente el desempeño de este, producto de que no se posee ningún tipo de modelo digital del molino SAG. Por lo que solo queda evaluar el modelo aprovechando la utilización de los datos históricos que se poseen. En ese sentido, según el libro anteriormente mencionado "**Recommender Systems**" [8], existen 3 tipos principales de evaluaciones para sistemas de recomendación, que corresponden a estudios de usuarios, evaluaciones en línea y evaluaciones fuera de línea con conjuntos de datos históricos. Siendo este último de interés para el tema. El diseño de la evaluación de sistemas de recomendación es muy similar al de los sistemas de evaluación de clasificadores debido a la similitud entre los problemas de recomendación y clasificación, por esto los datos se dividen en al menos los conjuntos típicos de entrenamiento, validación y test con el objetivo de prevenir un sobre-ajuste de los modelos en el conjunto de datos de entrenamiento.

Capítulo 4

Metodología

La implementación de esta metodología consiste en el entrenamiento de un agente de toma de decisiones mediante Aprendizaje Reforzado Profundo Fuera de Línea, específicamente a través del algoritmo Advantage Weighted Actor Critic (AWAC) [3] utilizando el lenguaje de programación *Python* y su framework de Aprendizaje Reforzado *D3rlpy* [13].

Como se a mencionado en secciones anteriores, este agente busca dar una recomendación del setpoint HH celda de carga. Esta recomendación posteriormente será contrastada con la calidad de las recomendaciones de los modelos ya existentes, es decir, el modelo estadístico y el modelo estadístico mejorado. Para esto se debe desarrollar una métrica de evaluación que este alineada con lo que se busca estabilizar, en este caso las caídas de TPH del molino SAG.

4.1. Generación de datos

Las fuentes de información contemplan los años 2020 y 2021, provenientes de dos sistemas principales, la primera fuente es **PI System** que contiene toda la información asociada a la sensorización de la planta (TPH, celda de carga, porcentaje de sólidos, rpm, granulometría etc). La segunda fuente es el cruce de **Dispatch** con el **modelo de bloques de la mina**. **Dispatch** contiene toda la información asociada a la posición GPS de las palas y camiones dentro de la mina, por otro lado, el modelo de bloques de la mina permite asociar a cada punto espacial de la mina sus propiedades mineralógicas (por ejemplo dureza) obtenidas a través de muestras de laboratorio. De este modo es posible asociar que tipo de mineral esta entrando al molino SAG.

Una vez generado el dataset consolidado, este es pre-procesado para remover escenarios anómalos y computar valores faltantes. Posteriomente, el dataset se divide en los conjuntos de entrenamiento y test, considerando el periodo 2020-04 y 2021-09 como conjunto test y el resto como conjunto de entrenamiento.

Finalmente, el dataset será procesado para definir los MDP en el formato que espera recibir el framework de Aprendizaje Reforzado *D3rlpy*.

4.2. Desarrollo de métrica

Para el desarrollo de la métrica de evaluacion de los modelos se deberán considerar 3 factores clave:

1. Se excluyen las caídas de TPH originadas por el sistema de control, es decir, escenarios en los cuales la señal de celda de carga esta demasiado cercana al setpoint HH celda de carga.

2. Se consideran los escenarios de pérdida de TPH cuando la señal de celda de carga supera la recomendación del setpoint HH celda de carga (TPH ganado al haber evitado dicha pérdida [+]).

3. Se consideran los escenarios pérdida de TPH cuando la señal de celda de carga no supera el setpoint HH celda de carga (TPH perdido por dar una recomendación incorrecta [-]).

4.3. Modelamiento

El agente recomendará el setpoint HH celda de carga que se deberá configurar en el sistema de control en los siguientes 5 minutos dado el estado actual del molino SAG.

De este modo los estados, acciones y recompensa estarán definidos por:

1. **Estados**:
   a) **Estados discretos**: Edad del molino.
   b) **Estados continuos**: Agua, porcentaje de sólidos, rpm, setpoint TPH, granulometría, celda de carga y setpoint HH celda de carga.

2. **Acciones**: Setpoint HH celda de carga en minuto 5, variable continua.

3. **Recompensa**: 1 si las pérdidas de TPH son menores a 100 y -1 si las pérdidas de TPH son mayores a 100. Esto último acorde a criterio experto.

En cuanto a la construcción de los episodios de entrenamiento, estos se construyen a partir del inicio de un evento de pérdida de TPH. Una vez identificado el inicio, se consideran los 5 minutos anteriores de la serie de tiempo, rango de tiempo donde se pudo haber generado la causa de la caída de TPH debido a altos niveles de celda de carga. Finalmente, el termino del episodio se considera cuando el evento de pérdida de TPH finaliza (es un evento continuo que tiene cierto tiempo de duración).

Por otro lado, para evidenciar el correcto entrenamiento del agente se mostraran las curvas de *Loss* para el *Actor* y el *Critic*. En la figura 4.1 se muestra un esquema del MDP que se desarrollara en el tiempo.

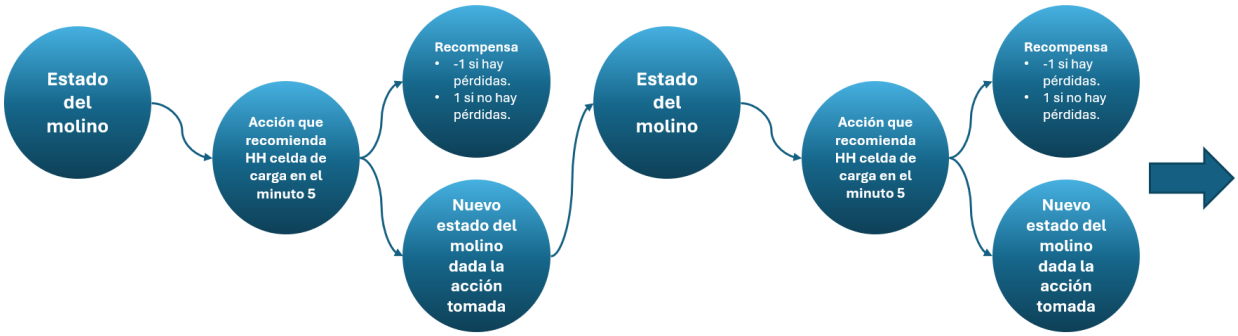

Figura 4.1: Diagrama MDP.

4.4. Evaluación y análisis de resultados

Una vez entrenado el agente, se evaluará la factibilidad de sus recomendaciones mediante:

1. **Comparación de la distribución de las recomendaciones generadas**: Análisis de las diferencias de la distribución de las recomendaciones del modelo RL vs el setpoint HH celda de carga del conjunto de entrenamiento y test.

2. **Análisis de series de tiempo**: Se analizarán las series de tiempo de las recomendaciones generadas para corroborar que puedan ser seguidas a lo largo del tiempo.

3. **Evaluación y comparación de modelos**: Se procederá a comparar el modelo RL, modelo estadístico y el modelo estadístico mejorado mediante la comparación de distribuciones, análisis de series de tiempo y la evaluación de desempeño de las recomendaciones a través de la métrica desarrollada.

.

Capítulo 5

Desarrollo

5.1. Métrica de desempeño

Como se menciono en el marco teórico, las caídas de TPH que serán evaluadas dentro de la métrica serán las asociadas a **embancamiento**. La métrica contiene dos componentes fundamentales, una que castiga malas recomendaciones y otra que premia buenas recomendaciones cuando existen pérdidas de TPH:

i) **Mala recomendación (pérdida de TPH)**: Dada una caída de TPH, si la recomendación de celda de carga es mayor al valor real de celda de carga, significa que la recomendación esta indicando que se podría cargar con mayor peso el SAG, sin embargo, ya con un peso menor a este ya se origina una pérdida. Debido a esto, la perdida asociada se imputa de manera negativa en la métrica.

ii) **Buena recomendación (ganancia de TPH)**: Dada una caída de TPH, si la recomendación de celda de carga es menor al valor real de celda de carga, significa que la recomendación esta indicando que no se debe cargar con más peso el molino SAG, ya que con un peso mayor al indicado se originará una pérdida de TPH. Debido a esto, la pérdida asociada se imputa de manera positiva en la métrica.

