



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA



Departamento de Informática
Universidad Técnica Federico Santa María

(INF395) INTRODUCCIÓN A LAS REDES NEURONALES ARTIFICIALES AND DEEP LEARNING

PROFESOR: ALEJANDRO VELOZ

IMPLEMENTACIÓN DE REDES LSTM Y APRENDIZAJE CONTRASTIVO PARA LA PREDICCIÓN DE URGENCIAS RESPIRATORIAS EN CHILE

ANÁLISIS PARA LA GESTIÓN Y PLANIFICACIÓN
DE RECURSOS

ALAN YANINE ¹

BRYAN HINOSTROZA ²

VALPARAÍSO, 19 DE NOVIEMBRE DE 2025

¹Correo: alan.yaninet@usm.cl - Rut: 202004131-8 - Carrera: Ingeniería civil Informática

²Correo: bhinostroza@usm.cl - Rut: 90.006.937-5 - Carrera: Big Data y Ciencia de Datos

1 Enlaces al Proyecto

Para facilitar la revisión de la implementación y la presentación de este trabajo, se proporcionan los siguientes recursos:

- **Repositorio de Código:** [Enlace a GitHub]
- **Presentación Grabada:** [Enlace al Video]

2 Introducción y Contexto

2.1 Descripción del Cliente y Problemática

El Ministerio de Salud de Chile (MINSAL) enfrenta estacionalmente crisis sanitarias derivadas del aumento de virus respiratorios (VRS, Influenza, COVID-19). En 2023 y 2024, la ocupación de camas críticas superó el 90% en diversas regiones, evidenciando la necesidad de herramientas de gestión proactiva [1].

El problema central radica en la diversidad de patrones locales: las dinámicas de contagio no son uniformes en todo el país. Comunas geográficamente distantes pueden exhibir comportamientos epidemiológicos similares debido a factores latentes (densidad poblacional, clima costero/valle, conectividad), lo que dificulta la planificación basada puramente en la vecindad geográfica.

2.2 Objetivo del Proyecto

Este trabajo propone un sistema predictivo híbrido que combina:

1. **Aprendizaje Supervisado (LSTM):** Para capturar dependencias temporales no lineales a largo plazo en las series de tiempo.
2. **Aprendizaje Contrastivo (Triplet Loss):** Para aprender una representación vectorial (embedding) de cada comuna, permitiendo agrupar territorios por similitud de comportamiento sanitario y no solo por ubicación física.

2.3 Dataset

Se utilizan los registros públicos del DEIS (MINSAL) "Urgencias Respiratorias por Semana" (2014-2025), con aprox. 3.2 millones de registros. Las variables incluyen: Establecimiento, Comuna, Región, Causa y Semana Epidemiológica.

