

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE OBRAS CIVILES

**Estimación de Series de Inundación por Tsunamis Utilizando
Datos de Sistemas de Navegación Global por Satélite (GNSS)
mediante Redes Neuronales**

JORGE NUÑEZ CODOSEO

**Ingeniero Civil
Magíster en Ciencias de la Ingeniería Civil**

12 de agosto de 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: "Estimación de Series de Inundación por Tsunamis Utilizando Datos de Sistemas de Navegación Global por Satélite (GNSS) mediante Redes Neuronales"

Nombre del candidato(a): Jorge Eliseo Nuñez Codoseo

Carrera / Grado: Ingeniería Civil / Magister en Ciencias de la Ingeniería Civil

Campus: Casa Central ; **Departamento:** Obras Civiles

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, **Patricio Catalán Mondaca**, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

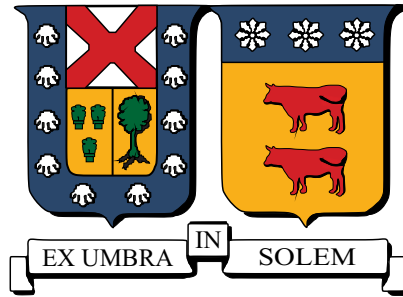
Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 12AGO25 ; Firma:

Estudiante o Candidato(a):

Fecha: 12AGO25 ; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE OBRAS CIVILES

**Estimación de Series de Inundación por Tsunamis Utilizando
Datos de Sistemas de Navegación Global por Satélite (GNSS)
mediante Redes Neuronales**

Memoria de Título y Tesis de Grado presentada por
JORGE NUÑEZ CODOSEO

Como requisito parcial para optar al título de
Ingeniero Civil
y al grado de
Magíster en Ciencias de la Ingeniería Civil

Profesor Guía Dr. Patricio Andrés Catalán Mondaca
Profesor Co-Referente Dr. Carlos Valle Vidal

12 de agosto de 2025

TÍTULO DE LA TESIS

Estimación de Series de Inundación por Tsunamis Utilizando Datos de Sistemas de Navegación Global por Satélite (GNSS) mediante Redes Neuronales

AUTOR

JORGE NUÑEZ CODOSEO

TRABAJO DE TESIS, presentado en cumplimiento parcial de los requisitos para el título de Ingeniero Civil y el grado de Magíster en Ciencias de la Ingeniería Civil de la Universidad Técnica Federico Santa María.

Dr. Patricio Andrés Catalán Mondaca

Dr. Carlos Valle Vidal

Valparaíso, Chile, Abril de 2025.

AGRADECIMIENTOS

Agradecer al profe Pato por depositar su confianza en mí y al profe Carlos por "*prestarme las botas sin conocerme*", dejándome tomar sus asignaturas como ramos libres, los cuales fueron vitales para el desarrollo de este trabajo.

Agradecer también a ANID/FONDEF ID19i10048 "Fortalecimiento del sistema de soporte dedecisiones de tsunami a través de modelado en tiempo cercano al real".

RESUMEN

Los sistemas de alerta temprana ante tsunamis requieren métodos eficientes y precisos para estimar el peligro en tiempo real. Aunque las modelaciones numéricas basadas en las ecuaciones de aguas poco profundas permiten calcular con alta fidelidad la propagación y la inundación por tsunami, su elevado costo computacional y la dependencia de una estimación precisa de la fuente sísmica limitan su aplicación en contextos operativos. En este trabajo, se propone un modelo basado en redes neuronales convolucionales unidimensionales (CNN 1D) que, a partir de registros GNSS de menos de 9 minutos, calcula la serie de inundación en tierra sin necesidad de estimar la fuente ni de esperar registros prolongados en el mar. Se comparan distintas metodologías, incluyendo un enfoque directo; un esquema en dos etapas que primero predice series de la elevación instantánea de la superficie libre (η) y luego la inundación; y una arquitectura Branch-CNN que busca mejorar la predicción de la inundación incorporando series η durante el entrenamiento. Los modelos fueron entrenados con una base de datos sintética compuesta por 13.000 escenarios generados mediante modelaciones numéricas del movimiento cinemático de la corteza terrestre, la propagación del tsunami y la inundación. Los resultados indican que el enfoque en dos etapas alcanza el mejor desempeño general. Por su parte, la arquitectura Branch-CNN mejora el proceso de aprendizaje al ajustar dinámicamente los pesos de pérdida, aunque no supera la precisión del enfoque en dos etapas. Adicionalmente, se validó el modelo utilizando registros GNSS reales del terremoto y tsunami de Illapel 2015 (Mw 8.4), obteniendo resultados altamente precisos para el pronóstico en Viña del Mar (ViB). Si bien en el caso de Coquimbo (CoB) la predicción no es tan exacta, esta coincidió con el nivel de peligrosidad observado, lo que refuerza el potencial de esta herramienta como una alternativa rápida y eficaz para el cálculo de series de tsunamis en sistemas de alerta temprana.

Palabras Claves: Series de inundación por tsunami, Redes neuronales convolucionales (CNN), Sistemas de Navegación Global por Satélite (GNSS).

ABSTRACT

Tsunami early warning systems require efficient and accurate methods to assess hazard in real time. Although numerical simulations based on the shallow water equations can compute tsunami propagation and inundation with high fidelity, their high computational cost and reliance on an accurate estimation of the seismic source limit their applicability in operational contexts. This work proposes a model based on one-dimensional convolutional neural networks (1D-CNN) that, using GNSS records shorter than 9 minutes, directly estimates the inland inundation time series without the need to estimate the source or wait for long offshore recordings. Several methodologies are compared, including a direct approach; a two-stage scheme that first predicts time series of the instantaneous free-surface elevation (η) and then the inundation; and a Branch-CNN architecture that aims to enhance inundation prediction by incorporating η series during training. The models were trained on a synthetic dataset comprising 13,000 scenarios generated through numerical simulations of the kinematic motion of the seafloor, tsunami propagation, and coastal inundation. Results show that the two-stage approach achieves the best overall performance. The Branch-CNN architecture, while improving the learning process by dynamically adjusting loss weights, does not surpass the accuracy of the two-stage scheme. Additionally, the model was validated using real GNSS records from the 2015 Illapel earthquake and tsunami (Mw 8.4), achieving highly accurate predictions for the Viña del Mar (ViB) case. While the prediction for Coquimbo (CoB) was less accurate, it still matched the observed hazard level, highlighting the potential of this tool as a fast and effective alternative for tsunami time series forecasting in early warning systems.

Keywords: Tsunami flood time series, Convolutional Neural Networks (CNN), Global Navigation Satellite Systems (GNSS).

LISTA DE ACRÓNIMOS

1D-CNN Modelos de Redes Neuronales Convolucionales unidimensionales. 1, 2, 5–8, 20

BCNN Branch Convolutional Neural Network. 9

CNN Modelo de Redes Neuronales Convolucionales. 5, 7, 16

CNN-GI Red Neuronal Convolutacional para Entrada GNSS y Salida de Inundación. 24

CNN-GPI Red Neuronal Convolutacional unidimensionales de dos etapas para Entradas GNSS, Propagación Intermedia y Salidas de Inundación. 24

GNSS Global Navigation Satellite System. 2, 3, 6–9, 11, 13–15, 20, 24

LSTM Long Short-Term Memory. 7

ML Machine Learning. 5, 11, 18

MLP Modelo de Perceptrón Multicapa. 5, 6

RNN Modelos de Redes Neuronales Recurrentes. 7

TEWS Tsunami Early Warning Systems. 1

VNMM Variación del Nivel Medio del Mar. 1, 8, 25

CONTENIDOS

AGRADECIMIENTOS	I
RESUMEN	III
ABSTRACT	V
ACRÓNIMOS	V
CONTENIDO	VII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XIII
INTRODUCCIÓN	1
OBJETIVOS	3
1. REVISIÓN BIBLIOGRÁFICA	5
1.1. Aplicaciones de Machine Learning al estudio de Tsunamis	5
1.2. Procesamiento de series temporales mediante Machine Learning	6
1.3. Definición del problema actual	8
2. METODOLOGÍA DE TRABAJO	11
2.1. Base de datos	12
2.2. Análisis de datos	16
2.2.1. Equilibrio de clases	16
2.2.2. Tiempo de inundación	17

2.3. Preprocesamiento	19
2.3.1. Resolución temporal	19
2.4. Modelos de <i>Machine Learning</i>	21
2.4.1. Aspectos Generales de Redes Neuronales Convolucionales 1D	21
2.4.2. Arquitecturas Convolucionales para Modelado de Inundación	25
2.4.2.1. Modelo Directo GNSS–Inundación (CNN-GI)	26
2.4.2.2. Modelo de Dos Etapas GNSS–Propagación–Inundación (CNN-GPI)	27
2.4.3. Red Neuronal Convolutiva Ramificada (BCNN)	28
2.4.4. Branch CNN con Bt-Strategy (BCNN-Bt)	30
2.5. Selección de Hiperparámetros	31
2.6. Entrenamiento, validación y test	34
3. RESULTADOS Y ANÁLISIS	37
4. DISCUSIÓN DE LOS RESULTADOS	45
CONCLUSIONES	49
REFERENCIAS	54

ÍNDICE DE FIGURAS

2.1. Topo-batimetría y dominios anidados utilizados para las simulaciones numéricas de tsunamis con Tsunami-HySEA.	14
2.2. Registro de señales Global Navigation Satellite System (GNSS), propagación e inundación.	15
2.3. Gráficos de dispersión de niveles de peligrosidad	18
2.4. Subsampling de series de tiempo	20
2.5. Diagrama kernel convolucional	22
2.6. Esquema de una capa densa	24
2.7. Esquema del bloque convolucional utilizado en los modelos propuestos.	26
2.8. Esquema de la arquitectura CNN-GI.	27
2.9. Esquema de la arquitectura CNN-GPI.	28
2.10. Esquema de la arquitectura de la red neuronal Branch-CNN con dos ramas.	29
3.1. Distribución del MSE en los distintos conjuntos de prueba.	38
3.2. Matriz de confusión del modelo CNN-GPI evaluado en el conjunto de prueba.	40
3.3. Predicción del modelo CNN-GPI para la zona ViB en el escenario # 9412 (Mw 8.9).	42
3.4. Predicción del modelo CNN-GPI para la zona CoB en el escenario # 9412 (Mw 8.9).	42
3.5. Predicción del modelo CNN-GPI para el evento real de Illapel 2015.	43
4.1. Predicción del modelo CNN-GPI con distintos niveles de ruido en las señales GNSS de entrada.	47

ÍNDICE DE TABLAS

2.2. Conjunto de hiperparámetros probados.	32
2.3. Ranking de las mejores combinaciones de hiperparámetros en validación.	33
2.4. Conjunto de hiperparámetros utilizados en <i>Bt-Strategy</i>	34
2.5. Las combinaciones de hiperparámetros con mejor desempeño para BCNN con la estrategia Bt-Strategy.	34
3.1. Métricas de desempeño del mejor modelo de red neuronal (NNM) en cada ubicación. Los datos resaltados en negrita corresponden al NNM final para cada boya objetivo.	37
3.2. Métricas de desempeño en el conjunto de prueba	39
4.1. Métricas de desempeño del modelo CNN-GPI al agregar distintos porcentajes de ruido browniano a las series GNSS de entrada.	46

INTRODUCCIÓN

Contar con una estimación rápida y precisa de posibles inundaciones por tsunamis sería un complemento valioso para los sistemas de alerta actuales en todo el mundo. Sin embargo, esto sigue siendo un desafío debido a la complejidad inherente al problema. Aunque la propagación e inundación por tsunamis están bien resueltas mediante modelaciones numéricas que emplean las ecuaciones de aguas poco profundas (*shallow water equations*), su aplicabilidad en situaciones de emergencia no es viable debido al elevado costo computacional que requieren. Además, estos métodos dependen de una estimación rápida y confiable de la fuente del terremoto, lo cual es difícil de obtener en tiempo real.

Los sistemas de alerta temprana de tsunamis (Tsunami Early Warning Systems (TEWS)) actuales han optado por estrategias que priorizan la velocidad de evaluación del peligro por sobre la precisión. Un ejemplo de esto son los TEWS en países como Japón y Chile ([Kamigaichi et al., 2009](#); [Catalán et al., 2020](#)), donde se han desarrollado bases de datos previamente calculadas en lugar de depender de cálculos en tiempo real. En estos sistemas, se utilizan criterios de máximo riesgo variando la ubicación del epicentro a lo largo del área de ruptura y midiendo la altura máxima en puntos predefinidos (aproximadamente a 200 metros de profundidad). Posteriormente, las alturas máximas del oleaje en mar abierto se proyectan hacia puntos cercanos a la costa mediante la aplicación de la Ley de Green ([Green et al., 1838](#)). No obstante, actualmente se desconoce la relación entre la elevación instantánea de la superficie libre ($\eta(t)$) y la profundidad de inundación ($d(t)$).

Los recientes eventos de campo cercano ocurridos en la costa de Chile, como el de Maule 2010 (Mw 8.8) e Illapel 2015 (Mw 8.4), registraron tiempos de arribo de las primeras ondas a la costa cercanos a los 20 minutos. Este margen es muy limitado para reaccionar ante eventos de este tipo, lo que dificulta la obtención de registros de las series de elevación instantánea de la superficie libre, necesarias para pronosticar la inundación mediante Modelos de Redes Neuronales Convolucionales unidimensionales (1D-CNN), como el utilizado por ([Makinoshima et al., 2021](#)). En ([Núñez et al., 2022](#)) abordan la escasez de sensores y el tiempo requerido para obtener registros reales en mar, mediante 1D-CNN para calcular rápidamente series de inundación por tsunamis a partir de series de registros de elevación instantánea de la superficie libre ($\eta(t)$) calculadas a partir de modelados numéricos de baja resolución.

Los resultados demostraron la factibilidad de calcular series de inundación a partir de registros de elevación instantánea de la superficie libre ($\eta(t)$) generados con modelaciones de baja

resolución. Esta metodología permite realizar los cálculos en aproximadamente 10 segundos, alcanzando una precisión comparable a la de un modelado de alta resolución, cuyo tiempo de computo es del orden de miles de segundos. Sin embargo, este enfoque requiere contar con una estimación rápida y confiable de la fuente sísmica.

En este estudio busca superar dichas limitaciones, explorando la factibilidad de utilizar datos GNSS, los cuales registran los movimientos verticales y horizontales del suelo y están disponibles de manera inmediata tras el inicio del sismo. Esto resulta especialmente relevante en el contexto de Chile, que cuenta con una extensa red de estaciones GNSS distribuidas a lo largo de su territorio.

Por lo tanto, el objetivo principal de este trabajo es desarrollar, ajustar y validar modelos de redes neuronales que utilicen exclusivamente señales GNSS durante la etapa de pronóstico, aprovechando los registros disponibles en tiempo real durante un evento sísmico. Finalmente, se evalúa el desempeño del modelo utilizando datos GNSS registradas durante el evento de Illapel 2015.

OBJETIVOS

Objetivos Generales

- **OG1:** Ajustar y validar un modelo de redes neuronales con el propósito de calcular series de profundidad de inundación por tsunamis en puntos de control en tierra en tiempo menor al real, utilizando como única variable de entrada datos de sistemas de navegación global por satélite (GNSS), con una precisión cercana a la de un modelado de alta resolución.

Objetivos Específicos

- **OE1:** Desarrollar y ajustar distintas configuraciones de redes neuronales para incorporar datos de registros de elevación instantánea de la superficie libre ($\eta(t)$) en la etapa de aprendizaje del modelo.
- **OE2:** Aplicar la metodología en bahías con comportamientos hidrodinámicos distintos, específicamente en Valparaíso y Coquimbo.
- **OE3:** Evaluar el desempeño del modelo utilizando un conjunto de nuevos escenarios no considerados durante las fase de entrenamiento y validación, reportando distintas métricas para comprobar su robustez.
- **OE4:** Validar el modelo previamente ajustado utilizando datos reales del evento de Illapel 2015 ($M_w 8.4$), comparando las predicciones generadas con las observaciones reales registradas.