A continuación se enumeran los pasos que se siguieron para desarrollar la métrica de evaluación:

1. **Cálculo de pérdidas de TPH**: Se calculan las diferencias del setpoint TPH con el TPH, las diferencias positivas mayores a 100 se consideran como caídas efectivas de TPH respecto de su setpoint (definido por Felipe Contreras).

2. **Cálculo de inicio de eventos de pérdidas de TPH**: Dado que las caídas de TPH son eventos continuos en el tiempo, se debe obtener el inicio de la pérdida de TPH.

3. **Identificación del contexto del evento de pérdida de TPH**: Para cada inicio de pérdida de TPH, se identifica si la señal de celda de carga esta cerca (menor a 1) o lejos (mayor a 1) del setpoint HH celda de carga histórico (definido por Carlos Orellana). Según esto se define si el evento de pérdida de TPH es por embancamiento o por actuación del sistema de control. Finalmente, se consideran solo los eventos por embancamiento.

4. **Identificación del contexto de la recomendación de celda de carga e imputación de la evaluación**: Para cada inicio de pérdida de TPH, se identifica si la señal de celda de carga esta por sobre o debajo la recomendación de celda de carga. Según esto, se imputa de manera negativa o positiva el TPH asociado al evento, de acuerdo a las definiciones de la métrica que fueron mencionadas. Finalmente, mientras mayor sea el valor de la métrica del modelo, este será mejor evaluado.

5.2. Modelo prescriptivo

5.2.1. Limpieza y estructura de datos

Luego de consolidar los datos (PI System y Dispatch), se obtiene la siguiente muestra de la tabla consolidada de datos por minuto:

Tabla 5.1: Consolidado de datos para entrenamiento del agente.

Tiempo	granulometría	celda de carga	rpm	porcentaje de sólidos	agua	TPH	HH TPH	LL celda de carga	HH celda de carga	dureza	Edad
2020-03-06 21:04:00	50.9	737.8	9.1	68.8	1882.4	4172.0	4220.0	750.0	830.0	86.0	2.0
2020-03-06 21:05:00	50.3	740.3	9.1	69.0	1882.0	4196.1	4220.0	750.0	830.0	86.0	2.0
2020-03-06 21:06:00	49.8	746.8	9.0	68.9	1880.2	4168.4	4220.0	750.0	830.0	86.0	2.0
2020-03-06 21:07:00	49.2	750.2	8.9	69.0	1884.9	4202.0	4220.0	750.0	830.0	86.0	2.0
2020-03-06 21:08:00	48.6	754.6	8.9	68.9	1879.7	4182.3	4220.0	750.0	830.0	86.0	2.0

5.2.1.1. Limpieza de datos

Una vez consolidados los datos se aplica la siguiente lógica de limpieza de datos:

- **Remoción de outliers I**:
  1. **Granulometría**: Variable entre 20 y 100. Adicionalmente se suaviza la señal con una media móvil centrada de 7 minutos
  2. **Celda de carga**: Variable menor a 1000.
  3. **TPH**: Variable entre 2000 y 4800.
  4. **HH TPH**: Variable mayor a 3000.
- **Remoción de outliers II mediante el criterio del rango intercuartil**: Para cada variable numérica se calculan sus cuartiles, para luego aplicar el criterio mencionado.
- **Interpolación**: Cada variable es interpolada linealmente con un limite máximo de 5 minutos.

5.2.1.2. Generación de planilla de eventos de interés y filtrado

Con los datos ya consolidados y procesados, se procede a estructurar una tabla de datos auxiliar que permite capturar los eventos de interés para entrenar el agente:

Tabla 5.2: Tabla de clasificación de eventos de pérdida de TPH.

inicio pérdida	fin pérdida	inicio evento	fin evento	impacto TPH	variación HH TPH	std HH TPH	Causalidad sistema de control	Causalidad molino vació	Tipo de evento	Flag	duración evento
2021-03-23 04:39:00	2021-03-23 09:49:00	2021-03-23 04:29:00	2021-03-23 09:59:00	154.8	False	0.0	False	False	Embancamiento	E	310.0
2021-03-23 10:28:00	2021-03-23 10:35:00	2021-03-23 10:18:00	2021-03-23 10:45:00	200	True	25.4	False	True	Molino vacio	M	7.0
2021-03-23 10:41:00	2021-03-23 11:37:00	2021-03-23 10:31:00	2021-03-23 11:47:00	500	False	0.0	False	True	Molino vacio	M	56.0
2021-03-23 11:54:00	2021-03-23 16:18:00	2021-03-23 11:44:00	2021-03-23 16:28:00	105	True	206.1	False	True	Molino vacio	M	264.0
2021-03-23 18:00:00	2021-03-24 00:31:00	2021-03-23 17:50:00	2021-03-24 00:41:00	255.5	True	77.5	True	False	Sistema de control	S	391.0
2021-03-24 01:07:00	2021-03-24 01:45:00	2021-03-24 00:57:00	2021-03-24 01:55:00	105	False	0.0	True	False	Sistema de control	S	38.0

Esta tabla permite filtrar eventos, de modo tal que las caídas de TPH no se deban a la actuación del sistema control o a otras causas, como por ejemplo cambios en el setpoint HH

TPH, que numéricamente son pérdidas en el transiente de llegar de un punto operativo a otro.

5.2.1.3. Estandarización, estructura de datos como MDP y división en entrenamiento/test

Luego de recopilar los eventos de pérdidas de TPH asociados a embancamiento, estos son unidos al conjunto de datos que no contiene pérdidas de TPH. Esto con el objetivo de entrenar al modelo con situaciones en las cuales se tomaron buenas y malas decisiones. Finalmente, los datos son estandarizados para el correcto entrenamiento del agente.

5.2.2. Resultados de entrenamiento del agente

Para el entrenamiento y test del agente AWAC se utilizaron 2 datasets, de 80.701 y de 69.685 registros respectivamente. En cuanto a la cantidad de episodios de entrenamiento, estos fueron en total de 1.493.

Para los parámetros de entrenamiento, se consideraron 10 épocas de entrenamiento, con un tamaño de batch de 1024 muestras. En cuanto a los hiperparámetros que el modelo utilizo:

- **Tasa de aprendizaje Actor**: 0.0003
- **Tasa de aprendizaje Critic**: 0.0003
- **Gamma**: 0.99
- **Intervalo de actualización Actor**: 1

Finalmente, en cuanto a los resultados del entrenamiento, en las figuras 5.1 y 5.2, se puede evidenciar como la función de Loss del Actor y el Critic disminuyen luego de las 10 épocas de entrenamiento.