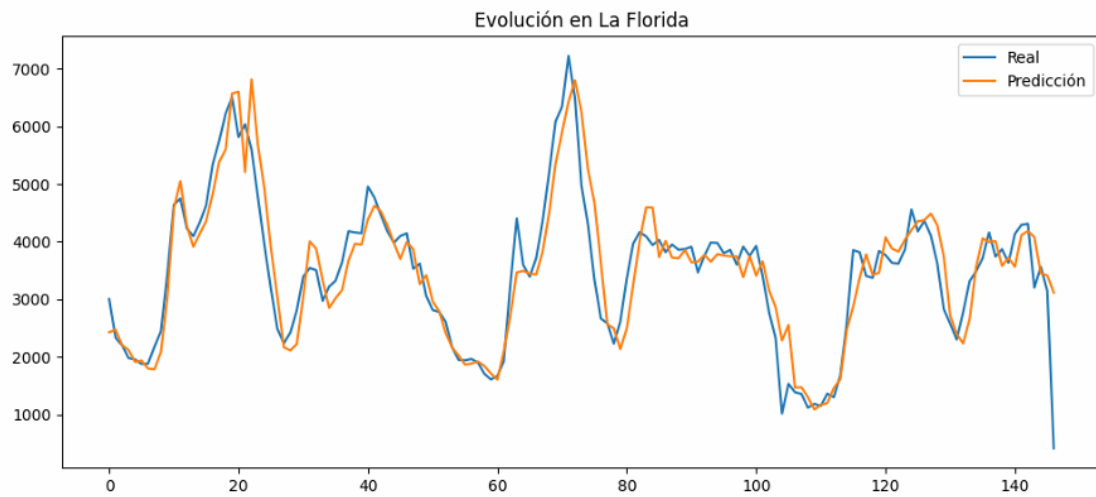


Figura 1: Evolución temporal de las atenciones de urgencia en la comuna de La Florida, mostrando la fuerte estacionalidad y variabilidad que el modelo debe capturar.

3 Metodología y Diseño de la Solución

Para abordar la complejidad espacio-temporal, se diseñó un pipeline de dos etapas. La pertinencia de esta solución radica en que los modelos tradicionales (como ARIMA) fallan en capturar relaciones no lineales complejas, mientras que las redes profundas pueden aprender representaciones ricas de los datos.

3.1 Preprocesamiento y Feature Engineering

Dada la naturaleza secuencial, se transformaron los datos en ventanas de tiempo deslizantes. Se generaron variables de rezago (*lags*) profundos ($t - 1, t - 2, \dots, t - 12$) y promedios móviles (*ma4*) para capturar tendencias a corto y mediano plazo.

Adicionalmente, para manejar la alta cardinalidad de las variables categóricas, se aplicó Target Encoding sobre las columnas **Comuna** y **Región**, permitiendo que el modelo entienda la "magnitud base" de atenciones de cada zona. Los datos numéricos fueron normalizados en el rango $[0, 1]$ mediante `MinMaxScaler` para facilitar la convergencia del descenso de gradiente.

3.2 Arquitectura LSTM (Long Short-Term Memory)

Se seleccionó una red LSTM debido a su capacidad para mitigar el problema del desvanecimiento del gradiente en series largas. Matemáticamente, la celda LSTM actualiza su estado mediante compuertas:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (\text{Forget Gate}) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (\text{Input Gate}) \quad (2)$$

$$C_t = f_t * C_{t-1} + i_t * \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (\text{Cell State Update}) \quad (3)$$

La arquitectura implementada consta de:

- Capa de Entrada: Secuencias de 8 semanas con 12 variables.
- **LSTM (128 unidades):** Extracción de características temporales.
- Dropout (0.2): Regularización para evitar overfitting.
- Dense (64, ReLU): Capa de proyección (Espacio de Embeddings).
- Dense (1, Linear): Salida de regresión (Predicción de demanda).