Capítulo 1

REVISIÓN BIBLIOGRÁFICA

1.1. Aplicaciones de Machine Learning al estudio de Tsunamis

Las aplicaciones de Machine Learning (ML) en el estudio de inundaciones por tsunamis son relativamente recientes, debido a la complejidad no lineal de este problema. Un estudio realizado por *Fauzi and Mizutani (2020)* emplea dos enfoques de ML para calcular mapas de inundación máxima por tsunamis en tiempos cercano al tiempo real. Para ello, utilizan un Modelo de Redes Neuronales Convolucionales (CNN) que reciben como entrada mapas de inundación máxima de nuevos eventos generados mediante modelación de baja resolución, y los asocia o clasifica con mapas de inundación máxima obtenidos a partir de modelaciones de alta resolución, previamente ajustados por el modelo. Además, aplican un Modelo de Perceptrón Multicapa (MLP), compuesto por 5 capas ocultas, que a partir de mapas de inundación máxima de baja resolución, recrea mapas de inundación máxima con una resolución comparable a una modelación de alta resolución. Estos modelos se probaron en un escenario hipotético de terremoto de Mw 8.7 en Japón. Los resultados son prometedores, aunque se ven limitados por el número reducido número de escenarios utilizados en el entrenamiento (328 escenarios).

Siguiendo esta línea, *Mulia et al. (2020)* profundizan en el trabajo de *Fauzi and Mizutani (2020)* y desarrollan una base de datos con 532 escenarios de tsunamis, sobre la cual ajustan modelos de redes feedforward con una, dos, tres y hasta cuatro capas ocultas, con el objetivo de capturar atributos más complejos y mejorar el rendimiento en la recreación de mapas de inundación máxima. Sin embargo, ninguno de estos trabajos aborda el estudio de las series de tiempo de inundación, las cuales contienen información importante como el tiempo de arribo, el máximo de inundación y el comportamiento temporal de la inundación, elementos fundamentales para la implementación de un sistema de alerta temprana.

Makinoshima et al. (2021) ajustan un 1D-CNN para recrear series de inundación de tsunamis en la bahía de Senday (Japón) generados por terremotos con magnitudes en el rango de

Mw 9.0 a 9.2, logrando una alta precisión. Este modelo utiliza observaciones reales de una red densa de tsunamímetros, y adicionalmente probaron incorporando datos geodésicos (3 boyas GPS en mar y 3 estaciones GNSS en tierra). Sin embargo, una posible desventaja es la necesidad de contar con una red de monitoreo costosa de implementar y el tiempo de registros, cercano a los 30 minutos, para un pronóstico confiable.

Así, *Núñez et al. (2022)* abordan la problemática de la falta de sensores en alta mar y la limitación de tiempo para registros reales de series de la elevación instantánea de la superficie libre (η), centrándose en el contexto de Chile. Para abordar este desafío, ajustan un modelo 1D-CNN que utilizan 6 series de registros η obtenidas de modelados de baja resolución y sin usar registros geodésicos (GNSS). Este enfoque se aplicó en dos áreas de estudio con comportamientos hidrodinámicos distintos, y los resultados obtenidos son alentadores. Sin embargo, es importante destacar que esta metodología aún requiere una estimación precisa y en tiempo real de la fuente del sismo, un problema que aún no ha sido completamente resuelto.

En contraste, *Rim et al. (2022)* se enfocaron en calcular las formas de onda de tsunamis en puntos cercanos a la costa, utilizando modelos de CNN. En este caso, emplearon exclusivamente datos de sistemas de navegación global por satélite (GNSS) como la única variable de entrada. Los resultados reportados son prometedores, ya que lograron recrear hasta 6 horas de elevación instantánea de la superficie libre (η) con menos de 9 minutos de registros GNSS. Sin embargo, este estudio no considera la inundación, y además, el área de interés se asemeja a un canal unidimensional, el cual puede actuar como filtro de la señal del tsunami. Por esta razón, es deseable indagar si esto es generalizable a costas abiertas con distintos comportamientos hidrodinámicos.

1.2. Procesamiento de series temporales mediante Machine Learning

Con los avances en la capacidad de cómputo y la creciente disponibilidad de grandes conjuntos de datos, se ha vuelto posible entrenar modelos de redes neuronales cada vez más complejos. Esto ha dado lugar al paradigma del aprendizaje profundo, que se caracteriza por la inclusión de numerosas capas ocultas y la capacidad de capturar y construir atributos más complejos a partir de atributos simples (*LeCun et al., 2015*). El aprendizaje profundo ha demostrado su eficacia en una variedad de aplicaciones, como el procesamiento de imágenes, videos, voz y texto.

En esta investigación, nuestro enfoque se dirige al procesamiento de señales, específicamente en tareas del tipo “*sequence to sequence*” (secuencia a secuencia). Entre los tipos de redes neuronales ampliamente utilizados en esta variedad de tareas, incluyendo el procesamiento de señales, se encuentran las redes *FeedForward* o MLP. Estos modelos operan bajo la suposición de independencia entre las variables, lo que significa que después de procesar un ejemplo, se pierde toda la información sobre el estado previo de la red. Esto presenta un desafío en situaciones en las que las variables tienen dependencias temporales o espaciales (*Sutskever et al., 2014; Lipton et al., 2015*).

Uno de los primeros intentos de incorporar dependencias temporales en modelos de redes neuronales se atribuye a *Jordan (1986)*, quien introdujo enlaces recurrentes para permitir el flujo de información a través de múltiples pasos de tiempo. Este marcó el inicio de Modelos de Redes Neuronales Recurrentes (RNN). A pesar de que estos modelos inicialmente enfrentaron desafíos, como la alta demanda computacional necesaria para su calibración y problemas relacionados con el desvanecimiento o explosión de los gradientes en tareas que requerían aprender dependencias de largo plazo (*Hochreiter et al., 2001*), han resurgido con el avance en la capacidad computacional y por mucho tiempo fueron indiscutibles en tareas de procesamiento de secuencias. Entre las arquitecturas modernas de RNN, destacan las Long Short-Term Memory (LSTM), introducidas por *Hochreiter and Schmidhuber (1997)* para abordar el problema del desvanecimiento del gradiente en dependencias a largo plazo.

En el último tiempo, debido al éxito de las CNN en el procesamiento de imágenes y videos, gracias a su capacidad para aprender objetos y patrones complejos, ha impulsado su aplicación al procesamiento de señales. En este contexto surgieron los modelos de 1D-CNN (*Kiranyaz et al., 2015*), que rápidamente demostraron un rendimiento destacado en diversas aplicaciones del procesamiento de señales (*Kiranyaz et al., 2021*). Los modelos 1D-CNN con arquitecturas compactas, donde una arquitectura de red neuronal compacta se refiere a un diseño eficiente, con un número reducido de parámetros y/o capas, sin comprometer significativamente su capacidad de aprendizaje, ofrecen un mejor rendimiento cuando se dispone de datos limitados. Además, no requieren hardware especializado, ya que pueden entrenarse utilizando la CPU de una computadora estándar (*Kiranyaz et al., 2021*).

Por otra parte, los modelos Branch Convolutional Neural Network (BCNN) han captado el interés en problemas de clasificación de imágenes, ya que, mediante la estrategia de entrenamiento *Bt-Strategy*, permiten realizar un aprendizaje jerárquico, desde lo general hacia lo específico (*Zhu and Bain, 2017*). Si bien su uso no es común en problemas de secuencias, se puede forzar al modelo BCNN a aprender representaciones útiles mediante la inclusión de

salidas auxiliares (ramas) con dependencias físicas, que guían al modelo a identificar patrones relevantes antes de generar la salida principal.

1.3. Definición del problema actual

El trabajo realizado por [Makinoshima et al. \(2021\)](#) es uno de los más completos para el cálculo en tiempo real de series de inundación por tsunami, ya que realizaron un análisis de sensibilidad considerando los tiempos de registro de los sensores en el mar y la presencia o ausencia de datos geodésicos (registros GPS en mar y registros GNSS en tierra). Los autores señalaron que, para tiempos cortos de registros, la incorporación de datos geodésicos ayudaba a disminuir el error. No obstante, para tiempos de registro más largos (cerca de los 40 minutos), el uso de estos datos no mostraba una mejora significativa en comparación con su ausencia, dado que la señal del tsunami domina el comportamiento. Esto resalta la influencia de cada variable según la duración del registro y recalca la importancia de utilizar datos de elevación instantánea de la superficie libre (η) para obtener resultados precisos en el cálculo de series de profundidad de inundación.

Para abordar el desafío asociado con la disponibilidad de instrumentación y elevado tiempo requerido para el registro de datos en mar utilizados como entrada en el modelo de red neuronal (1D-CNN), [Núñez et al. \(2022\)](#) llevaron a cabo un enfoque similar al de [Makinoshima et al. \(2021\)](#). Ajustaron un modelo 1D-CNN que, en lugar de utilizar registros reales de series η , recibía series de registros provenientes de modelados numéricos de baja resolución. Utilizaron una base de datos más completa en cuanto a la magnitud sísmica (Mw 8.0 - 9.2) y reportaron excelentes resultados en el desempeño del modelo para el cálculo de series de inundación, destacando el uso de la función de activación ReLu en la capa de salida la cual ayudó a resolver el problema de valores de inundación negativos y obtener series completamente nulas en los casos que no generaban inundación, ayudando a disminuir el número de alertas falsas.

Aunque esta metodología resuelve el problema de disponer de una red de sensores en el mar y un tiempo prolongado de registro (como en el caso de [Makinoshima et al. \(2021\)](#)), aún plantea el desafío de obtener una estimación confiable de la fuente del terremoto en tiempo real para utilizarla como condición inicial durante el modelado numérico de baja resolución. No disponer de una fuente confiable no garantiza la concordancia de los resultados del modelo ajustado y los efectos reales observados. En este contexto, los resultados obtenidos por [Rim et al. \(2022\)](#) sugieren que es factible utilizar datos GNSS como variable de entrada en un modelo de red neuronal para calcular la elevación instantánea de la superficie libre (η).

Estas series podrían utilizarse posteriormente para ajustar un segundo modelo similar al de [Makinoshima et al. \(2021\)](#) o [Núñez et al. \(2022\)](#), con el fin de obtener la serie de inundación. Esto es especialmente relevante en países como Chile, que cuenta con una extensa red de estaciones GNSS. Al utilizar registros GNSS reales, se podría superar la limitación asociada a la estimación en tiempo real de la fuente sísmica.

Sin embargo, un estudio reciente realizado por [Rim et al. \(2024\)](#) identificó un desafío crítico asociado al ruido presente en los registros reales de datos GNSS. Para abordar este problema, llevaron a cabo un análisis de sensibilidad mediante la introducción de pequeñas perturbaciones en las señales GNSS con el fin de emular la presencia de ruido real que normalmente se encuentra en estos registros. Los resultados de este estudio revelaron diferencias en los pronósticos cercanas al 24 % e incluso identificaron casos más extremos con diferencias superiores al 50 % en ejemplos adversos que fueron generados de manera intencional para evaluar el desempeño del modelo. Este hallazgo destaca la necesidad crítica de realizar un adecuado filtrado de los datos GNSS antes de utilizarlos como entrada para garantizar el óptimo rendimiento del modelo, especialmente en un contexto de aplicaciones en situaciones de eventos reales.

A partir de lo anterior, se formula la siguiente hipótesis de investigación: **“Es factible obtener predicciones confiables de series de profundidad de inundación por tsunami utilizando únicamente señales GNSS como variables de entrada en un modelo de redes neuronales”**

Para evaluar esta hipótesis, se proponen las siguientes alternativas de investigación para abordar la pregunta propuesta:

- Utilizar un enfoque similar al de [Rim et al. \(2022\)](#), en el cual los registros GNSS se emplearían para recrear las formas de onda en puntos del mar. Estas formas de onda se utilizarían como entradas para ajustar un modelo similar al utilizado en [Núñez et al. \(2022\)](#), lo que permitiría recrear las series de inundación en puntos en tierra.
- Considerar la posibilidad de ajustar un nuevo modelo capaz de establecer una relación directa entre los registros GNSS y las series de inundación en tierra.
- Una tercera opción, menos evidente, sería utilizar un modelo de red *BCNN* ([Zhu and Bain, 2017](#)), que permitiría que la red, mediante la estrategia de ajuste *Bt-Strategy* ([Zhu and Bain, 2017](#)), extraiga o aprenda características de las series en el mar en sus primeras capas, para luego relacionarlas con la salida (inundación) en sus capas posteriores. Este enfoque permitiría recrear tanto las series en el mar como las series

de inundación en tierra.

Capítulo 2

METODOLOGÍA DE TRABAJO

El rendimiento de un modelo ML en la resolución de tareas específicas depende directamente de la disponibilidad de una base de datos adecuada. Esta base de datos debe contener una amplia variedad de ejemplos que el modelo pueda utilizar durante su fase de entrenamiento para ajustar los parámetros libres que lo componen (pesos y sesgos). Sin embargo, en el caso de Megaterremotos, la obtención de registros reales es limitada debido a la baja ocurrencia de eventos de alta magnitud capaces de desencadenar tsunamis significativos.

Para superar esta limitación, es necesario generar y trabajar con bases de datos sintéticas, lo que permite modelar miles de posibles escenarios de terremotos con el potencial de generar tsunamis peligrosos en las costas de Chile. Una vez generados los modelos de ruptura de los distintos sismos, se realizan modelados numéricos para determinar el movimiento cinemático de la corteza terrestre, así como la propagación y la posterior inundación generada por los tsunamis, producto de la perturbación inicial en la superficie libre del océano, asociada al desplazamiento cosísmico.

En cada modelado numérico se registran las series de desplazamientos verticales y horizontales de la corteza, equivalentes a los registrados por estaciones GNSS, junto con la elevación instantánea de la superficie libre del mar y la profundidad de inundación en los diferentes puntos de interés.

Estas series se utilizan para entrenar modelos de redes neuronales, con el objetivo de que estos puedan relacionar los desplazamientos del suelo registrados por las estaciones GNSS con la profundidad de inundación que podría producir un eventual tsunami. Dado que no existe una relación directa y evidente entre estas variables, se estudiará la factibilidad de incorporar, como variable intermedia oculta, las series de elevación instantánea de la superficie libre del mar, actuando como un puente entre la entrada y la salida del modelo. Considerando que estos registros en el mar contienen información del comportamiento propio del tsunami en la zona costera y, además, permiten representar de forma más completa y realista la secuencia física del fenómeno —desde su generación, pasando por la propagación, hasta la inundación—.

Se plantea la hipótesis de que esta información adicional podría mejorar el desempeño del modelo en la predicción de las series de inundación.

Para facilitar el aprendizaje, se realiza un análisis preliminar de los datos, centrado en el balance de clases (entendido como los escenarios que generan y no generan inundación), así como los tiempos de arribo, el tiempo del máximo de inundación y los valores máximos alcanzados. Además, se lleva a cabo un preprocesamiento de los datos, variando la resolución y la duración de los registros.

Una vez seleccionados los conjuntos de entrenamiento, validación y prueba, se exploran distintas configuraciones de modelos de redes neuronales. Para una presentación más clara y ordenada, la metodología se estructura en las siguientes secciones: 1) Base de datos, 2) Análisis de datos, 3) Preprocesamiento, 4) Modelos de *Machine Learning*, 5) Selección de Hiperparámetros, 6) Entrenamiento y Validación.

2.1. Base de datos

En este estudio, se ha generado una base de datos de sismos y tsunamis que contempla 13.000 escenarios distribuidos uniformemente en magnitud, variando en el rango de Mw 8.0 a 9.2. Cada modelo de falla que compone esta base de datos ha sido generado mediante el código de acceso libre MudPy ([Melgar et al., 2016](#)). Este código recibe como parámetro de entrada la geometría de la zona de estudio, sobre la cual se generan los distintos modelos de ruptura. Para este trabajo, se utilizó la geometría correspondiente a la zona 2 del estudio de [Poulos et al. \(2019\)](#) (ver Figura 2.1), compuesta por 2.131 subfallas de 10 km × 10 km. Esta resolución es suficiente para determinar, mediante la expansión de Karhunen-Loève (K-L), una distribución estocástica del deslizamiento dentro del área de ruptura.