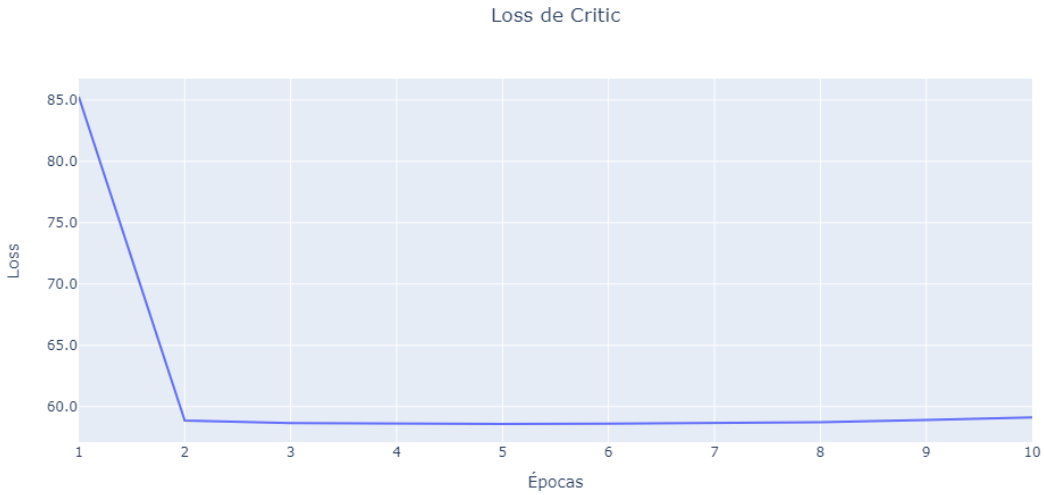

Figura 5.1: Loss de Critic.

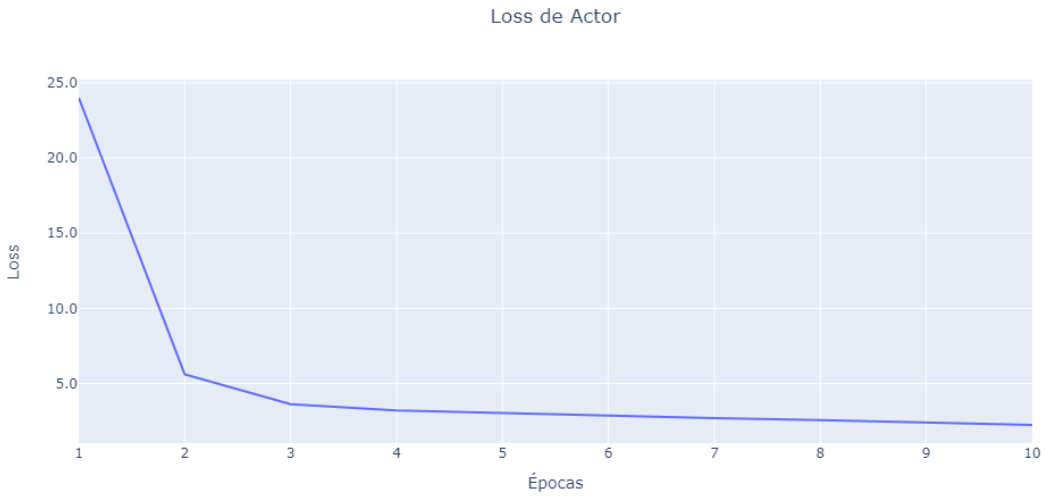

Figura 5.2: Loss de Actor.

Luego del proceso de entrenamiento, se procede a utilizar el agente para generar recomendaciones. Como análisis preliminar se generan las recomendaciones tanto para el conjunto de entrenamiento como de test.

Al comparar las recomendaciones generadas del agente a través del conjunto de entrenamiento con las acciones históricas del mismo conjunto, se puede observar a partir de la figura 5.3 y la tabla 5.3, que existe un desplazamiento hacia la derecha en la distribución de recomendaciones del modelo RL respecto de las acciones efectivamente tomadas, esto se puede asociar a que los datos son de entrenamiento, debido a esto, el modelo puede estar sobre-ajustado a estos datos, lo que puede generar un sesgo hacia acciones con valores más altos que maximizan su recompensa. Por otro lado, se puede observar que los valores de las recomendaciones están dentro de los rangos esperados y que la variabilidad de la toma de decisiones disminuye.

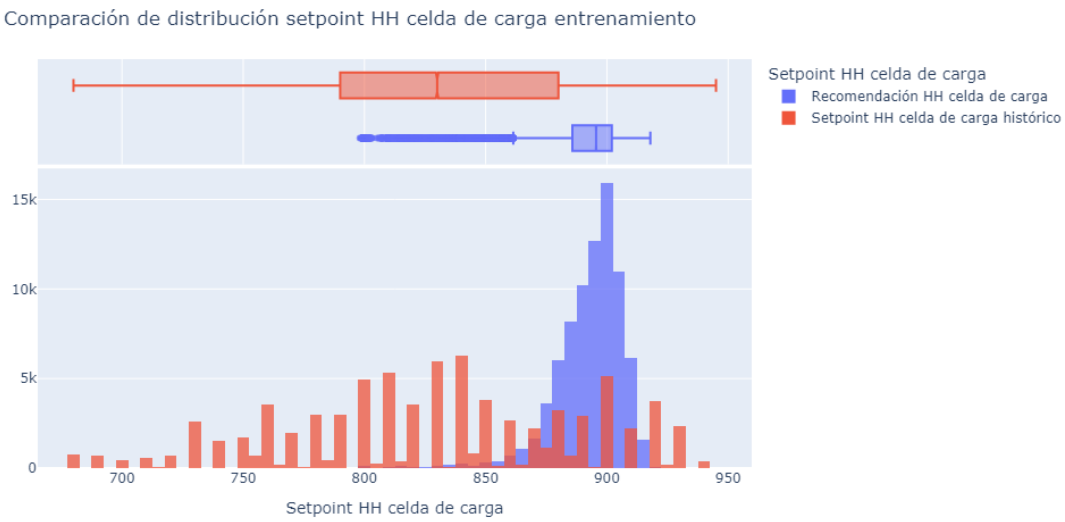

Figura 5.3: Comparación de distribuciones de política y acciones de entrenamiento.

Tabla 5.3: Comparación de promedio y desviación estándar en conjunto de entrenamiento.

	Promedio	**Desviación estándar**
HH celda de carga	829	60
Recomendación HH celda de carga	893	14

Por otro lado, al comparar las recomendaciones generadas del agente a través del conjunto de test, se puede observar a partir de la figura 5.4 y la tabla 5.4, que la distribución de acciones recomendadas respecto a las efectivamente tomadas se encuentran centradas en torno a promedios similares y que la variabilidad de la toma de decisiones al igual que en las recomendaciones de entrenamiento disminuyen. A partir del análisis, se puede concluir de que el agente esta aprendiendo políticas que son consistentes con la toma de decisiones desde un punto de vista estadístico en términos del promedio, desviación estándar y del rango de valores que toma respecto de la distribución de acciones históricas tomadas.

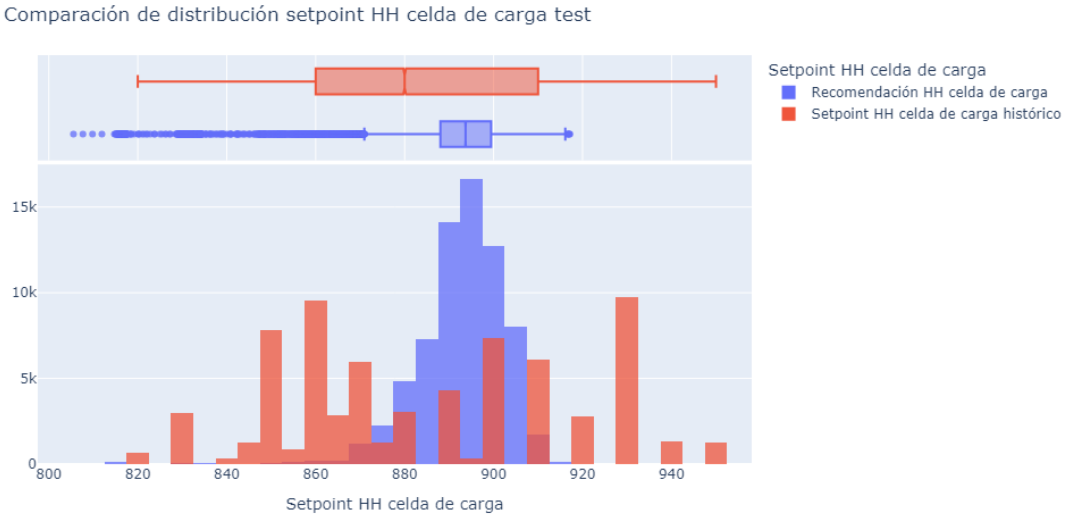

Figura 5.4: Comparación de distribuciones de política y acciones de test.