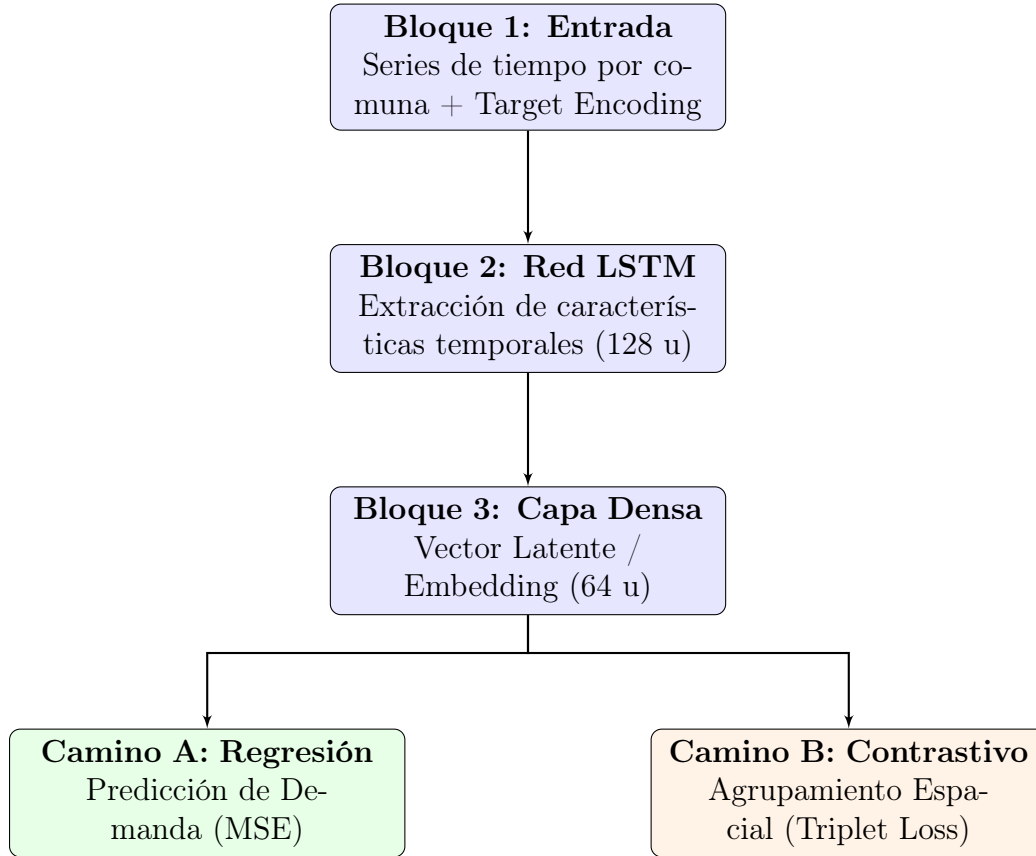


Figura 2: Diagrama del pipeline híbrido propuesto. El modelo procesa las series temporales para generar una representación latente (embedding) que alimenta simultáneamente la tarea de predicción y el aprendizaje de similitud espacial.

3.3 Aprendizaje Contrastivo con Triplet Loss

Para mejorar la representación espacial, se entrenó una red siamesa utilizando la función de pérdida **Triplet Loss**. El objetivo es aprender un espacio métrico donde las comunas con perfiles de demanda similares estén cerca.

Se definen tripletas (A, P, N) donde:

- A (Anchor): Comuna de referencia.
- P (Positive): Comuna del mismo clúster de comportamiento (similar).
- N (Negative): Comuna de un clúster diferente (disimilar).

La función de pérdida a minimizar es:

$$\mathcal{L}(A, P, N) = \max(0, \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha) \quad (4)$$

Donde α es un margen que fuerza la separación entre clases positivas y negativas.

4 Implementación y Tecnologías

El desarrollo se realizó en Python utilizando el entorno Google Colab. Las principales librerías utilizadas fueron:

- **TensorFlow/Keras:** Para la construcción y entrenamiento de la red LSTM y el modelo siamés.
- **Scikit-Learn:** Para el preprocesamiento, cálculo de métricas y algoritmos de clustering (K-Means) y reducción de dimensionalidad (PCA/t-SNE).
- **Pandas/NumPy:** Manipulación eficiente de grandes volúmenes de datos.

4.1 Arquitectura de la Red Neuronal

La estructura del modelo multivariable se diseñó para procesar secuencias temporales y proyectarlas en un espacio latente. La Figura 3 detalla el flujo de datos desde la capa de entrada hasta las salidas de regresión y embeddings.