Para cada modelo de falla generado, se realiza calcula el movimiento cinemático de los deslizamientos, así como de la propagación e inundación del tsunami. El movimiento cinemático de los deslizamientos se modela mediante el software MudPy, que emplea la distribución del deslizamiento y las propiedades cinemáticas mediante un enfoque basado en la multiplicación de matrices de la Función de Green (GF) elastodinámica para sintetizar formas de onda en distintos puntos definidos (estaciones GNSS). Una vez construida la matriz GF, esta se multiplica por el vector de deslizamiento (S), obteniendo así las formas de onda de deslizamiento superficial de la corteza terrestre en una componente vertical y dos horizontales ([Melgar et al., 2016](#)).

Por otro lado, el modelado numérico de propagación e inundación del tsunami se realizó con el software TsunamiHySea (*Macías et al., 2017*), el cual resuelve el sistema de ecuaciones no lineales de aguas poco profundas. A partir del modelo de falla y del modelo topo-batimétrico, se determinan las condiciones iniciales y de borde necesarias para la simulación. El modelo topo-batimétrico está compuesto por cuatro mallas anidadas con resoluciones de 30, 7.5, 1.875 y 0.234 segundos de arco, respectivamente, como se muestra en la Figura 2.1. La malla de mayor resolución cubre el área urbana de las ciudades de Valparaíso, Viña del Mar, Coquimbo y La Serena.

En cada modelado, se registran las series temporales de desplazamiento del suelo (vertical y horizontal), variación de la superficie libre del mar y la profundidad de inundación. Para llevar a cabo estos registros, se ubicaron conjuntos de boyas virtuales en el mar y en tierra, especificando las coordenadas geográficas de cada una (latitud y longitud). Con esta información, el software MudPy y TsunamiHySea registran los datos asociados a la cuadrícula más cercana a cada coordenada especificada, según el paso de tiempo determinado. Las boyas virtuales simulan sensores reales que registrarían información relacionada con el desplazamiento del suelo y el tsunami. Se utilizó un conjunto de 48 estaciones GNSS distribuidas a lo largo de todo el territorio chileno (círculos blancos en la Figura 2.1) y configuraciones de seis boyas en el mar (triángulos de color cyan en la Figura 2.1), distribuidas en la isóbata de 50 metros de profundidad, con el objetivo de abarcar toda la extensión de la bahía en estudio. La motivación detrás de este esquema de modelación y registro de datos es capturar los efectos locales que pueden variar a lo largo de la bahía. Los sensores virtuales de registros de profundidad de inundación se ubicaron en puntos cercanos a la costa de las ciudades de Viña del Mar (ViB) y Coquimbo (CoB) (estrellas de color rojo en la Figura 2.1), usando sólo dos puntos para efectos de la simulación.

En la Figura 2.2 se presenta: a) la deformación inicial de la superficie de un escenario sísmico; b) el dominio local de la bahía de Valparaíso con los sensores virtuales antes detallados; y c) primero, el registro de las tres componentes (una vertical y dos horizontales) registradas por la estación GNSS ubicada en la zona; segundo las líneas de color cyan representan los seis registros de η ; y por último, la línea roja representa el registro de profundidad de inundación en el sensor ubicado en la ciudad de Viña del Mar (ViB).

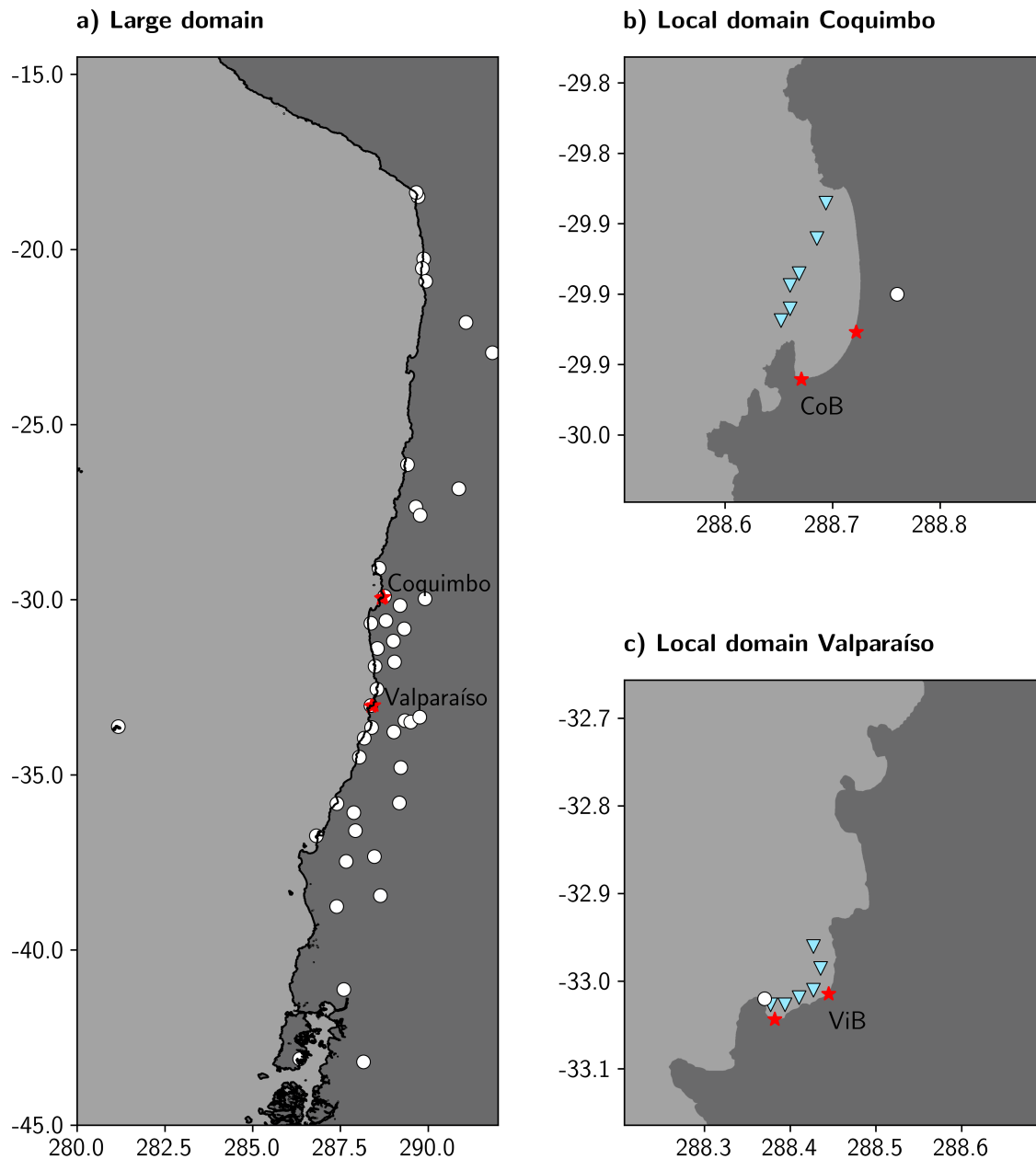


Figura 2.1: Topo-batimetría y dominios anidados utilizados para las simulaciones numéricas de tsunamis con Tsunami-HySEA. a) Nivel 1, con una resolución de 1 minuto de arco; los puntos blancos representan las estaciones GNSS, y las estrellas rojas marcan bahías en estudio: Valparaíso y Coquimbo. b) y c) Nivel 3, correspondiente a los dominios locales con una resolución de 1.875 segundos de arco; los triángulos cian indican las boyas en el mar, mientras que los puntos rojos representan los puntos de registro en tierra.

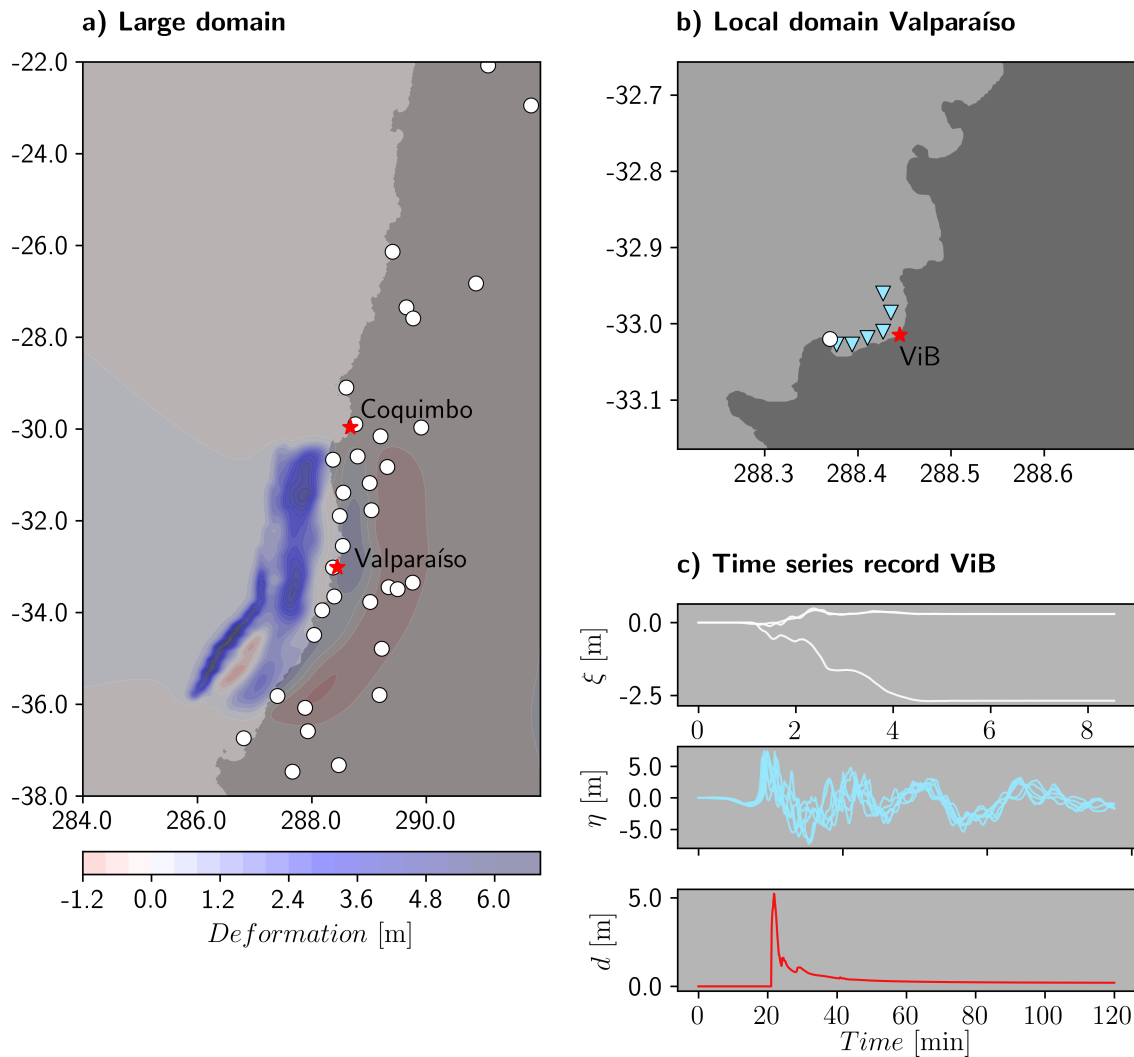


Figura 2.2: Registro de señales GNSS, propagación e inundación. a) Deformación inicial de la superficie asociada a un evento sísmico. b) Estaciones de registro GNSS (círculo blanco), boyas en el mar (triángulos cyan) y sensores en tierra (estrellas rojas). c) Series de registro: las primeras tres series (blanca) corresponde al registro GNSS (una componente vertical y dos horizontales), la segunda (cyan) representa los registros de las seis boyas en el mar, y la última (roja) muestra el registro de inundación en tierra.

2.2. Análisis de datos

2.2.1. Equilibrio de clases

Para el correcto entrenamiento de los modelos de ML, es el análisis de la proporción de casos que generan o no generan inundación, lo que permite determinar si la base de datos presenta un balance o desbalance de clases. Para cada una de las series de tiempo de inundación de los 13.000 escenarios utilizados en este estudio, se asignó una etiqueta: “Inunda” o “NoInunda”. Una serie se clasificó como Inunda si presentaba un valor máximo mayor a cero en los puntos de medición en tierra, y como NoInunda si resultaba ser completamente nula (ver Tabla 2.1).

Tabla 2.1: Distribución de escenarios que inundan y no inundan en cada punto objetivo en tierra.

Boya	Zona	Elev. m.s.n.m	Sí Inunda [%]	No Inunda [%]
CoB	Coquimbo	1.96	63.9	36.1
LSB	La Serena	2.69	41.0	59.0
VaB	Valparaíso	4.56	9.0	91.0
ViB	Viña del Mar	4.77	15.1	84.9

En este caso, la amplia gama de magnitudes sísmicas consideradas podría inducir a que una proporción significativa de escenarios no genere inundación. Por el contrario, considerar únicamente eventos de gran magnitud sesgaría el conjunto de datos hacia aquellos casos que sí producen inundación. La Tabla 1 muestra los porcentajes de “Inunda” y “NoInunda” para cada zona de estudio. Se observa que la ubicación de CoB es más propensa a inundarse, mientras que VaB es la menos propensa a sufrir inundación. Esta diferencia se explica principalmente por la altitud (en metros sobre el nivel del mar (m s.n.m)) de cada zona, así como a efectos locales como, la resonancia en el caso de CoB.

Dado el alto desbalance observado en la boya ViB (ver Tabla 2.1), se decidió mantener solo un 20 % de los escenarios que no presentan inundación, con el objetivo de evitar que el modelo se enfoque exclusivamente en aprender los casos sin inundación. En cambio, para el caso de CoB, se conservaron todos los escenarios, favoreciendo así el aprendizaje de los casos que sí presentan inundación.

2.2.2. Tiempo de inundación

El intervalo de tiempo que se modela, junto con la resolución temporal, determina la longitud o el número de puntos en la serie de tiempo. Aunque las series originales abarcan 6 horas de propagación e inundación de tsunamis. Se busca reducir el intervalo de tiempo modelado o la longitud de la serie de tiempo, lo que permitiría utilizar modelos más pequeños que requieran menos recursos en la etapa de entrenamiento y validación.

Para determinar el intervalo de tiempo a modelar mínimo sin pérdida de información relevante, se realizó un análisis del tiempo de arribo (entendido como el momento en que una celda, inicialmente seca, se inunda) y del tiempo en que se alcanza la profundidad máxima de inundación en los puntos de observación en tierra. Basándonos en los resultados obtenidos sobre la distribución de los tiempos de arribo y el tiempo del máximo de inundación (ver Figura 2.3), en el caso de Viña del Mar y Valparaíso, la mayoría de los arribos (94 %) ocurre dentro de la primera hora de registro, mientras que el 91 % de los máximos de inundación se alcanza dentro de las dos primeras horas.

Basados en los comportamientos observados en las distintas bahías, se definieron los siguientes criterios: el tiempo de modelado mínimo debe incluir al menos un 98 % de los arribos y un 90 % de los máximos registrados. Esto permitió reducir las series de tiempo a cuatro horas en Coquimbo-La Serena y a dos horas en Valparaíso-Viña del Mar. La diferencia en los tiempos entre ambas bahías se debe principalmente a efectos locales propios de cada una, donde se puede ver que ambas ciudades presentan comportamientos muy distintos. ViB tiene tiempos cortos de arribo, pero presenta dos clúster de tiempo máximo con una clara separación de profundidad máxima. CoB presenta una distribución muy uniforme en ambas variable.