Tabla 5.4: Comparación de promedio y desviación estándar en conjunto de test.

	Promedio	**Desviación estándar**
HH celda de carga	885	32
Recomendación HH celda de carga	893	10

5.3. Evaluación y análisis de resultados

5.3.1. Análisis de políticas óptimas como series de tiempo

Otro punto relevante a analizar para la política aprendida, es la validación de que estas recomendaciones sean factibles de seguir a lo largo del tiempo, debido a esto, a continuación se muestran 3 escenarios en donde se evidencia el comportamiento de la recomendación de celda de carga como serie de tiempo:

- **Escenario de pérdida de TPH con buena recomendación**: En la figura 5.5, se observa un escenario en donde se genera una caída de TPH entre las 10:53 y las 10:59, la cual se asocia a que el molino fue cargado con una cantidad de mineral mayor al que podía procesar.

  Por otro lado, la recomendación de celda de carga del agente indica que se debe restringir el nivel de llenado, evidenciándose una buena recomendación de celda de carga, ya que, minutos antes cuando la recomendación del agente es superada se genera la pérdida de TPH.

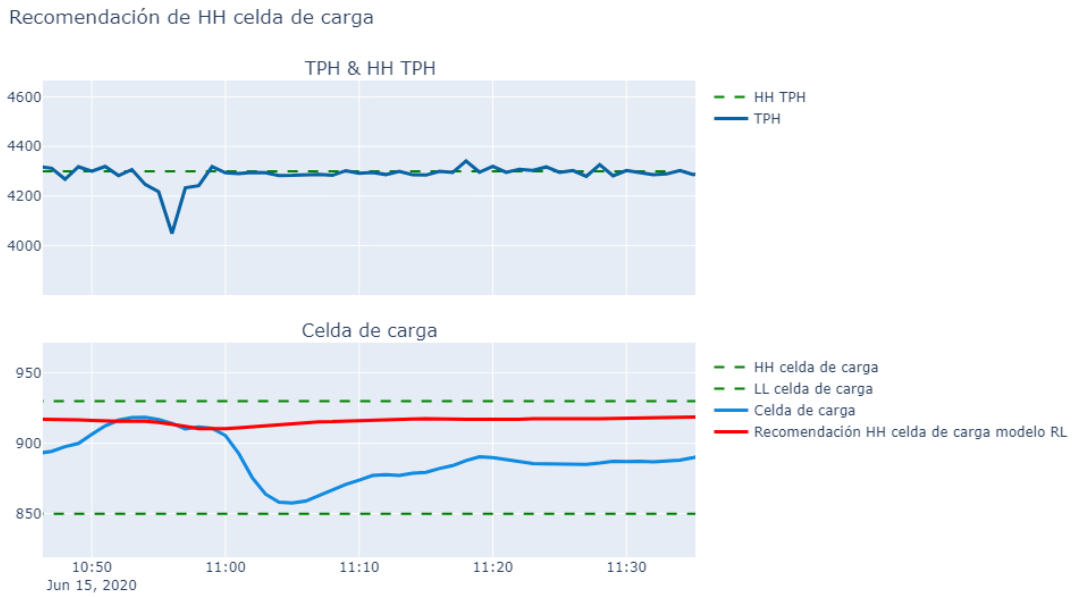

Figura 5.5: Evento de pérdida de TPH con buena recomendación.

- **Escenario de pérdida de TPH con mala recomendación**: En la figura 5.6, se observa un escenario en donde se genera una caída de TPH, la cual tiene su máximo a las 16:29. En este caso se observa que la recomendación del agente indica que el molino puede seguir siendo cargado con una cantidad mayor de mineral, sin embargo, pese a que el nivel de carga real del molino es mucho menor a la recomendación se genera una pérdida de TPH. En este caso, la recomendación del agente no es correcta.

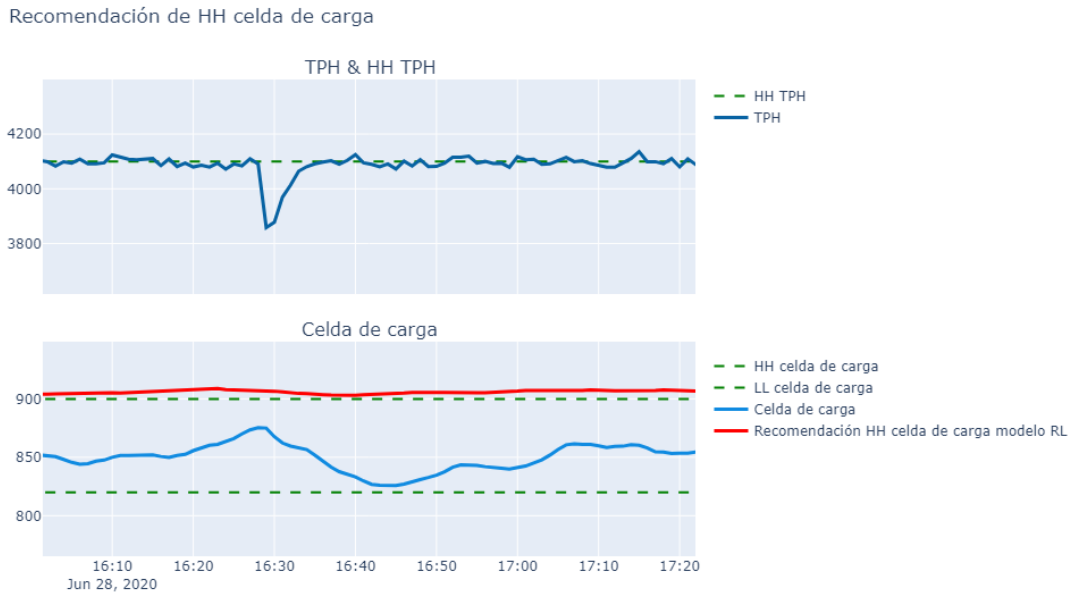

Figura 5.6: Evento de pérdida de TPH con mala recomendación.

- **Escenario sin pérdida de TPH**: En la figura 5.7, se observa un escenario en donde no hay pérdidas de TPH. En este caso, se pretende visualizar como se comporta la recomendación cuando no hay pérdidas de TPH. como se puede apreciar, la recomendación posee una baja variabilidad a lo largo del tiempo, por lo que tiene un comportamiento esperado en contextos sin pérdida de TPH.