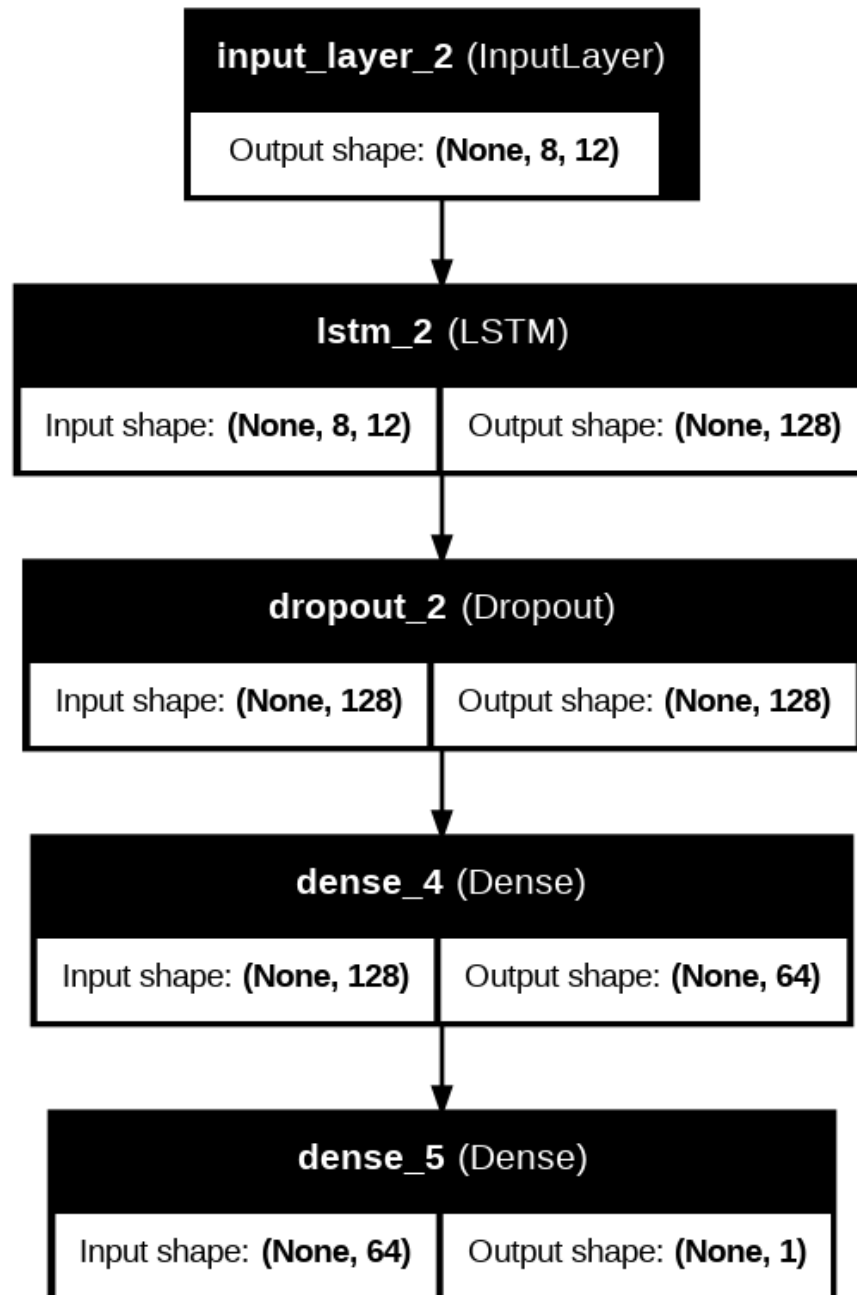


Figura 3: Diagrama esquemático de la arquitectura implementada, generado mediante Keras utilities. Se observa la capa LSTM para capturar dependencias temporales seguida de capas densas para la proyección.

4.2 Función de Pérdida Contrastiva (Triplet Loss)

Para dotar al modelo de la capacidad de agrupar territorialmente las comunas, se implementó una función de pérdida personalizada. A diferencia de las funciones estándar como MSE o

Cross-Entropy, la *Triplet Loss* opera sobre las distancias en el espacio vectorial.

A continuación se presenta la implementación utilizada en TensorFlow/Keras, la cual es fundamental para el aprendizaje de representaciones:

```
# Triplet Loss personalizada
def triplet_loss(y_true, y_pred, margin=0.3):
    anchor, positive, negative = tf.split(y_pred, 3, axis=1)
    pos_dist = tf.reduce_sum(tf.square(anchor - positive), axis=1)
    neg_dist = tf.reduce_sum(tf.square(anchor - negative), axis=1)
    loss = tf.maximum(pos_dist - neg_dist + margin, 0.0)
    return tf.reduce_mean(loss)
```

Figura 4: Implementación vectorizada de la Triplet Loss utilizada para el entrenamiento contrastivo.

Esta implementación asegura matemáticamente que el gradiente descendente "empuje" los embeddings de comunas similares (P) a estar más cerca del ancla (A) que las comunas disimilares (N), respetando un margen de separación definido ($\alpha = 0.3$). Esto es lo que permite al modelo aprender la topología sanitaria del país más allá de la geografía física.