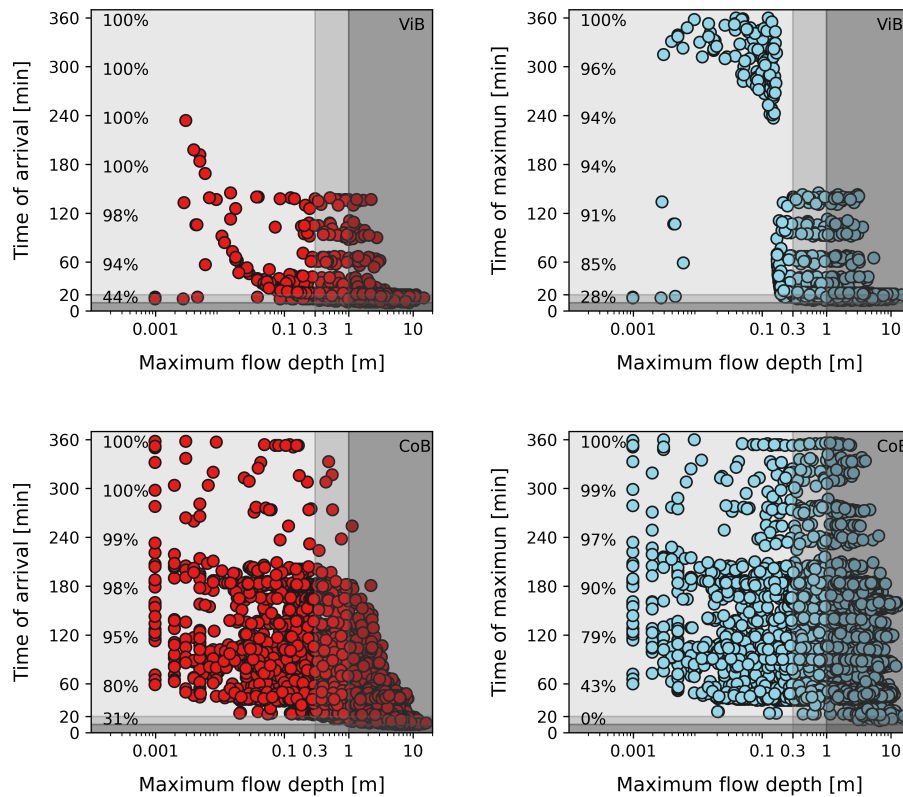


Figura 2.3: Gráficos de dispersión de la distribución conjunta, donde los gráficos a la izquierda muestran la profundidad máxima del flujo y el tiempo de llegada del tsunami; los gráficos a la derecha muestran la profundidad máxima del flujo y el tiempo hasta el máximo, para todos los escenarios (representados por puntos). Las regiones coloreadas verticalmente distinguen entre los valores de amplitud de tsunami utilizados para categorizar el peligro *Catalán et al. (2020)*, mientras que los colores horizontales distinguen entre llegadas muy rápidas, rápidas y tardías, siguiendo *Zamora et al. (2021)*. Los valores porcentuales a la izquierda indican la fracción acumulada de eventos en función del tiempo.

2.3. Preprocesamiento

Un adecuado preprocesamiento de datos puede marcar una gran diferencia en el rendimiento de un modelo de redes neuronales. Por lo tanto, se llevó a cabo un estudio de la resolución temporal y del tiempo de modelado requerido para las distintas series, que guarda una relación directa con la profundidad y el ancho en los modelos CNN. Un equilibrio efectivo entre estas variables garantiza modelos con mejor rendimiento y eficientes en los recursos computacionales necesarios durante el entrenamiento (*Tan and Le, 2019*). Se espera que la variación en la resolución conserve características de interés en la serie, como los tiempos de arribo y los valores máximos. Asimismo, se investigó el balance de clases, entendiéndose este como el porcentaje de ejemplos que provocan y no provocan inundación para los distintos puntos de interés. Estos análisis se detallan a continuación:

2.3.1. Resolución temporal

Consiste en variar la resolución temporal de las series modeladas del tsunami. Aunque originalmente se registraron cada 5 segundos, esta resolución podría representar un exceso de información para este tipo de datos, ya que se trata de olas de período largo, en el orden de varios minutos. Por lo tanto, es factible realizar un submuestreo (subsampling) que conserve la información relevante de la serie temporal. Sin embargo, a diferencia de las series de tiempo en el mar, las series de inundación presentan características de corta duración, por lo que el submuestreo debe realizarse en función de estas, evitando así la pérdida de información relevante en la serie objetivo.

Con este propósito, se llevó a cabo un análisis de sensibilidad al submuestreo, como se muestra en la Figura 2.4, donde se compara una serie temporal de alta resolución (línea gris continua) con las series submuestreadas. Se estableció un intervalo de 1 minuto como límite superior para el submuestreo, de manera de preservar elementos clave como el tiempo de arribo y los máximos secundarios. Esta elección permite reducir el número total de muestras a 1/12 del tamaño original. El efecto combinado del recorte y el submuestreo permitió reducir la longitud de las series de 1441 a 121 puntos para Valparaíso-Viña del Mar, y de 2881 a 241 puntos para Coquimbo-La Serena.

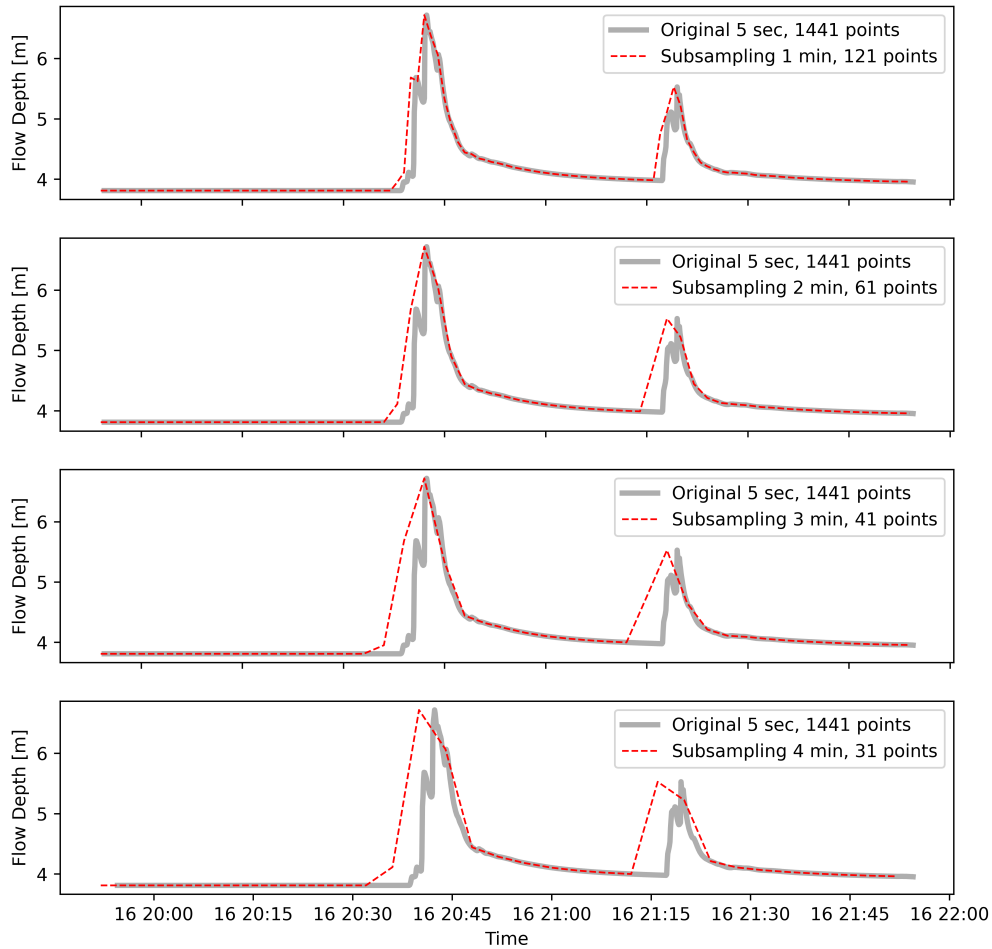


Figura 2.4: Series de tiempo de inundación de la boya ViB (Viña del Mar), asociada al escenario 9725 (Mw 9.2). Las líneas gruesas representan los registros cada 10 segundos en la simulación numérica, mientras que las líneas segmentadas corresponden a las series subsampladas. El subsampleo se realizó de manera conservadora, manteniendo el valor máximo de la serie en el intervalo de tiempo.

2.4. Modelos de *Machine Learning*

2.4.1. Aspectos Generales de Redes Neuronales Convolucionales 1D

En esta investigación se implementaron tres modelos de redes neuronales que utilizan como base capas convolucionales 1D (CNN-1D), debido a su buen desempeño en problemas de análisis de señales, incluso cuando se cuenta con un conjunto de datos limitado. Un modelo 1D-CNN es capaz de aprender patrones dentro de una secuencia de entrada \mathbf{x}_n y relacionarlos con los distintos elementos de una secuencia de salida \mathbf{y}_n . El objetivo es que el modelo, una vez calibrado, sea capaz de recrear la serie de inundación a partir de secuencias de entrada obtenidas de los registros de desplazamientos del suelo mediante estaciones GNSS, con tiempos de registro inferiores a nueve minutos. Para lograr esto, la red debe ajustar todos los parámetros o pesos que la componen. La cantidad de pesos depende de la arquitectura previamente definida, la cual está determinada por diversos hiperparámetros.

Una arquitectura 1D-CNN está compuesta por varias capas convolucionales, las cuales, a su vez, se componen de \mathbf{n} filtros. Cada filtro o kernel se especializa en aprender una característica específica de la entrada. Un kernel es un vector de pesos $\mathbf{k} : \{k_1, k_2, \dots, k_\beta\} \in \mathbb{R}^\beta$ de longitud $\beta \in \mathbb{N}$, que se desliza y multiplica la(s) secuencia(s) de entrada $\mathbf{x} : \{x_1, x_2, \dots, x_\alpha\} \in \mathbb{R}^\alpha$ de largo $\alpha \in \mathbb{N}$. El producto elemento a elemento del kernel con una porción de secuencia del mismo tamaño se conoce como característica (*feature*) (Fig. 2.5). A través de la operación convolución, el kernel se desliza a lo largo de toda la secuencia, lo que permite detectar patrones dentro de ella y generar un mapa de características como resultado. Formalmente, una característica se calcula como:

$$s_n = (\mathbf{x} * \mathbf{k})(n) = \sum_{\ell=1}^n \left(\sum_{p=-\infty}^{\infty} x^{(\ell)}[p] \cdot k^{(\ell)}[p - n + 1] \right), \quad (2.1)$$

donde $n \in \{1, \dots, \alpha\}$. En este caso, se utilizó padding = “same”, el cual consiste en rellenar con ceros los extremos de la secuencia de entrada antes de aplicar el filtro, de modo que la salida tenga la misma longitud que la entrada.

Los atributos s_n aprendidos en la capa convolucional se transfieren a la siguiente capa convolucional mediante una función activación, como se detalla a continuación:

$$y(n) = \sigma(b + (\mathbf{x} * \mathbf{k})(n)), \quad (2.2)$$

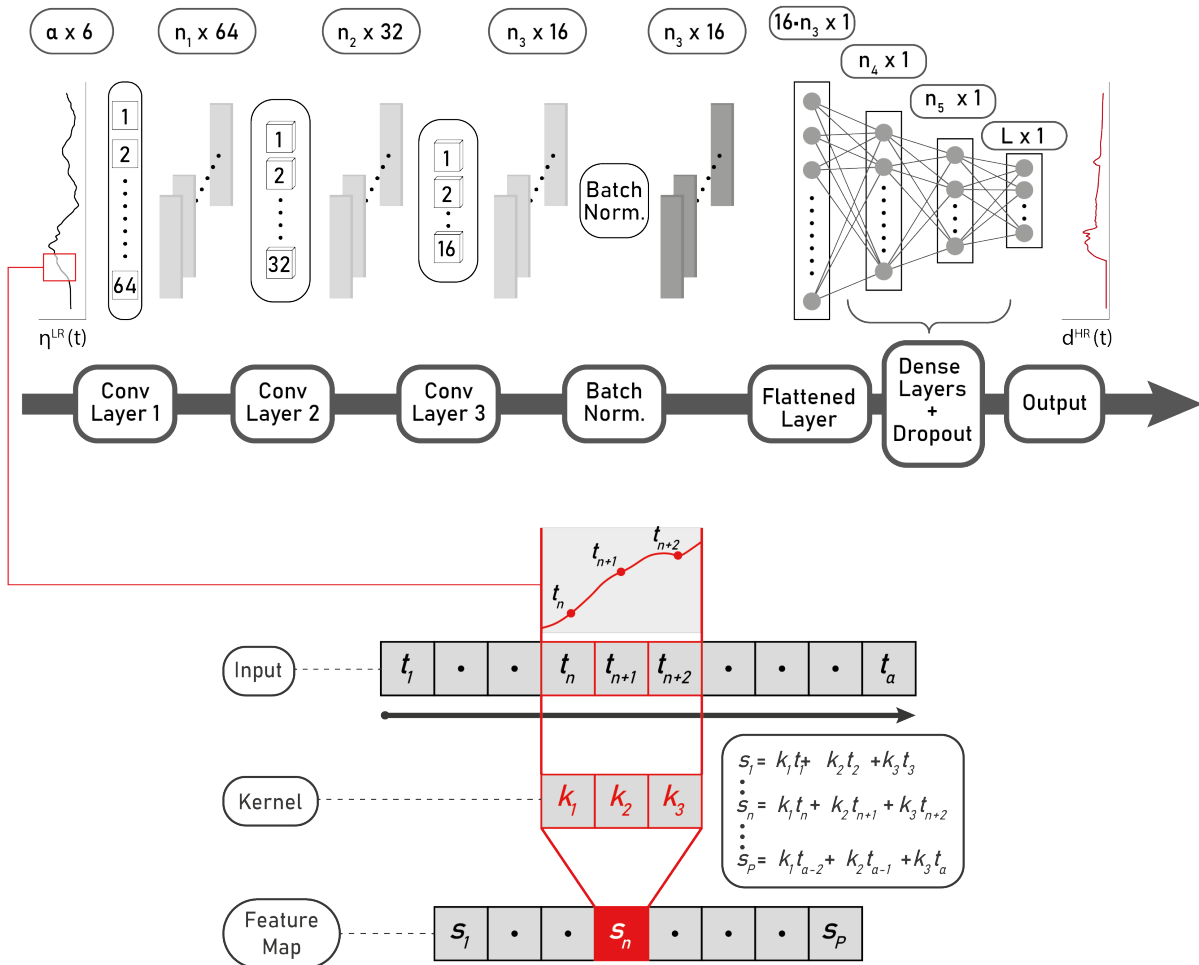


Figura 2.5: En la parte superior de la imagen se muestra la arquitectura de una red neuronal convolucional. En la parte inferior, se presenta un esquema de una operación convolucional unidimensional, donde un kernel (\mathbf{k}) de longitud tres opera sobre una secuencia (\mathbf{x}) de longitud α .

donde $b \in \mathbb{R}$ corresponde al intercepto (parámetro entrenable) y $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ es la función de activación (función no lineal, definida como un hiperparámetro).

La función de activación ReLU es una de las más utilizadas en redes profundas, ya que resuelve el problema de saturación presente en funciones del tipo sigmoide logística y tangente hiperbólica (tanh), lo que conlleva a sufrir el problema del gradiente desvaneciente (*Nair and Hinton, 2010*). Sin embargo, un inconveniente de las funciones ReLU es el fenómeno conocido como “neuronas muertas” o inactivas para entradas negativas, lo que impide el flujo de información de dichas entradas. Para abordar este problema, surge una variante: la función lineal rectificada con fuga, conocida como “LeakyReLU” (*Maas et al., 2013*), en la cual los valores negativos son multiplicados por un parámetro α , que es un hiperparámetro del modelo. Una propuesta interesante para resolver el problema de encontrar el valor más adecuado de α en LeakyReLU es la función parametric ReLU, en la que el parámetro α de cada neurona es ajustable durante el entrenamiento (*He et al., 2015*). Sin embargo, esta flexibilidad puede llevar fácilmente al sobreajuste del modelo.