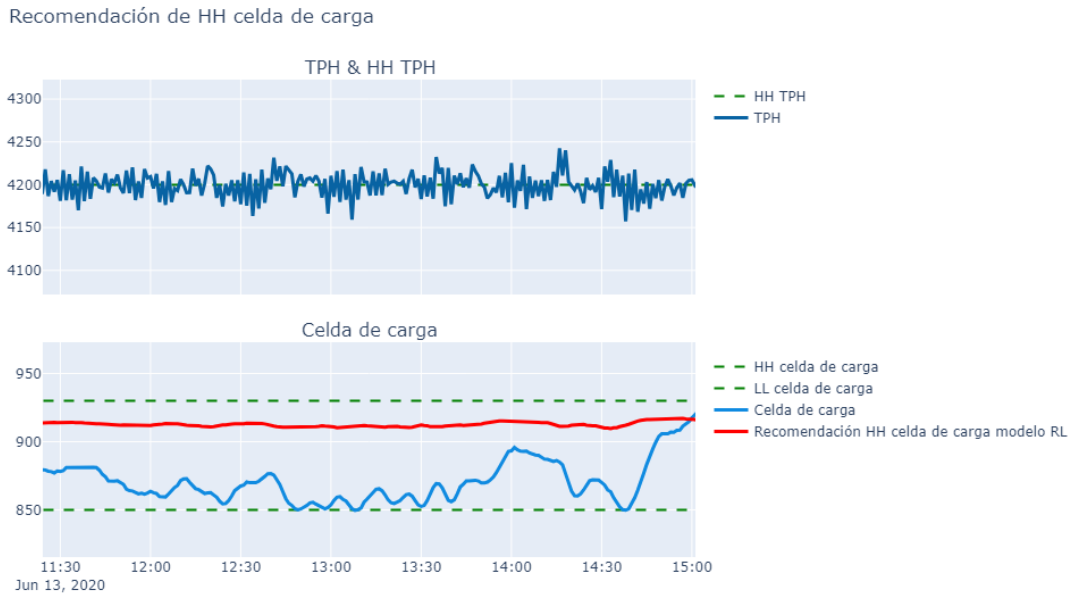

Figura 5.7: Escenario sin pérdida de TPH.

5.3.2. Comparación de modelos mediante distribución de recomendaciones

En esta subsección se comparan los modelos a través de la visualización de un boxplot y una tabla de valores resumen de sus distribuciones. A partir de la figura 5.8 y la tabla 5.5, se puede apreciar que los valores de HH celda de carga históricos (Boxplot morado) tienden a tomar valores discretos a comparación de los modelos que tienen una distribución más uniforme de valores. Por otro lado, se observa que la distribución de recomendaciones del modelo de RL tiende a estar entre medio de la distribución del modelo base y el modelo base mejorado. Finalmente, como era de esperarse, el modelo base mejorado es el que tiene mayor tendencia a tomar valores menores debido a la forma en la que se construyo, es decir minimizando pérdidas, a comparación del modelo base que se construye maximizando TPH.

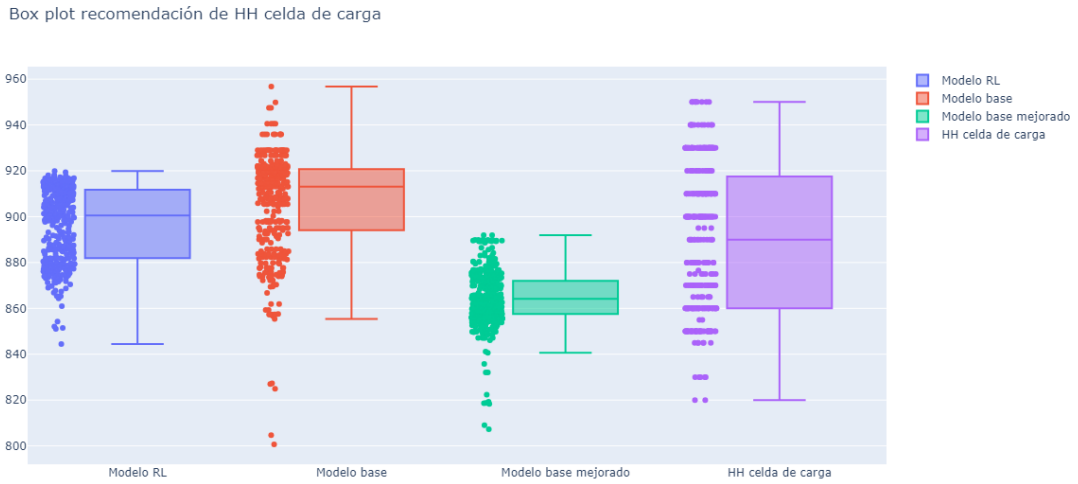

Figura 5.8: Box plot comparación de distribución de recomendaciones modelos HH celda de carga.

Tabla 5.5: Tabla comparativa de estadísticos de distribución de recomendaciones.

	Modelo RL	**Modelo base**	**Modelo base mejorado**	**HH celda de carga**
promedio	896.7	906.2	864.0	889.7
std	16.1	21.8	12.5	31.4
min	844.4	800.6	807.2	820
25 %	881.9	894.1	857.5	860
50 %	900.5	913.0	864.1	890
75 %	911.7	920.7	871.9	915
max	919.9	956.7	891.9	950

5.3.3. Comparación de modelos mediante análisis de series de tiempo

En esta subsección se describen las comparaciones de los modelos a través del análisis de sus series de tiempo.

5.3.3.1. Escenarios con recomendación correcta

En esta subsección se muestran 4 escenarios en los cuales la recomendación del agente fue correcta.

En la figura 5.9 se observa una pérdida de TPH que inicia a las 2:15 y que termina a las 2:29. Minutos antes a la caída, la señal de celda de carga supera la recomendación del modelo RL (curva roja), siendo aun más conservadora la recomendación del modelo base mejorado (curva rosada). Por otro lado, la recomendación del modelo base es más agresiva, recomendando cargar con un nivel muy similar al definido en ese momento, sin embargo, esa recomendación es incorrecta debido a la pérdida que se genera.

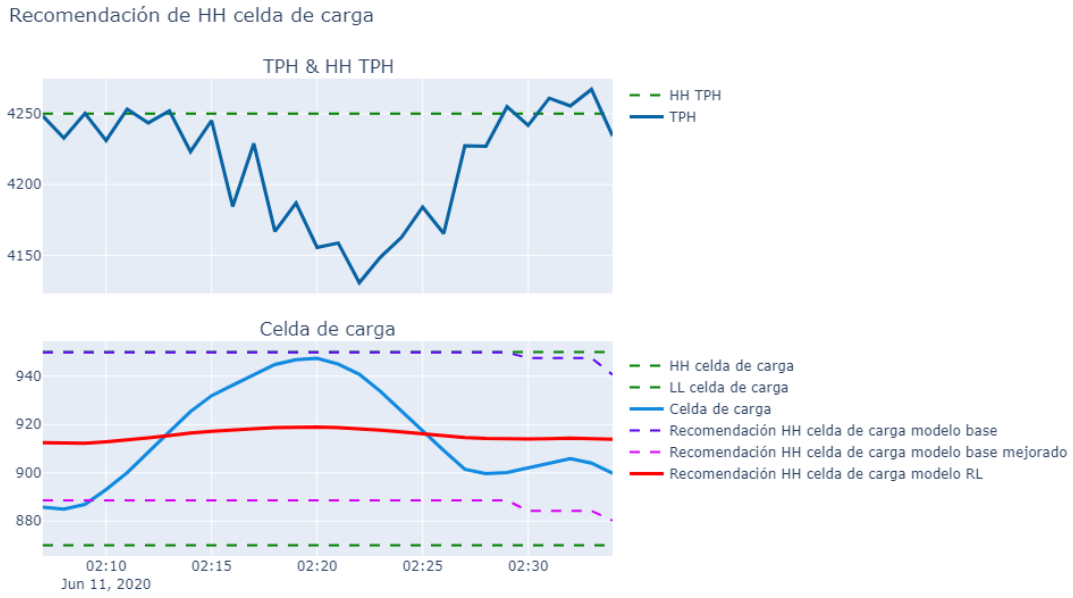

Figura 5.9: Escenario de recomendación correcta I.