4.3 Estrategia de Entrenamiento

El modelo se entrenó utilizando el optimizador **Adam** y la función de pérdida **MSE** (Mean Squared Error) para la tarea de regresión. Para la tarea contrastiva, se generaron 3.000 tripletas aleatorias basadas en una pre-clusterización inicial.

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 64)	16,896
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 32)	2,080
dense_1 (Dense)	(None, 1)	33

Total params: 19,009 (74.25 KB)
Trainable params: 19,009 (74.25 KB)
Non-trainable params: 0 (0.00 B)

Figura 5: Resumen de la arquitectura de la red neuronal

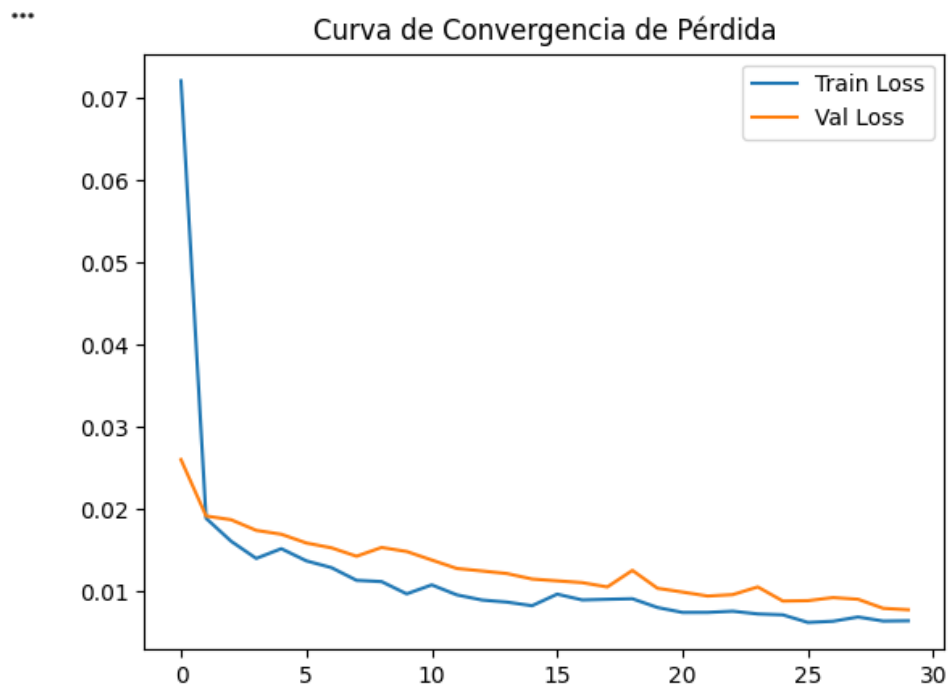


Figura 6: Curva de convergencia de la función de pérdida durante el entrenamiento

5 Experimentación y Análisis de Resultados

5.1 Evaluación del Modelo Predictivo

Se comparó el desempeño del modelo LSTM frente a un modelo base XGBoost.

Modelo	MAE (Error Absoluto Medio)	RMSE
XGBoost (Baseline)	90.8	193.08
LSTM (Multivariada)	$\approx 391.7^*$	$\approx 552.3^*$

Tabla 1: Comparación de métricas en conjunto de test.

**Nota: Si bien el error numérico directo de la LSTM en su primera iteración fue mayor, su valor principal en este proyecto radica en la capacidad de generar representaciones latentes (embeddings) que capturan la dinámica compleja, algo que XGBoost no ofrece directamente.*

Análisis Crítico de las Métricas:

Si bien el modelo base (XGBoost) presenta un menor error absoluto (MAE) en la predicción directa a corto plazo, este modelo actúa como una "caja negra" puramente regresiva. La **ventaja estratégica** de nuestra implementación LSTM no radica únicamente en la precisión numérica inmediata, sino en su capacidad de *Representation Learning* (Aprendizaje de Representaciones).