Después de cada capa convolucional, se añade una capa de normalización por lotes (Batch Normalization), la cual normaliza, escala y desplaza la salida de la capa anterior mediante sus parámetros γ y β (ver Algoritmo 1). El uso de esta técnica se ha convertido en una práctica habitual en arquitecturas de redes neuronales profundas, ya que estabiliza el aprendizaje de la red y permite utilizar tasas de aprendizaje más altas, lo que se traduce en un entrenamiento más rápido y eficaz (*Ioffe and Szegedy, 2015*). Además, se incorpora una capa de MaxPooling después de cada dos capas convolucionales, con el fin de reducir la dimensionalidad y extraer las características más relevantes.

Algorithm 1 Transformación de Normalización por Lotes (BatchNorm)

INPUT: Mini-batch $\mathcal{B} = \{x_1, x_2, \dots, x_m\}$

Parámetros aprendibles: γ, β

OUTPUT: $\{y_i = \mathbf{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ media del mini-batch}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ varianza del mini-batch}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalizar}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \mathbf{BN}_{\gamma, \beta}(x_i) \quad // \text{ escalar y desplazar}$$

La parte final de la arquitectura de red definida para esta investigación corresponde a capas densas o *fully-connected*. Estas capas son responsables de relacionar todos los atributos apren-

dados en las capas convolucionales con la capa de salida. Donde cada neurona combina de forma no lineal todos los atributos generados en el nivel anterior $\mathbf{a}^{(\ell)} : \{a_1^{(\ell)}, a_2^{(\ell)}, \dots, a_I^{(\ell)}\} \rightarrow \mathbb{R}^I$, mediante un vector de pesos $\mathbf{w}_s^{(\ell)} : \{w_{1s}^{(\ell)}, w_{2s}^{(\ell)}, \dots, w_{Is}^{(\ell)}\} \rightarrow \mathbb{R}^I$ y luego transforma la señal total calculada usando una función de activación (Fig. 2.6).

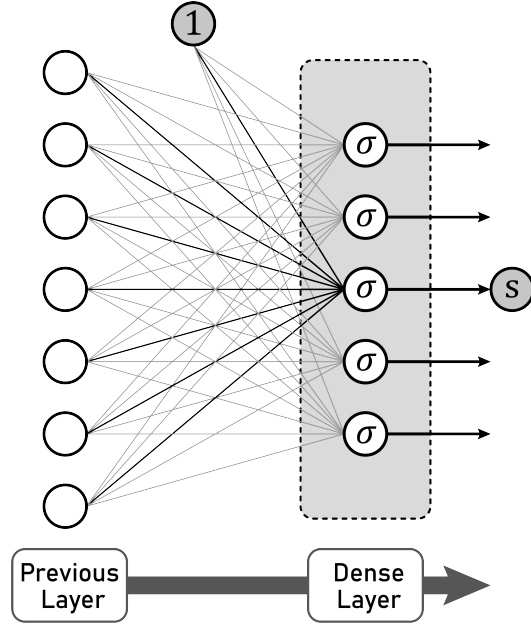


Figura 2.6: Esquema de las conexiones en una capa completamente conectada (capa densa). Se observa que cada neurona de la capa $\ell + 1$ está conectada a todas las neuronas de la capa anterior ℓ , cada conexión con su propio peso asociado

Donde el valor de un atributo $a_s^{\ell+1}$, asociado a la salida de la capa densa se puede calcular según la siguiente expresión:

$$\begin{aligned} a_s^{(\ell+1)} &= \sigma \left(\mathbf{w}_s^{(\ell)T} \mathbf{a}^{(\ell)} + b_s^{(\ell)} \right) \\ &= \sigma \left(\sum_{i=1}^I w_{si}^{(\ell)} a_i^{(\ell)} + w_{s0}^{(\ell)} \right), \end{aligned} \quad (2.3)$$

donde $b_s^{(\ell)} \in \mathbb{R}$ corresponde al intercepto (parámetro entrenable) y $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ es la función de activación (función no lineal habitualmente, definido como un hiper-parámetro no entrenable). En este caso se utilizó una función lineal rectificadora ($\text{relu}(x) = \max(x, 0)$), complementado con el uso de Dropout ([Srivastava et al., 2014](#)). Dropout es un método de regularización durante el entrenamiento, el cual “apaga” una tasa $p \in [0, 1]$ de neuronas en la capa de forma aleatoria. Esto evita que cada neurona se adapte demasiado a los datos de

entrenamiento, ganando con ello una mayor capacidad de generalización. Lo anterior implica un mejor desempeño, al momento de predecir ejemplos que no fueron utilizados en la fase de entrenamiento. En este caso, usamos $p = 0.1(10\%)$.

Siguiendo una línea poco tradicional en modelos de regresión, se optó por utilizar una función lineal rectificadora ($\text{relu}(x) = \max(x, 0)$) en la capa de salida. La función relu solo permite valores de salida mayor o igual a cero. Esto es deseado, dado la naturaleza del problema, recordando que el objetivo principal es recrear series de inundación por tsunamis. Esta función permite al modelo entregar valores físicamente correctos y a la vez ayuda a predecir de manera precisa los casos que no inundan, cuya respuesta esperada es una serie nula.

2.4.2. Arquitecturas Convolucionales para Modelado de Inundación

A continuación se describen los modelos propuestos CNN-GI; CNN-GP y CNN-PI los cuales comparten una arquitectura base convolucional similar, compuesta por cuatro bloques de convolución 1D con un número decreciente de filtros (128, 64, 32 y 16). En esta investigación, se definen los “**bloques de convolución**” como un conjunto compuesto por dos capas convolucionales 1D con “ n ” filtros y un tamaño de kernel “ k ”, utilizando la función de activación LeakyReLU. Después de cada capa convolucional, se agrega una capa de *Batch Normalization* (Ioffe and Szegedy, 2015) y, al final del bloque, una capa de *MaxPooling1D* (ver Figura 2.7).

Luego de estos bloques, la salida se transforma en un vector unidimensional mediante una capa *flatten*, y pasa a través de dos capas densas, cuyo número de neuronas en cada capa corresponde a un hiperparámetro a definir. Para mejorar la capacidad de generalización, se emplea *Dropout* en cada capa densa.

La diferencia fundamental entre los modelos radica en las variables de entrada y salida, así como en la función de activación de salida y función de pérdida.

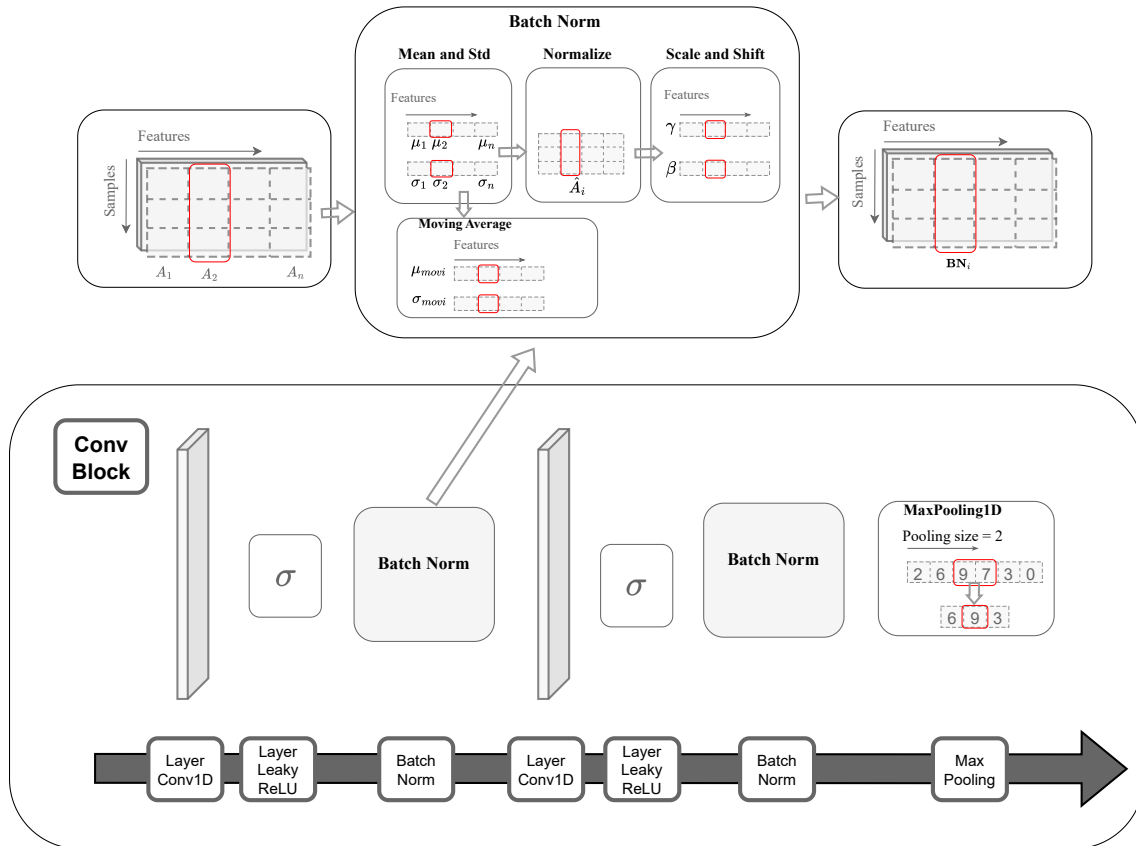


Figura 2.7: Esquema del bloque convolucional utilizado en los modelos propuestos. Cada bloque está compuesto por dos capas convolucionales 1D con activación LeakyReLU, seguidas por capas de Batch Normalization. Al final del bloque, se incluye una capa de MaxPooling1D. El número de filtros y el tamaño del kernel se definen como hiperparámetros.

2.4.2.1. Modelo Directo GNSS–Inundación (CNN-GI)

Este modelo recibe señales GNSS como entrada y genera directamente las series de inundación. La salida final se calcula mediante una capa densa con activación ReLU. El modelo se entrena minimizando la pérdida MSE (error cuadrático medio), optimizado con Adam.

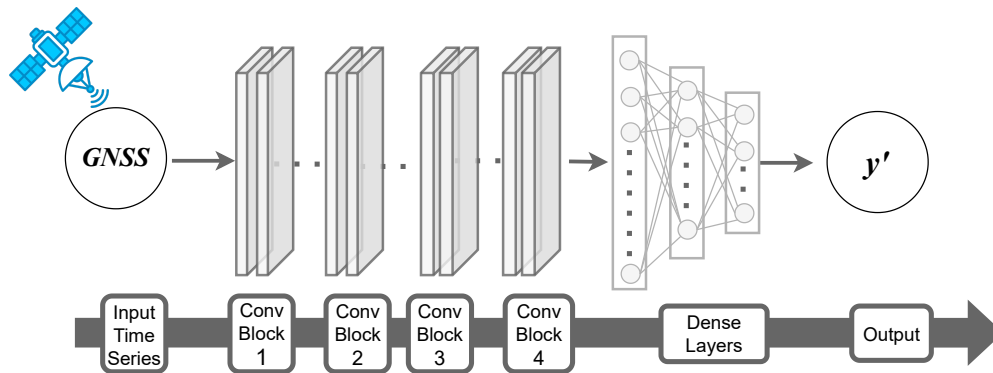


Figura 2.8: Esquema de la arquitectura CNN-GI. El modelo recibe como entrada una serie temporal de señales GNSS y, a través de una red convolucional 1D, calcula directamente la serie de inundación en tierra. Esta arquitectura representa un enfoque directo.

2.4.2.2. Modelo de Dos Etapas GNSS–Propagación–Inundación (CNN-GPI)

Basado en los trabajos previos realizados por *Núñez et al. (2022)* y *Rim et al. (2022)* este enfoque se divide en dos etapas:

- **Primera etapa:** Es un modelo multi objetivo que recibe señales GNSS y predice seis series de elevación instantánea de la superficie libre (η). Cada una de las salidas es una capa densa con activación lineal y se utiliza MAE (error absoluto medio) como función de pérdida.
- **Segunda etapa:** Toma como entrada las series (η) resultantes del modelo anterior y predice la serie de inundación. La salida final utiliza una activación ReLU, y se minimiza la pérdida MSE durante el entrenamiento.

Ambas etapas comparten la arquitectura convolucional base descrita anteriormente. Esta estrategia busca capturar de forma intermedia el efecto de la propagación del tsunami antes de predecir la inundación final, permitiendo así representar de forma más completa y realista la secuencia física del fenómeno, desde su generación y propagación hasta la inundación.

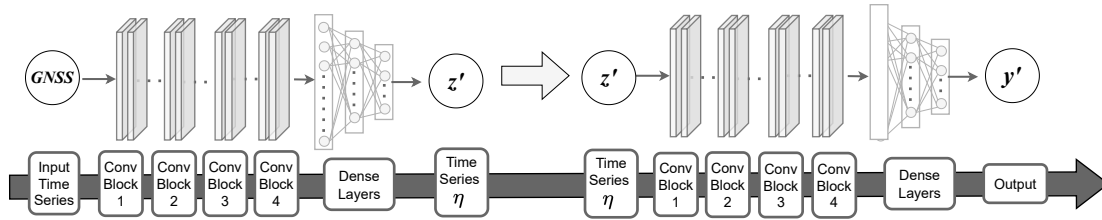


Figura 2.9: Esquema de la arquitectura CNN-GPI con dos etapas. En una primera etapa, una red convolucional estima la elevación instantánea de la superficie libre (η) a partir de las señales GNSS. Luego, una segunda red convolucional utiliza esa salida intermedia para calcular la serie de inundación en tierra. Este enfoque busca descomponer el problema en dos subprocesos relacionados, incorporando explícitamente la propagación del tsunami como variable intermedia.

2.4.3. Red Neuronal Convolucional Ramificada (BCNN)

Los modelos BCNN (*Zhu and Bain, 2017*) son arquitecturas de redes neuronales convolucionales que se caracterizan por tener “ramas” separadas que divergen desde una parte compartida de la red. Estas ramas permiten que la red procese distintas características o aspectos de los datos de manera más específica y, al final, produzca múltiples salidas a partir de las entradas compartidas. En el contexto de predicción y análisis, cada rama puede estar especializada para resolver un subproblema particular, facilitando una mejora en el aprendizaje de patrones complejos que influyen en el objetivo principal.

Comparación con una 1D-CNN clásica:

- Modelado de características específicas: Una Branch-CNN puede aprender características o relaciones específicas que son más difíciles de identificar o tratar con una red CNN clásica, que tiene una única ruta de aprendizaje. En este caso, una rama puede aprender patrones específicos de la elevación instantánea de la superficie libre (η), mientras que otra predice la inundación.
- Compartición y especialización: La parte inicial compartida de la Branch-CNN permite extraer características generales de la señal, y luego cada rama especializa la información, lo que puede llevar a una mejor representación y, potencialmente, a un mejor rendimiento en problemas multisalida.
- Descomposición de problemas complejos: En casos donde existe una relación causal o influyente entre las salidas (como el impacto de la elevación instantánea de la superficie

libre (η) en las inundaciones), una Branch-CNN puede modelar explícitamente estas interacciones.