En la figura 5.10 se observa una caída de TPH entre las 4:07 y las 4:25. Minutos antes de la caída, se supera la recomendación del modelo RL y la del modelo base (ambos se encuentran recomendando niveles similares de carga). Por otro lado, nuevamente el modelo base mejorado se caracteriza por dar recomendaciones demasiado conservadoras.

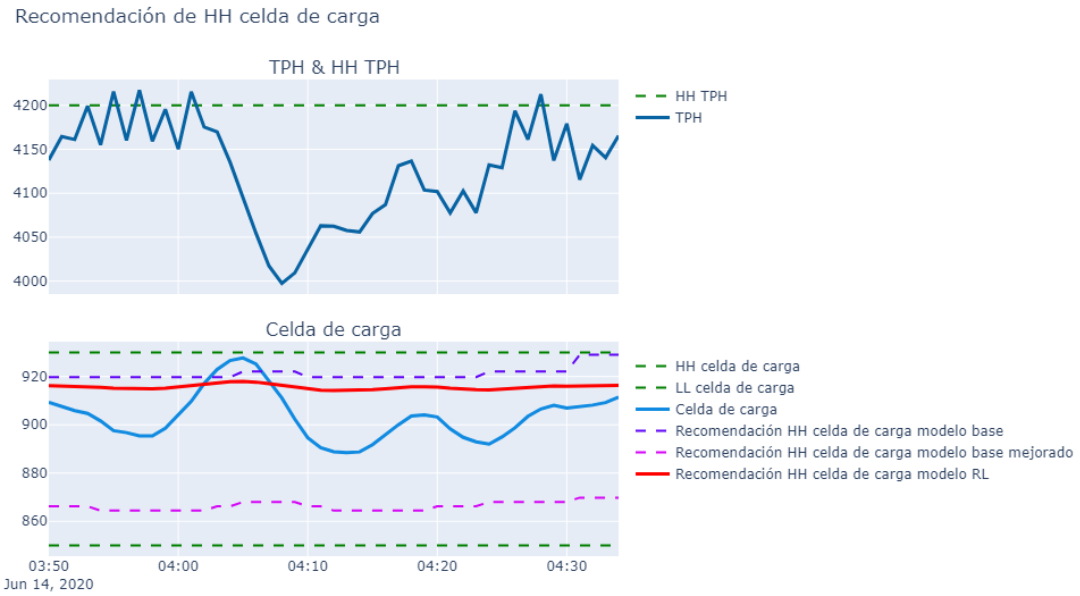

Figura 5.10: Escenario de recomendación correcta II.

En la figura 5.11 se observa una caída de TPH entre las 8:23 y las 8:34. Minutos antes se supera la recomendación del modelo RL, sin embargo, el modelo base da una recomendación incorrecta. Por otro lado, si bien el modelo base mejorado indica cargar con un nivel menor de carga, esta se caracteriza por ser demasiado conservadora respecto a los niveles usados en ese momento.

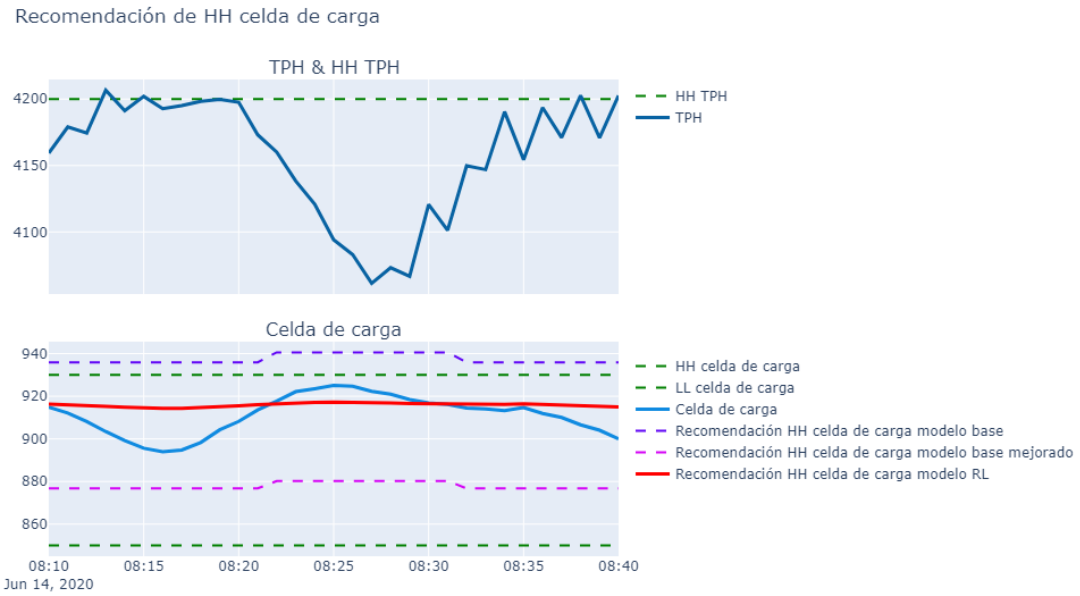

Figura 5.11: Escenario de recomendación correcta III.

En la figura 5.12 se observa una caída de TPH entre las 10:55 y las 10:58. Minutos antes a

la caída de TPH, se supera la recomendación de TPH. La recomendación del modelo base es similar a la utilizada en ese momento, sin embargo, es incorrecta. Por otro lado, nuevamente, la recomendación del modelo base mejorado se caracteriza por ser demasiado conservadora.

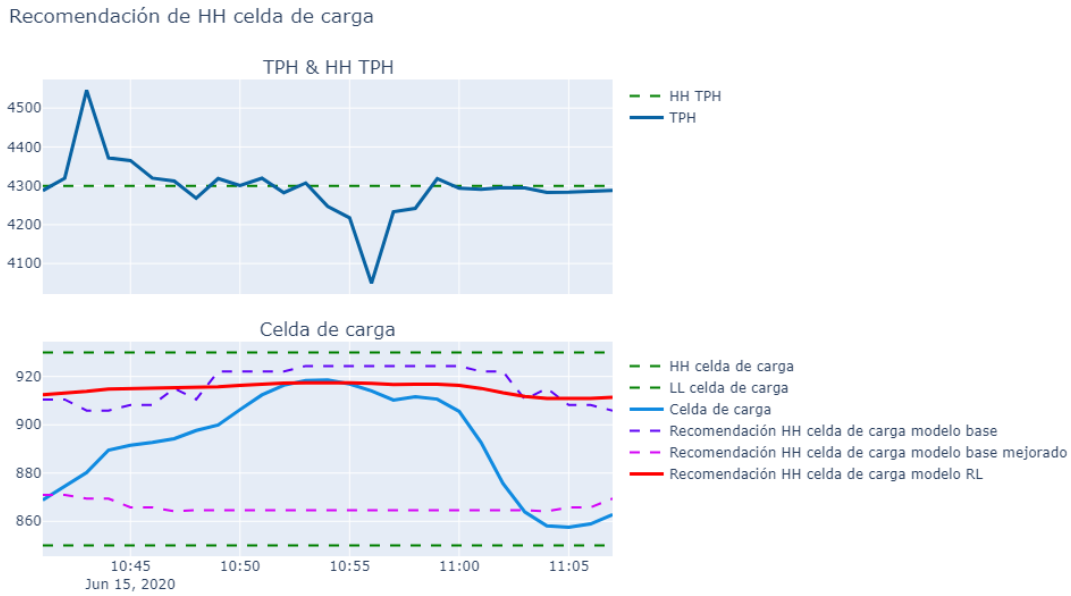

Figura 5.12: Escenario de recomendación correcta IV.

5.3.3.2. Escenarios con recomendación incorrecta

En esta subsección se muestran 4 escenarios en los cuales la recomendación del agente fue incorrecta.