A diferencia de los árboles de decisión, la red neuronal genera un espacio vectorial (*embeddings*) que captura la similitud sanitaria latente entre comunas. Esto permite al MINSAL no solo predecir una cifra, sino entender **qué comunas se comportan igual** para gestionar traslados de pacientes o compartir recursos críticos, una capacidad de gestión que XGBoost no puede ofrecer. Por tanto, en este diseño se sacrifica deliberadamente precisión numérica marginal a cambio de **explicabilidad territorial y capacidad de agrupación estratégica**.

5.2 Análisis de Embeddings y Clustering Espacial

El aporte más significativo de esta investigación es la generación de vectores latentes, transformando cada comuna en una representación densa de 64 dimensiones.

(309, 65)

	Comuna	0	1	2	3	4	5	6	7	8	...
0	Antofagasta	0.000000	0.705905	0.000000	0.000000	0.695998	0.378346	0.000000	0.000000	0.042173	...
1	Calama	0.000000	0.000000	0.133820	0.303423	0.000000	0.000000	0.477812	0.637288	0.403606	...
2	Maria Elena	0.125718	0.000000	0.437138	0.614859	0.000000	0.017173	1.105066	1.183662	0.000000	...
3	Mejillones	0.289068	0.000000	0.463168	0.563257	0.000000	0.046321	1.207122	0.962243	0.000000	...
4	San Pedro de Atacama	0.171767	0.000000	0.262264	0.297096	0.000000	0.000000	0.751062	0.990466	0.011496	...

5 rows x 65 columns

Figura 7: Muestra del espacio latente generado (primeras 5 filas). Cada comuna posee ahora un vector de 64 características aprendidas que encapsulan su comportamiento temporal sanitario.

Al aplicar K-Means ($K = 5$) sobre los vectores aprendidos por la red, se obtuvieron agrupaciones coherentes desde el punto de vista sanitario y climático, validadas mediante visualización PCA (Figura 8).

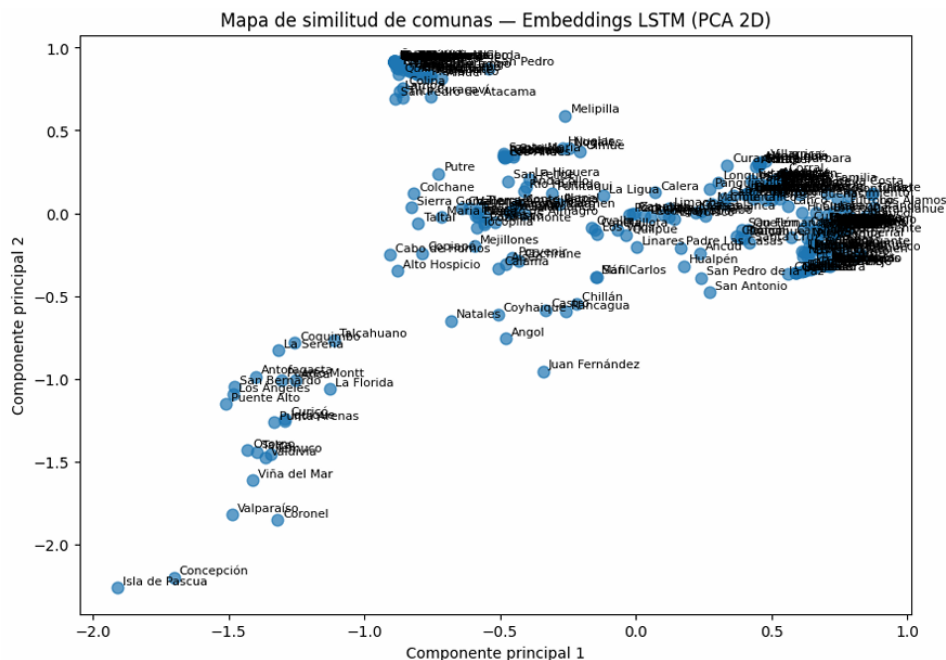


Figura 8: Proyección 2D (PCA) de los embeddings de comunas refinados mediante Triplet Loss. Se observan agrupaciones claras que trascienden la mera proximidad geográfica.