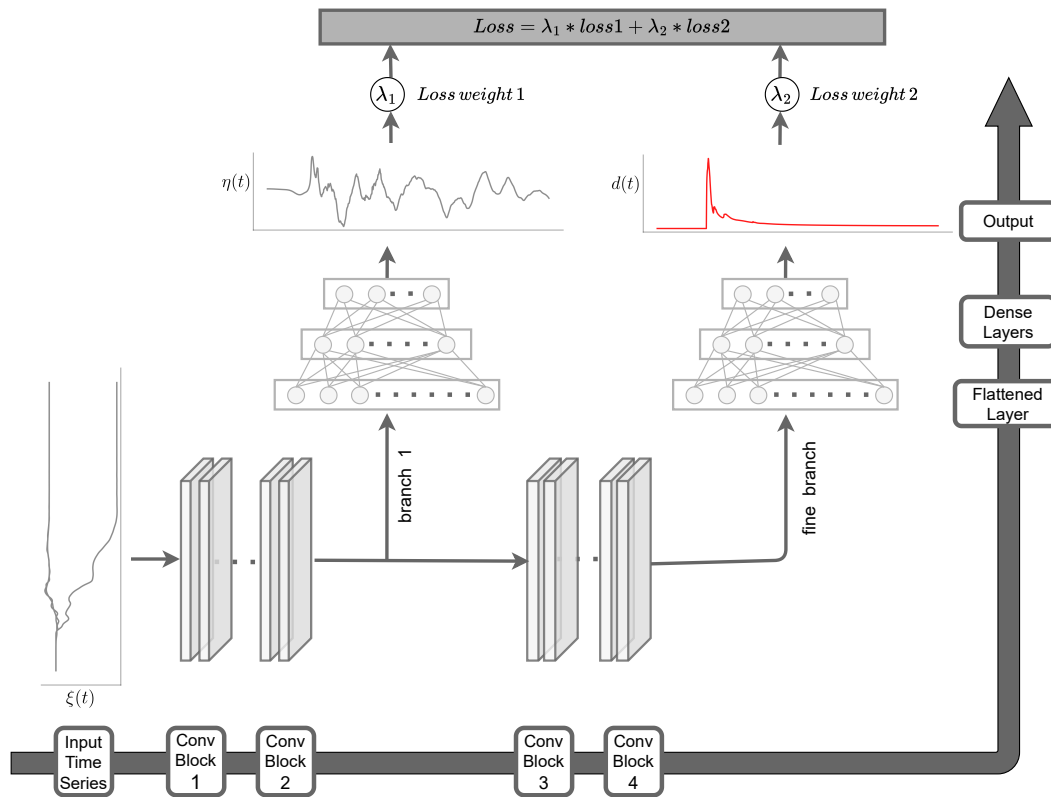


Figura 2.10: Esquema de la arquitectura de la red neuronal Branch-CNN con dos ramas. La red recibe señales GNSS como entrada y, tras una extracción inicial de características, se bifurca en dos ramas: una para predecir la propagación del tsunami en el mar y otra para estimar la inundación en tierra.

Descripción del modelo

El modelo Branch-CNN utilizado en este trabajo está diseñado para procesar señales GNSS y predecir tanto la variación de la superficie del mar como las inundaciones. La arquitectura se estructura en una parte compartida inicial y dos ramas especializadas, como se ilustra en la Figura 2.10.

Tras el segundo bloque, la red se bifurca en dos ramas:

- Branch 1: Las características extraídas en los primeros bloques convolucionales se trans-

forman en un vector unidimensional (capa *flattened*), luego las representaciones pasan por dos capas densas, cuyo número de neuronas es un hiperparámetro a definir. Estas capas, regularizadas con *BatchNormalization* y Dropout, conectan las características globales con la predicción de la serie de elevación instantánea de la superficie libre (η). La salida de esta rama se obtiene mediante una capa densa con activación lineal.

- Fine branch: La segunda rama continúa procesando las características mediante dos bloques de convolución adicionales, que emplean 32 y 16 filtros, respectivamente. Las características extraídas en los bloques convolucionales se transforman en un vector unidimensional (capa *flattened*), luego las características pasan por dos capas densas, también regularizadas con *BatchNormalization* y Dropout. La predicción final, correspondiente a la serie de inundación, se realiza mediante una capa densa con activación ReLU.

El modelo se compila con el optimizador Adam, utilizando pérdidas MAE y MSE para las salidas de la variación de la superficie del mar y la inundación, respectivamente. Ambos términos de pérdida tienen igual peso, permitiendo al modelo ajustar sus parámetros de manera equilibrada para minimizar los errores en ambas predicciones.

2.4.4. Branch CNN con Bt-Strategy (BCNN-Bt)

Este modelo BCNN utiliza la arquitectura descrita previamente, que toma como entrada señales GNSS y tiene dos ramas de salida: una para predecir la serie de elevación instantánea de la superficie libre (η) y otra para predecir la inundación. Se aplica la estrategia *BT-Strategy* ([Zhu and Bain, 2017](#)) para modificar los pesos de las funciones de pérdida asociadas a cada una de las ramas, lo que permite controlar la importancia relativa de cada predicción durante el entrenamiento.

Pérdidas y Pesos de BT-Strategy:

Se definen dos variables de pesos λ_1 y λ_2 , asociadas a las funciones de pérdida de cada rama: la rama *branch 1* para la predicción de la elevación instantánea de la superficie libre (η) y la rama *fine branch* para la predicción de la serie de inundación. La pérdida total del modelo se calcula como una combinación lineal de las pérdidas individuales de cada rama:

$$\mathcal{L}_{total} = \lambda_1 \cdot \mathcal{L}_1 + \lambda_2 \cdot \mathcal{L}_2.$$

Inicialmente, $\lambda_1 = 1.0$ y $\lambda_2 = 0.0$, lo que implica que durante las primeras épocas del entrenamiento, solo se considera la pérdida asociada a la serie η . Luego, a partir de una época específica, los valores de λ_1 y λ_2 se actualizan dinámicamente utilizando un callback personalizado llamado `LossWeightsModifier`. Este callback utiliza listas predefinidas `lambda1_values` y `lambda2_values` para asignar nuevos pesos en función de la época, lo que permite al modelo cambiar gradualmente su enfoque desde la primera salida (η) hacia la segunda (inundación). Los pesos λ_1 y λ_2 y la época donde se actualizan son hiperparámetros a definir.

Ventaja de la BT-Strategy

La modificación dinámica de los pesos mediante la BT-Strategy permite controlar progresivamente la atención del modelo en cada salida, facilitando un aprendizaje más eficiente y específico. Este enfoque resulta especialmente útil en problemas donde una salida auxiliar (rama secundaria) en este caso con dependencias físicas, pueda guiar al modelo en la identificación de patrones relevantes antes de generar la salida principal. En este caso, se busca que el modelo aprenda características de la elevación instantánea de la superficie libre del mar que influyan posteriormente en la predicción de inundaciones, maximizando así el aprendizaje conjunto.

2.5. Selección de Hiperparámetros

La selección de los hiperparámetros para cada uno de los modelos descritos anteriormente (CNN-GI; CNN-GPI; BCNN; BCNN-Bt) se realizó mediante una búsqueda en grilla (*grid search*). Este procedimiento itera sobre el dominio de cinco hiperparámetros, los cuales se presentan en la Tabla 2.2.

Para reducir el número de combinaciones probadas, se mantuvieron fijos algunos hiperparámetros, como se detalla en la Tabla 2.2. Estos fueron determinados a partir de búsquedas previas, lo que permitió realizar una búsqueda más exhaustiva de los hiperparámetros más relevantes para nuestra tarea. Además, mediante el uso de un conjunto reducido de datos y la implementación de *early stopping* con una paciencia de 20 ([Caruana et al., 2001](#)), se logró disminuir el tiempo requerido por el algoritmo de búsqueda.

Tabla 2.2: Conjunto de hiperparámetros probados.

Hyperparameters	
Variable	Range of values
Kernel size β	[3; 16; 32]
Dropout $p\%$	[10; 30; 50]
LeakyReLU α	[0; 0.01; 0.1; 0.3; 0.5; 0.7; 0.9]
Neurons D1	[361; 721; 1441]
Neurons D2	[361; 721; 1441]
Constant	
Dense layers	2
Convolutional layers	8
Patience	20
layers activation function	LeakyReLU
Filters	128-64-32-16
Buoy Depth (m)	50
Batch size	32
Learning rate	10^{-4}
Output layer activation function	ReLU

De las 567 combinaciones evaluadas, se seleccionaron las cinco configuraciones con el menor error en el conjunto de validación para cada modelo. Estas combinaciones, presentadas en la Tabla 2.3, fueron evaluadas utilizando la métrica de desempeño *Mean Squared Error (MSE)*:

$$MSE_j = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^L (F_{j,i}(X_i)_k - Y_{i,k})^2, \quad (2.4)$$

donde F_j representa el modelo que se está ajustando, X_i corresponde a las secuencias de entrada GNSS, Y_i a la series de tiempo de inundación, y ambos son elementos del conjunto de validación de tamaño N . Dado que la diferencia entre las mejores combinaciones de hiperparámetros es baja, se selecciona la configuración con los valores de hiperparámetros más recurrentes, los cuales se destacan en negrita para cada modelo en la Tabla 2.3.

Tabla 2.3: Ranking de las mejores combinaciones de hiperparámetros en validación.

Network Model	Hyperparameters					MSE Val. Eq.(2.4)	
	Neurons Layer 1	Neurons Layer 2	LeakyReLU α	Kern. size	Dropout $p\%$	Inund. [m^2]	No Inund. [m^2]
CNN-GI	721	361	0.7	32	10	0.124109	0.002406
	721	721	0.7	32	10	0.125363	0.002336
	721	721	0.5	32	10	0.125856	0.003104
	721	721	0.5	16	10	0.126064	0.003291
	1441	721	0.7	32	10	0.127869	0.002488
CNN-GP	721	721	0.7	32	50	0.583310	—
	721	721	0.7	32	10	0.585744	—
	721	721	0.7	3.0	50	0.646714	—
	721	721	0.3	32	10	0.657620	—
	721	721	0.7	3.0	10	0.661809	—
CNN-PI	721	361	0.5	32	10	0.084968	0.001387
	721	721	0.5	32	10	0.085428	0.001343
	721	721	0.5	16	10	0.085894	0.001448
	721	361	0.5	16	10	0.086336	0.001160
	721	361	0.3	16	10	0.087423	0.001341
BCNN	721	721	0.3	16	30	0.121722	0.002055
	721	361	0.7	16	10	0.121917	0.001510
	721	361	0.7	32	10	0.121988	0.001679
	721	721	0.7	32	10	0.122124	0.001921
	721	721	0.7	32	30	0.122973	0.001382

Tabla 2.4: Conjunto de hiperparámetros utilizados en *Bt-Strategy*.

Hyperparameters	
Variable	Range of values
Loss weight λ_1	[0.1; 0.5; 0.9]
Loss weight λ_2	[0.1; 0.5; 0.9]
Modifier LW	[50; 100; 300]

Una vez definidos los hiperparámetros de la arquitectura **BCNN**, se realizó una exploración adicional para determinar los valores óptimos de λ_1 , λ_2 y el parámetro **Modifier LW** (número de la época que se actualizan los pesos de las pérdidas λ_1 y λ_2) en la arquitectura **BCNN-Bt** (ver Tabla 2.4). Estos resultados se presentan en la Tabla 2.5. Los resultados muestran que el mejor desempeño se obtiene cuando el entrenamiento considera únicamente la primera rama ($\lambda_1 = 1.0$, $\lambda_2 = 0.0$) hasta la época 300, y luego se aplica una ponderación de pérdidas igual a 0.5 para ambas pérdidas. Esta configuración se destaca en **negrita** en la Tabla 2.5.

Tabla 2.5: Las combinaciones de hiperparámetros con mejor desempeño para BCNN con la estrategia Bt-Strategy.

Network Model	Hyperparameters			MSE Val. Eq.(2.4)	
	Loss Weights λ_1	Loss Weights λ_2	Modifier LW <i>epoch</i>	Inund. [m^2]	No Inund. [m^2]
BCNN-BT	0.5	0.5	300	0.116795	0.001467
	0.9	0.1	100	0.117088	0.001165
	0.5	0.5	100	0.117597	0.001605
	0.1	0.9	100	0.118152	0.001616
	0.1	0.9	50	0.11839	0.002423
	0.1	0.9	300	0.119010	0.001296
	0.5	0.5	50	0.119216	0.001794
	0.9	0.1	50	0.121184	0.001574
	0.9	0.1	300	0.122525	0.000899

2.6. Entrenamiento, validación y test

Debido a la mínima diferencia observada entre las mejores combinaciones en validación, se seleccionó un modelo para cada metodología, compuesto por los hiperparámetros más frecuentes destacados en **negrita** en las Tablas 2.3 y 2.5. Los modelos seleccionados se sometieron

a un nuevo entrenamiento con un mayor número de escenarios y un aumento en la cantidad de épocas.

El ajuste de los pesos de los modelos se realizó utilizando el optimizador Adam, con una tasa de aprendizaje “ η ” igual a 10^{-4} . Este optimizador es un método de gradiente descendente estocástico que emplea el cálculo de los momentos de primer y segundo orden (media y varianza) de los gradientes (*Kingma and Ba, 2015*) para reducir el error de la función de pérdida. Su funcionamiento se detalla en el Algoritmo 2.

Adicionalmente, se utilizó *early stopping* con una paciencia igual a 50 (*Caruana et al., 2001*) para evitar el sobreajuste del modelo, lo cual es deseable ya que favorece la capacidad de generalización, es decir, un mejor desempeño en el pronóstico de escenarios que no fueron usados durante la etapa de entrenamiento.

Algorithm 2 Actualización de pesos con Adam

Require:

- 1: W_0
 - 2: $m_0 \leftarrow 0$
 - 3: $v_0 \leftarrow 0$
 - 4: $t \leftarrow 0$
 - 5: **while** W_t not converged **do**
 - 6: Random batch n examples
 - 7: $t \leftarrow t + 1$
 - 8: $g_t \leftarrow \frac{1}{n} \sum_j \nabla \ell(f(\mathbf{x}_j, y_j))$
 - 9: $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
 - 10: $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t \odot g_t$
 - 11: $\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t}$
 - 12: $\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t}$
 - 13: $W_t \leftarrow W_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$
 - 14: **end while**
 - 15: **return** W_t
-

Para mitigar sesgos asociados a la partición de los conjuntos de entrenamiento, validación y prueba, se generaron 10 particiones distintas, cada una asociada a una semilla que garantiza su reproducibilidad. Los resultados obtenidos por los 10 modelos entrenados en su respectivo conjunto de prueba fueron promediados y se presentan en la Tabla 3.1

Dado que las series de inundación presentan un gran número de valores iguales a cero, lo cuales podrían enmascarar el desempeño reflejado en la métrica MSE, los nuevos modelos entrenados se evaluaron con métricas adicionales. Esto permite seleccionar un modelo de manera más conservadora. Similar al trabajo de *Núñez et al. (2022)*, utilizamos la métrica G

(ver Ecuación 2.5), que representa un error cuadrático normalizado:

$$G_j = \frac{1}{N_S} \sum_{i=1}^{N_S} \left[1 - 2 \frac{\sum_{k=1}^L F_{j,i}(\eta_\ell)_k * Y_{i,k}}{\sum_{k=1}^L F_{j,i}^2(\eta_\ell)_k + \sum_{k=1}^L Y_{i,k}^2} \right]. \quad (2.5)$$

Esta métrica evalúa la similitud entre $F_{j,i}$ y las predicciones reales Y_i , normalizada por la suma cuadrada de ambas señales. Esto permite comparar el error de señales con diferentes magnitudes. Un valor bajo de G indica una alta similitud entre la predicción y el valor real, mientras que un valor más alto refleja una mayor discrepancia ([Davies, 2019](#)).

Además, se calculó la diferencia entre los valores máximos y los tiempos de arribo de forma similar al trabajo de [Núñez et al. \(2022\)](#). El error en la profundidad máxima se mide como la diferencia entre el máximo de la serie real $F_{j,i}$ y el valor de la predicción en el mismo instante en que se produce dicho máximo:

$$E_{d,i} = F(t = t_{max\{Y_i\}}) - max\{Y_i\}, \quad (2.6)$$

y la diferencia en el tiempo de arribo t^a , se calcula como:

$$E_{t,i} = t_{F_i}^a - t_{Y_i}^a. \quad (2.7)$$

Esta información es especialmente relevante en el contexto de un sistema de alerta temprana, ya que permite clasificar rápidamente la amenaza según los criterios de peligro establecidos en [Zamora et al. \(2021\)](#).