En la figura 5.13 se observa una caída de TPH entre las 12:35 y las 12:37. Minutos antes se observa que el modelo base indica una buena recomendación, ya que, al superar su recomendación se genera la caída de TPH. Por otro lado, la recomendación del modelo RL es incorrecta, ya que, indica cargar con un nivel mayor el molino. Nuevamente, la recomendación del modelo base mejorado es la más conservadora.

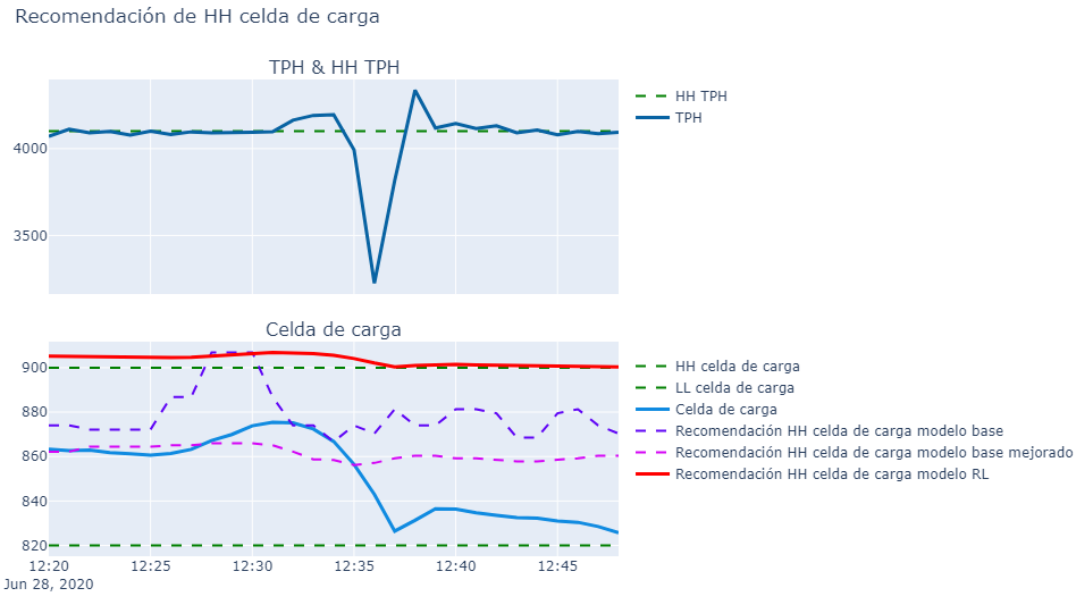

Figura 5.13: Escenario de recomendación incorrecta I.

En la figura 5.14 se observa una caida de TPH entre las 16:28 y las 16:35. Minutos antes se supera la recomendación del modelo base, por lo que se valida su recomendación como correcta. Nuevamente la recomendación del modelo RL es incorrecta al ser más agresiva y la recomendación del modelo base mejorado es la más conservadora.

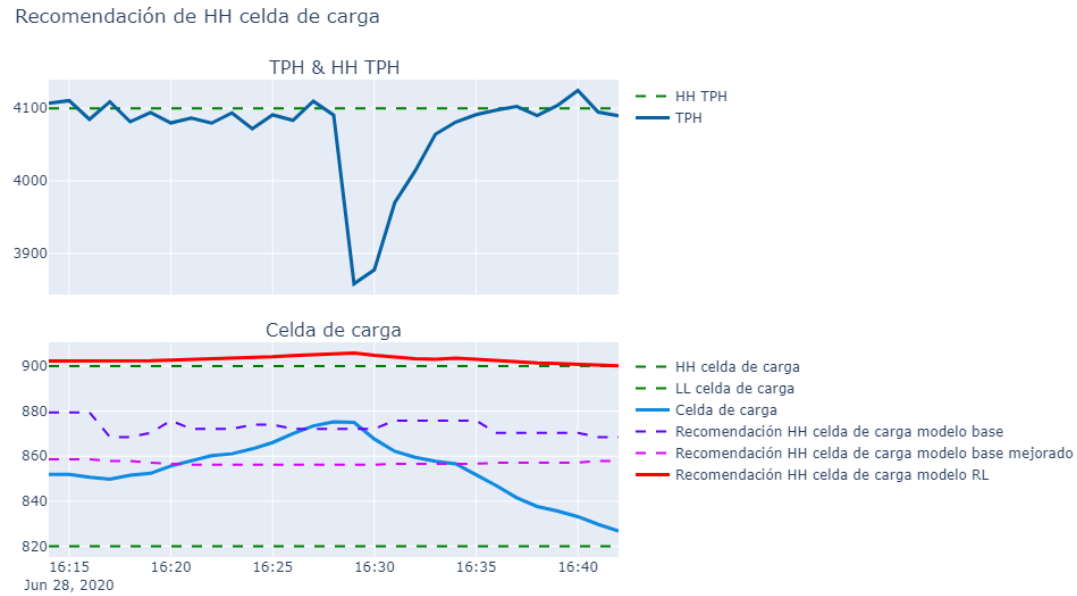

Figura 5.14: Escenario de recomendación incorrecta II.

En la figura 5.15 ser observa una caída de TPH entre las 20:40 y las 21:22. En este caso

ninguna recomendación es correcta, ya que, todas indican cargar con un nivel mayor el molino, sin embargo, con un nivel de carga menor, ya se genera una pérdida de TPH.

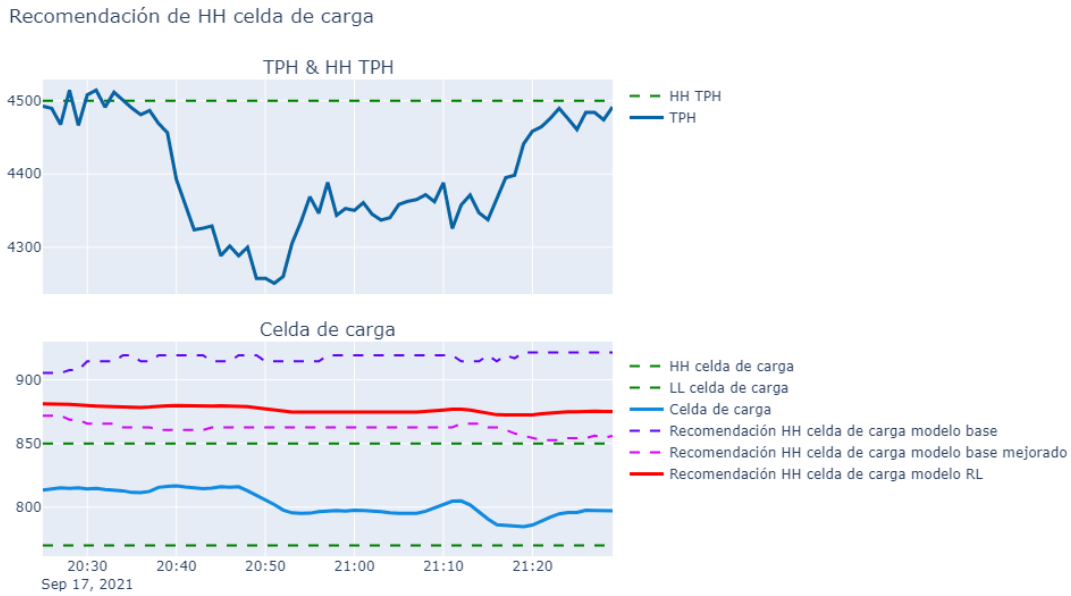

Figura 5.15: Escenario de recomendación incorrecta III.