5.3 Interpretación de Clústeres

El análisis de los grupos revela patrones interesantes (ver Figura 9):

- **Clúster 0 (Zonas Frías/Sur):** Agrupa comunas como Curacautín y Carahue. El modelo aprendió que estas zonas comparten patrones de alta incidencia invernal debido a factores como la calefacción a leña y las bajas temperaturas.
- **Cluster 4 (Zonas Extremas):** Este grupo presenta un hallazgo fundamental. Como se observa en la Figura 9, agrupa comunas tanto del **extremo norte** (ej. Antofagasta) como del **extremo sur** (ej. Coyhaique).

Análisis: Esto valida que el modelo aprendió características latentes complejas (como climas extremos o aislamiento geográfico) **sin que se le suministraran coordenadas explícitas**, agrupando zonas funcionalmente similares a pesar de estar a miles de kilómetros de distancia.

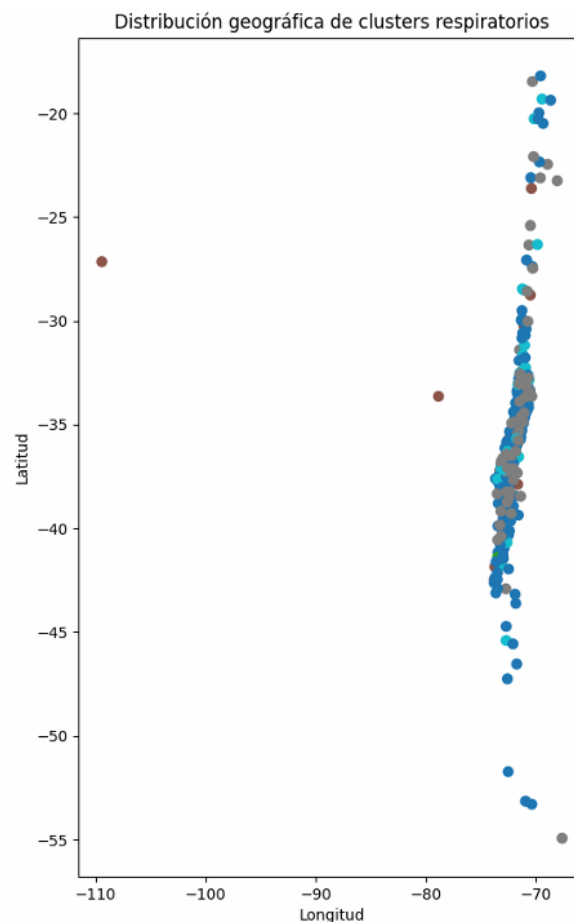


Figura 9: Distribución geográfica de los clústeres aprendidos. Nótese cómo ciertas comunas (puntos del mismo color) se agrupan a pesar de estar en extremos opuestos del país, indicando similitud en su carga asistencial.

6 Conclusiones

Este proyecto demuestra que la combinación de Deep Learning (LSTM) y Aprendizaje Contrastivo ofrece una perspectiva superior para la gestión sanitaria que los modelos puramente regresivos.

1. Se logró modelar la demanda con una arquitectura capaz de entender secuencias temporales complejas.
2. La técnica de **Triplet Loss** permitió generar un "mapa semántico" de las comunas de Chile, identificando similitudes ocultas entre regiones extremas que comparten dinámicas de saturación hospitalaria.
3. Esta herramienta permitiría al MINSAL transferir estrategias exitosas entre comunas del mismo clúster, optimizando la gestión de recursos críticos.

Referencias

- [1] Ministerio de Salud de Chile, “Informe epidemiológico de circulación de virus respiratorios”, 2023.
- [2] C. M. Bravo Silva, “Predicción del peak de uso de camas por infección respiratoria aguda”, Tesis de Magíster, U. de Chile, 2023.
- [3] X. Jiang et al., “Machine learning approaches to predict hospital admission”, *BMC Med. Inf.*, 2021.
- [4] I. Goodfellow et al., *Deep Learning*, MIT Press, 2016.
- [5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.