En este mismo contexto, se calculó el número de alertas falsas y no emitidas para cada modelo, clasificadas en tres categorías. La categoría A corresponde a eventos con una profundidad de flujo máxima inferior a 0.3 metros; la categoría B incluye eventos con profundidades de flujo entre 0.3 y 1.0 metros; y la categoría C abarca eventos de alta peligrosidad, con profundidades de flujo superiores a 1.0 metros (ver Tabla 3.1).

Capítulo 3

RESULTADOS Y ANÁLISIS

En este capítulo se presentan los resultados obtenidos en el conjunto de prueba para los cuatro modelos de redes neuronales diseñados para resolver el problema de cálculo de series de inundación a partir de registros de señales GNSS. La Tabla 3.1 muestra las métricas de desempeño obtenidas por los distintos modelos, clasificadas por zonas de estudio. El modelo con el mejor desempeño global se destaca en negrita.

Tabla 3.1: Métricas de desempeño del mejor modelo de red neuronal (NNM) en cada ubicación. Los datos resaltados en negrita corresponden al NNM final para cada boya objetivo.

Network Model	<i>MSE</i> Test Eq. 2.4		<i>G</i> Test Eq. 2.5		Test		False Alarm			Missed Alarm		
	Inund. [m^2]	No Inund. [m^2]	Inund. [-]	No Inund. [-]	$E_{d,i}$ [m]	$E_{t,i}$ [min]	A	B	C	A	B	C
CoB-CNN-GI	0.1264	0.0002	0.3262	0.0661	1.12	9.84	59	13	1	8	4	0
CoB-CNN-GPI	0.1133	0.0002	0.3156	0.0432	1.04	8.8	36	10	0	10	6	1
CoB-BCNN	0.1368	0.0004	0.3487	0.2048	1.19	10.22	195	27	1	7	4	1
CoB-BCNN-Bt	0.1237	0.0003	0.3269	0.1191	1.09	10.21	107	22	1	5	3	0
ViB-CNN-GI	0.1278	0.0027	0.4120	0.1865	1.28	4.50	90	95	17	5	5	6
ViB-CNN-GPI	0.1109	0.0019	0.383	0.1562	1.22	5.19	104	54	11	5	4	4
ViB-BCNN	0.1280	0.0018	0.4473	0.2089	1.27	4.97	141	74	12	4	6	5
ViB-BCNN-Bt	0.1225	0.0024	0.3806	0.2166	1.26	5.33	161	61	14	4	4	4

Para la zona de Coquimbo (CoB), el modelo **CNN-GPI** obtuvo el mejor desempeño en todas las métricas. En contraste, en la zona de Viña del Mar (ViB) no se observó un modelo claramente superior. Dado que en algunas métricas las diferencias entre los modelos no fueron significativas, se realizó un análisis más detallado del error cuadrático medio (MSE) en los casos de inundación para determinar la mejor opción. Los resultados de este análisis, presentados en la Figura 3.1, muestran que, en ambas zonas, los modelos CNN-GPI presentan un menor error medio y una varianza reducida en comparación con los demás modelos. Además, los modelos ajustados en la zona CoB exhiben una varianza general más baja, lo que podría deberse a una mayor homogeneidad en su base de datos, equilibrando los ejemplos con y sin inundación. Esto proporciona un mayor número de muestras representativas durante el

entrenamiento, favoreciendo un ajuste más robusto del modelo.

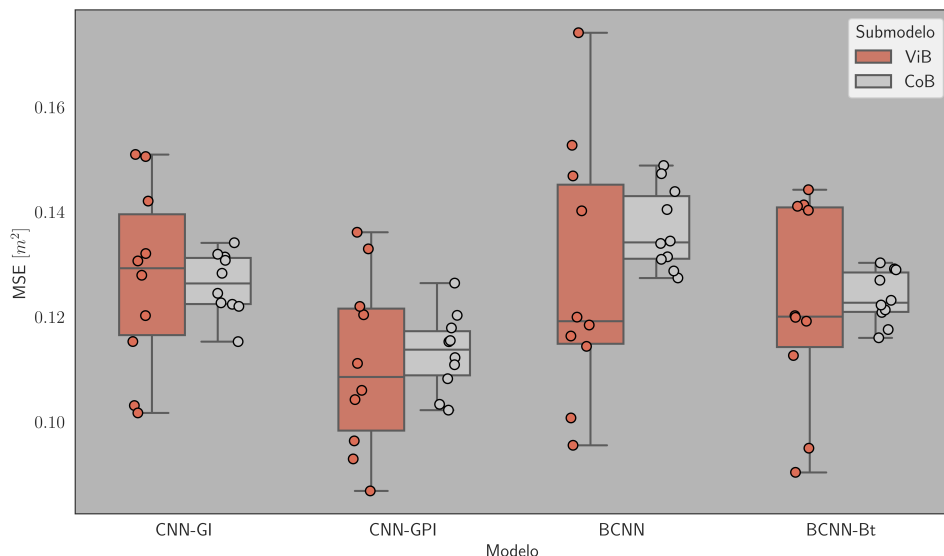


Figura 3.1: Distribución del MSE en los distintos conjuntos de prueba. Los puntos representan el error obtenido por cada modelo, asociado a cada semilla utilizada para la separación de los conjuntos durante el ajuste y la prueba.

Con base en las distintas métricas y análisis realizados, el modelo **CNN-GPI** demuestra ser la mejor opción tanto para Coquimbo (CoB) como para Viña del Mar (ViB). Esto refuerza la importancia de considerar información que refleje los efectos hidrodinámicos locales de cada bahía, los cuales estarían contenidos en las series de elevación instantánea de la superficie libre (η). Si bien los modelos **BCNN** también incorporan información de η durante la etapa de entrenamiento, lo hacen mediante una penalización “suave”, añadiendo un término adicional en la función de pérdida (*loss weighting*). Sin embargo, esta estrategia no considera un control específico, ya que durante todo el entrenamiento ambas pérdidas se ponderan por igual, favoreciendo la minimización de la pérdida global sin priorizar explícitamente el pronóstico de la inundación.

El modelo **BCNN-Bt**, en cambio, incorpora la estrategia **Bt-Strategy** para ajustar dinámicamente los pesos de las pérdidas durante el entrenamiento, priorizando distintos objetivos conforme este avanza. En las primeras épocas, el modelo se enfoca en aprender características y relaciones entre las señales GNSS y la propagación del tsunami (η) a través de sus bloques convolucionales primarios. Posteriormente, mediante un ajuste en pesos de las pérdidas (λ_1 y λ_2), el modelo se orienta a minimizar el error asociado a la inundación, aprovechando la información aprendida previamente, con el objetivo de capturar los efectos hidrodinámicos locales característicos de cada bahía. Aunque esta técnica mejoró el desempeño respecto al

modelo **BCNN**, aún obtuvo resultados inferiores al enfoque de dos etapas **CNN-GPI**.

Por su parte, el modelo **CNN-GPI** incorpora la información de elevación instantánea de la superficie libre (η) mediante una penalización “dura”, descomponiendo el problema en dos fases. En la primera etapa, el modelo se ve forzado a aprender la relación entre las señales GNSS y la serie η ; luego, estas salidas intermedias se utilizan como entrada de un segundo modelo que estima la inundación. Este enfoque permite resolver el problema siguiendo un flujo de información más realista, simulando la secuencia temporal natural: generación (GNSS), propagación (η) e inundación.

Al comparar los resultados de nuestro mejor modelo (**CNN-GPI**) con el desempeño del modelo presentado por [Núñez et al. \(2022\)](#), se observa que el nuestro es significativamente inferior en términos de precisión. En promedio, nuestro modelo presenta un error de aproximadamente 33 cm, en comparación con los 13 cm reportados por [Núñez et al. \(2022\)](#). Además, se registran diferencias en los valores máximos de las series de inundación del orden de [1.04 a 1.22] metros, mientras que en [Núñez et al. \(2022\)](#) estas diferencias se encuentran entre [0.22 a 0.35] metros. También se observa un mayor número de alertas falsas y no emitidas, incluso en eventos de categoría “C” (alta peligrosidad).

Cabe destacar que el modelo propuesto por [Núñez et al. \(2022\)](#) utiliza como entrada las series de elevación instantánea de la superficie libre (η), obtenidas a partir de modelaciones numéricas que asimilan mediciones reales de esta variable. En cambio, nuestro modelo **CNN-GPI** emplea señales GNSS como entrada para inferir primero las series η antes de estimar la inundación, introduciendo así un error adicional en la etapa intermedia del pronóstico.

Tabla 3.2: Métricas de desempeño en el conjunto de prueba

Class	ViB-CNN-GPI				CoB-CNN-GPI			
	Precision	Recall	F1-score	Support	Precision	Recall	F1-score	Support
0	0.99	0.81	0.89	1084	0.91	0.90	0.90	442
A	0.06	0.32	0.1	34	0.32	0.41	0.36	95
B	0.13	0.38	0.19	39	0.31	0.41	0.35	111
C	0.81	0.59	0.68	126	0.96	0.87	0.91	635

Si bien la precisión obtenida por los modelos propuestos para el cálculo de series de inundación por tsunami a partir de registros GNSS presenta errores considerables en métricas como MSE, así como en la generación de alertas falsas y alertas no emitidas (en comparación con el desempeño reportado por ([Núñez et al., 2022](#))), evaluamos su aplicabilidad en un sistema

a)

		Predicted Class			
		0	A	B	C
True Class	0	877	153	49	5
	A	3	11	14	6
	B	3	15	15	6
	C	0	12	40	74

b)

		Predicted Class			
		0	A	B	C
True Class	0	397	37	8	0
	A	28	39	24	4
	B	9	35	45	22
	C	2	11	67	555

Figura 3.2: Matriz de confusión del modelo CNN-GPI evaluado sobre el conjunto de 1283 ejemplos de prueba, mostrando los resultados de clasificación para las cuatro clases definidas. a) Modelo ajustado en ViB. b) Modelo ajustado en CoB.

de alerta temprana cuantificando su desempeño como clasificador de niveles de riesgo. Para ello, se consideraron cuatro clases: “0” (sin inundación), “A” (máximo < 0.3 metros), “B” ($0.3 \leq \text{máximo} < 1.0$) y “C” (máximo ≥ 1.0 metros).

En la Figura 3.2 se muestra la matriz de confusión obtenida sobre el conjunto de prueba para uno de los modelos CNN-GPI ajustados. Además, se calcularon distintas métricas utilizadas comúnmente en clasificación, cuyos resultados se presentan en la Tabla 3.2. En esta se observa un excelente desempeño del modelo, con un 99 % de precisión en la detección de eventos sin inundación (clase “0”) y un 81 % de precisión para los eventos de alta peligrosidad (clase “C”). Como es esperable en problemas de esta naturaleza, el menor rendimiento se presenta en las clases cercanas a los umbrales de clasificación, correspondientes a inundaciones de baja profundidad. En estos casos, el modelo tiende a confundir los eventos, generando así alertas falsas o alertas no emitidas.

Los resultados obtenidos con la metodología en dos pasos muestran que cuando la predicción de la serie de elevación instantánea de la superficie libre (η) asociada a la propagación del tsunami (P) es precisa, el modelo de inundación también arroja buenos resultados. Este enfoque permite descomponer el problema y capturar de manera más efectiva la relación entre las señales GNSS y la inundación costera.

Sin embargo, se observan diferencias en el comportamiento de los modelos ajustados para distintas zonas. En particular, los modelos entrenados con datos de Viña del Mar tienden a generar pronósticos más conservadores (ver Figura 3.3), mientras que los modelos ajustados para la Bahía de Coquimbo muestran una tendencia a subestimar la inundación (ver Figura 3.4). Este comportamiento puede deberse a la presencia de efectos locales de resonancia en la Bahía de Coquimbo, que pueden amplificar la oscilación del nivel del mar, generando una respuesta no capturada completamente por el modelo.

Para evaluar la capacidad predictiva del modelo con datos reales, se realizó una prueba utilizando registros GNSS del terremoto de Illapel 2015 Mw 8.4. En Viña del Mar, el modelo logró reconstruir correctamente la forma y el promedio de la serie de variación del nivel del mar, prediciendo la no ocurrencia de inundación, lo que concuerda con la realidad observada. En contraste, en la Bahía de Coquimbo, el modelo subestimó significativamente la amplitud de la serie en mar, lo que resultó en una predicción erróneamente baja de la inundación en tierra. En este caso, el modelo estimó un valor máximo de 1.49 metros, mientras que los valores reales indican un máximo cercano a 5.86 metros, lo que sugiere que no está capturando completamente los efectos locales de resonancia que amplificaron la altura del tsunami en esta zona.

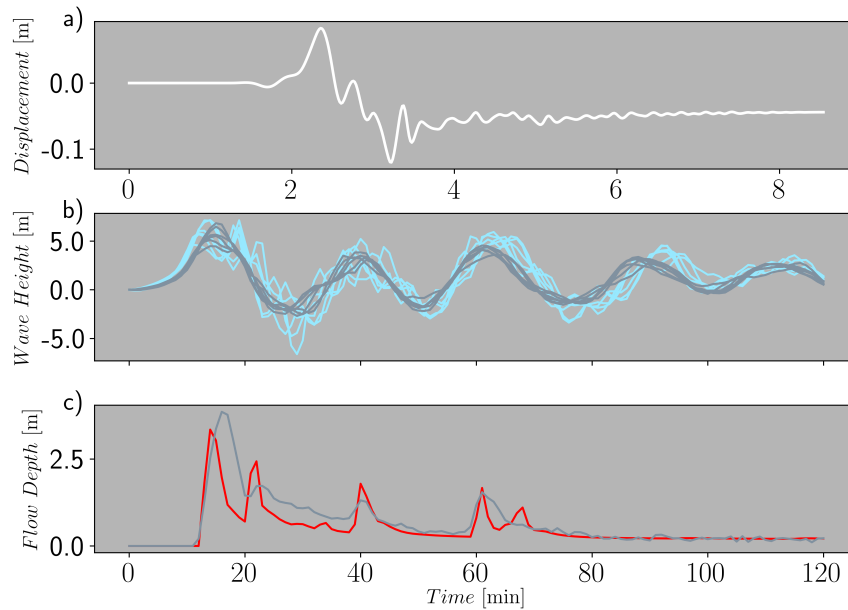


Figura 3.3: Predicción del modelo CNN-GPI para la zona ViB en el escenario # 9412 (Mw 8.9). a) Desplazamiento horizontal en la dirección Este. b) Variación del nivel del mar, donde las líneas magenta representan los valores reales y las grises, la predicción del modelo. c) Profundidad de inundación, con la línea roja indicando los valores reales y la gris, la predicción del modelo CNN-GPI.

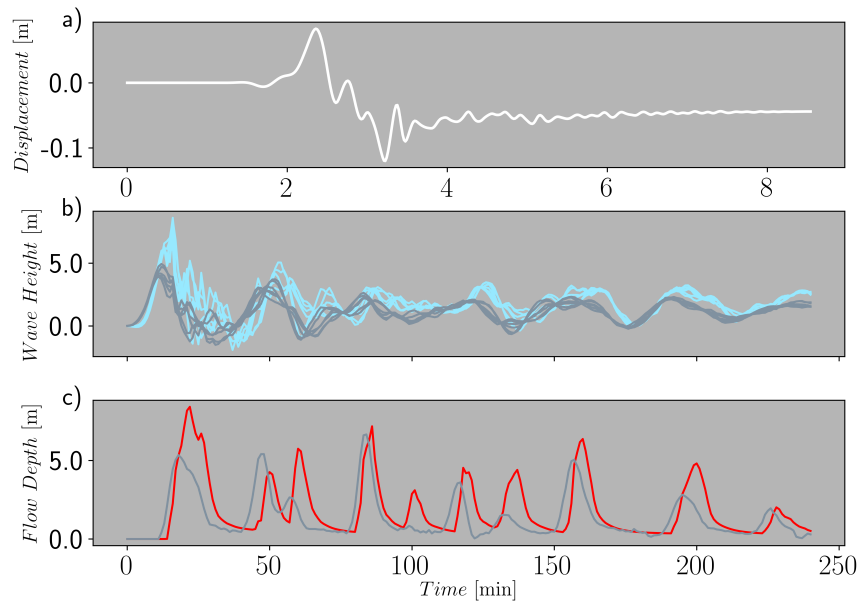


Figura 3.4: Predicción del modelo CNN-GPI para la zona CoB en el escenario # 9412 (Mw 8.9). a) Desplazamiento horizontal en la dirección Este. b) Variación del nivel del mar, donde las líneas magenta representan los valores reales y las grises, la predicción del modelo. c) Profundidad de inundación, con la línea roja indicando los valores reales y la gris, la predicción del modelo CNN-GPI.