En la figura 5.16 se observa una caída de TPH entre las 5:08 y las 6:06. Minutos antes a la caída de TPH se observa que se supera la recomendación del modelo mejorado, por lo que en este caso su recomendación es correcta. Por otro lado, las recomendaciones del modelo RL y el modelo base son incorrectas debido a que indican cargar el molino con un nivel mayor de carga.

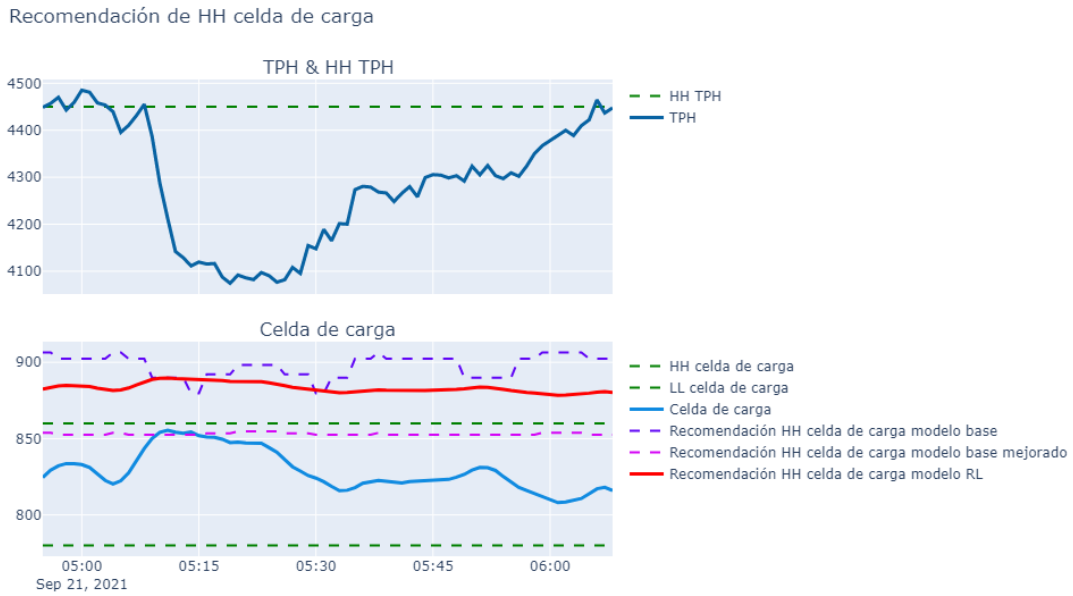

Figura 5.16: Escenario de recomendación incorrecta IV.

5.3.4. Evaluación de modelos mediante métrica desarrollada

Recordando que la métrica desarrollada se compone de dos componentes:

[TPH por recomendaciones correctas] - [TPH por recomendaciones incorrectas]

La evaluación de los modelos entrego los siguientes resultados:

Tabla 5.6: Evaluación de modelos.

	Modelo RL	**Modelo base**	**Modelo base mejorado**
Evaluación	-42020	-45389	29575

A partir de la tabla 5.6 se puede ver que el ranking en orden descendente (mejor a peor modelo) fue:

1. **Modelo base mejorado**
2. **Modelo RL**
3. **Modelo base**

Por otro lado, a partir de la información de la tabla 5.5, si los modelos son ordenados en orden ascendente en términos de la mediana de sus recomendaciones:

1. **Modelo base mejorado**
2. **Modelo RL**
3. **Modelo base**

De esto se puede concluir que en la medida de que las recomendaciones tiendan a ser más bajas, el modelo será mejor evaluado. Esto último tiene sentido, ya que, al dar recomendaciones más bajas, es más probable capturar eventos de pérdida de TPH que se generan al superar la recomendación, lo que permite aumentar el valor de la componente de la métrica [**TPH por recomendaciones correctas**].

Por otro lado, también permite minimizar la componente [**TPH por recomendaciones incorrectas**], ya que es menos probable capturar eventos de caída de TPH en donde la recomendación este por sobre la señal de celda de carga.

Finalmente, se puede concluir que por el diseño de la métrica, se genera de manera natural un sesgo hacia recomendaciones más bajas. Es importante recordar que el diseño de la métrica no considera los eventos en donde las caídas de TPH son debido a la actuación del sistema de control, por lo que no existe una componente que premie eventos en donde la recomendación de celda de carga deba ser más alta.

Capítulo 6

Conclusiones y trabajo futuro

Se logro demostrar que es posible que el agente aprenda políticas de toma de decisiones consistentes con la realidad mediante un algoritmo de RL profundo fuera de línea, alcanzando además resultados prometedores al ser comparado con el modelo estadístico en términos de la evaluación mediante la métrica desarrollada y el análisis de series de tiempo.

Se concluye que los resultados han sido satisfactorios, sin embargo, si bien las políticas aprendidas son consistentes con los resultados esperados a partir del conocimiento previo, no se puede decir de manera categórica que las políticas aprendidas son óptimas o seguras debido a que no es posible la evaluación en el ambiente real o al menos un ambiente simulado del molino SAG.

Como trabajo futuro se propone experimentar con el modelo a escala del Molino SAG que posee el AMTC (modelo Magotteaux). Esto con objetivo de generar datos que permitan el entrenamiento fuera del línea del agente para posteriormente ser evaluado en línea en el modelo Magotteaux y adicionalmente observar el proceso el re-entrenamiento del modelo al interactuar con el ambiente.

Aprendizaje Reforzado Profundo Fuera de Línea es un método muy interesante para aprender políticas óptimas solamente a partir de data histórica, sin embargo, surge la necesidad de poder testear las políticas aprendidas, lo cual no es posible con el estado del arte actual. Por lo mencionado anteriormente, se propone como linea de investigación futura el diseño de metodologías que permitan testear de manera robusta las políticas aprendidas a partir del mismo tipo de estructura de información con la cual se entrena el algoritmo.

Adicionalmente, Como trabajo futuro se propone lograr encontrar la forma de que el agente sea capaz de responder eficientemente frente a escenarios de caídas de TPH tanto por embancamiento como por accionamiento del sistema de control, ya que esto permitiría abordar en completitud las causas de las caídas de TPH.

Por otro lado, también se propone experimentar la utilización de modelos predictivos para lograr desarrollar una especie de ambiente simulado o modelo dinámico del molino SAG el cual puede ser útil para simular escenarios o para ser utilizado directamente como ambiente de entrenamiento en algoritmos de Aprendizaje Reforzado.

Bibliografía

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review," *and Perspectives on Open Problems*, vol. 5, 2020.

[3] A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[4] M. Yahyaei, M. Hilden, F. Shi, L. X. Liu, G. Ballantyne, and S. Palaniandy, "Comminution," *Production, Handling and Characterization of Particulate Materials*, pp. 157–199, 2016.

[5] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, pp. 387–395, Pmlr, 2014.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[7] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.

[8] C. C. Aggarwal *et al.*, *Recommender systems*, vol. 1. Springer, 2016.

[9] M. M. Afsar, T. Crump, and B. Far, "Reinforcement learning based recommender systems: A survey," *ACM Computing Surveys*, vol. 55, no. 7, pp. 1–38, 2022.

[10] X. Zhan, H. Xu, Y. Zhang, X. Zhu, H. Yin, and Y. Zheng, "Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 4680–4688, 2022.

[11] M. Curilem, G. Acuña, F. Cubillos, and E. Vyhmeister, "Neural networks and support vector machine models applied to energy consumption optimization in semiautogeneous grinding," vol. 25, pp. 761–766, 01 2011.

[12] J. Becker, J. L. Salazar Navarrete, L. Magne, and F. Cubillos, "Real-time optimization of sag mills using genetic algorithms," 12 2022.

[13] T. Seno and M. Imai, "d3rlpy: An offline deep reinforcement learning library," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 14205–14224, 2022.