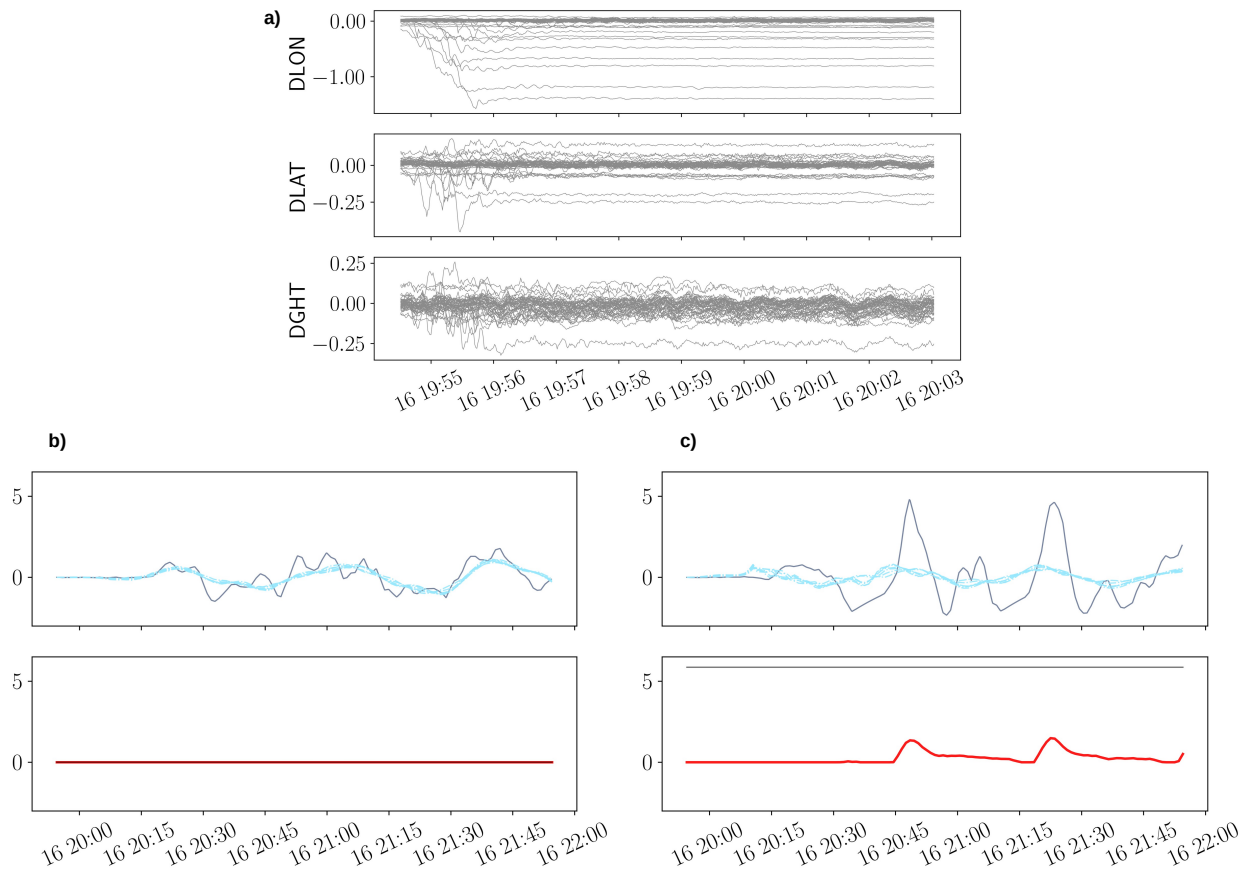


Figura 3.5: Predicción del modelo CNN-GPI para el evento real de Illapel 2015. a) Registros GNSS reales. b) Predicción del modelo en Viña del Mar: la parte superior muestra la predicción en mar y la inferior, en tierra. c) Predicción del modelo en Coquimbo: la parte superior muestra la predicción en mar y la inferior, en tierra.

Capítulo 4

DISCUSIÓN DE LOS RESULTADOS

Aunque los modelos propuestos basados en redes neuronales para calcular series de inundación a partir de registros GNSS no han alcanzado el rendimiento deseado en distintas métricas, su aplicación como clasificadores de amenazas de inundación resulta viable. Esto se debe principalmente a su capacidad para discriminar correctamente entre escenarios sin inundación y eventos de alta peligrosidad, utilizando únicamente como entrada registros GNSS con una duración inferior a 9 minutos.

El principal logro de este enfoque es su aplicabilidad durante eventos reales de emergencia. Si bien la precisión del modelo basado en GNSS es inferior a la del modelo que utiliza las series de elevación instantánea de la superficie libre (η) (Núñez *et al.*, 2022), su ventaja radica en la accesibilidad y disponibilidad en tiempo real de los datos GNSS. En contraste, los modelos basados en registros de series η que dependen de sistemas avanzados de monitoreo en el mar, los cuales no siempre están instalados en todas las regiones de interés y pueden quedar fuera de servicio en situaciones extremas. Por el contrario, las redes de estaciones GNSS están ampliamente distribuidas en muchas regiones del mundo, lo que convierte al modelo propuesto en una herramienta eficaz para generar pronósticos de inundación en tiempo real.

A pesar de las ventajas de utilizar datos GNSS, los resultados de esta investigación subrayan la importancia de contar con información de la serie de elevación instantánea de la superficie libre (η). Las series de registros η reflejan los efectos locales específicos de cada bahía en la propagación por tsunami, lo que contribuye a mejorar la precisión de las predicciones. Sin embargo, uno de los principales desafíos del modelo Red Neuronal Convolutiva unidimensional de dos etapas para Entradas GNSS, Propagación Intermedia y Salidas de Inundación (CNN-GPI) es la propagación de errores: cualquier imprecisión en la predicción de la serie puente (elevación instantánea de la superficie libre (η)) se transmite a la predicción final, generando incertidumbre y reduciendo la precisión en la serie de inundación.

El problema de sensibilidad ante pequeñas variaciones en la entrada ha sido señalado en trabajos previos, como el de Núñez *et al.* (2022), donde se destaca la importancia de contar

con información precisa para evitar que pequeñas diferencias en los datos de entrada generen grandes desviaciones en los resultados. En esta misma línea, el trabajo de [Rim et al. \(2024\)](#) muestra cómo la presencia de ruido en los registros GNSS reales puede incrementar la incertidumbre de la predicción.

En la Tabla 4.1 se cuantifica el desempeño del modelo ante distintos porcentajes de ruido browniano, donde se observa que la zona de CoB es más sensible al ruido en las señales GNSS en comparación con el modelo ajustado en ViB. Las métricas asociadas a señales GNSS con un 20% de ruido muestran un aumento considerable del error, con diferencias notables en amplitud y forma (ver Figura 4.1). Además, se detectó un incremento significativo en la cantidad de alertas falsas, lo que sugiere que los eventos de menor magnitud son los más afectados por la presencia de ruido, pudiéndose incluso perder el registro GNSS real.

Esta sensibilidad a pequeñas variaciones en los datos resalta la necesidad de mejorar la calidad de los registros GNSS y, posiblemente, utilizar técnicas adicionales de filtrado o depuración de datos para reducir los efectos del ruido y mejorar la precisión del modelo. En este sentido, se podría desarrollar un modelo de red neuronal especializado en la reducción de ruido, siguiendo un enfoque similar al propuesto por [Thomas et al. \(2023\)](#).

Tabla 4.1: Métricas de desempeño del modelo CNN-GPI al agregar distintos porcentajes de ruido browniano a las series GNSS de entrada.

Network Model	MSE Test Eq. 2.4		G Test Eq. 2.5		Test		False Alarm			Missed Alarm		
	Inund. [m ²]	No Inund. [m ²]	Inund. [-]	No Inund. [-]	$E_{d,i}$ [m]	$E_{t,i}$ [min]	A	B	C	A	B	C
CNN-GPI												
CoB	0.1109	0.0002	0.3026	0.0475	1.09	9.17	13	8	0	28	9	2
Noise 5 %	0.1303	0.0003	0.3144	0.0769	1.11	9.81	25	9	0	28	8	4
Noise 10 %	0.1851	0.0012	0.3471	0.1222	1.20	11.71	33	16	5	20	9	2
Noise 20 %	0.4077	0.012	0.4236	0.2557	1.42	14.65	48	47	18	16	4	2
ViB	0.0869	0.0013	0.3467	0.1541	1.09	5.27	113	49	5	3	3	0
Noise 5 %	0.0901	0.0016	0.3542	0.1531	1.06	5.76	105	55	6	4	2	0
Noise 10 %	0.1153	0.0029	0.3877	0.1744	1.10	6.93	106	66	17	4	2	1
Noise 20 %	0.2523	0.0121	0.4688	0.2362	1.29	8.74	117	88	51	4	1	1

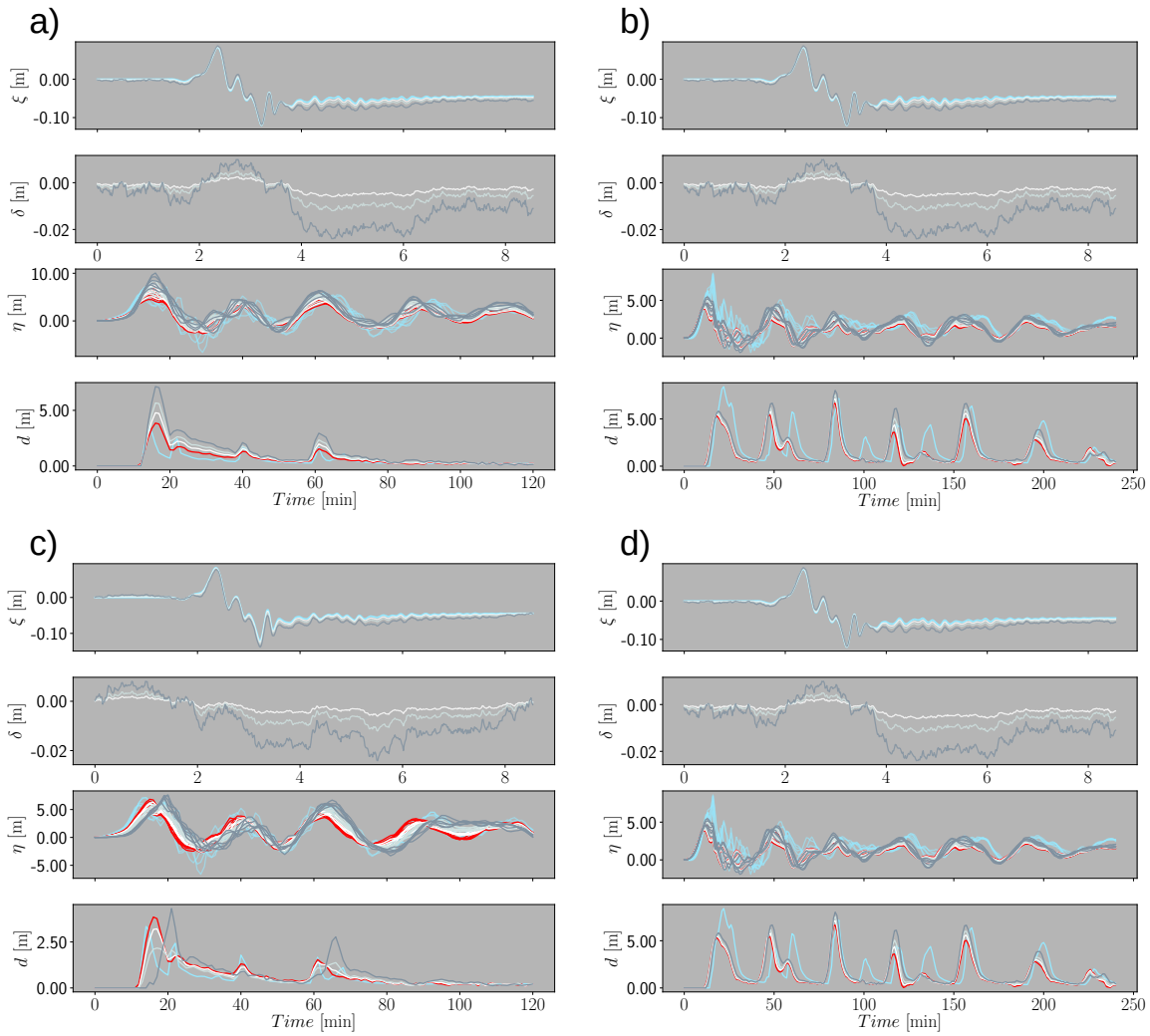


Figura 4.1: Predicción del modelo CNN-GPI con distintos niveles de ruido en las señales GNSS de entrada. En las subfiguras a) y c) se presenta la respuesta del modelo ajustado en ViB, considerando variaciones en el porcentaje y la forma del ruido browniano. En b) y d) se muestra la respuesta del modelo ajustado en CoB. En todas las figuras, las líneas blancas representan la predicción del modelo con un 5% de ruido, las líneas gris claro corresponden a un 10% y las líneas gris oscuro a un 20% de ruido. Además, las líneas rojas muestran la predicción del modelo sin ruido, mientras que las líneas magenta representan los valores reales.

CONCLUSIONES

En resumen, las distintas configuraciones o modelos de redes neuronales desarrollados para calcular series de inundación a partir de señales GNSS ofrecen una alternativa prometedora, especialmente en situaciones donde no se dispone de registros de series η . Si bien los modelos ajustados presentan desempeños similares en bahías con comportamientos hidrodinámicos distintos, como Valparaíso y Coquimbo, aún no alcanzan el nivel de precisión de los modelos que utilizan como variable de entrada la elevación instantánea de la superficie libre (η). No obstante, su principal valor radica en lo accesibles y prácticos que resultan. La posibilidad de contar con datos GNSS en tiempo real los convierte en una herramienta crucial durante emergencias por tsunamis, donde cada minuto cuenta.

Una de las mayores fortalezas de este enfoque es que puede aplicarse incluso cuando los sistemas de monitoreo marítimo están ausentes o no operan debido a las condiciones extremas que suelen acompañar a estos eventos. Además, al concentrar la información más relevante en los primeros nueve minutos, los registros GNSS permiten generar alertas rápidas y efectivas, algo fundamental para salvar vidas y mitigar riesgos en zonas costeras.

Si bien aún queda trabajo por hacer para mejorar la precisión, los modelos propuestos ya han demostrado su capacidad para diferenciar con éxito entre escenarios sin inundación y aquellos de alta peligrosidad. Esto los convierte en una opción valiosa dentro de los sistemas de alerta temprana, especialmente en regiones donde el monitoreo en mar es limitado.

En trabajos futuros será importante enfocarse en perfeccionar las predicciones, especialmente en los casos más complejos, como aquellos cercanos a los límites entre niveles de peligro. También será fundamental abordar el impacto del ruido presente en los registros GNSS. Al probar los modelos ajustados con los registros GNSS reales del terremoto y tsunami de Illapel 2015 (Mw 8.4), el modelo ajustado en Viña del Mar (ViB) mostró un buen desempeño tanto en el cálculo de la serie η como en la inundación. Por el contrario, en el caso de Coquimbo, si bien el cálculo del modelo concuerda con el nivel de amenaza, su desempeño no es tan acertado en la serie en mar η ni en el cálculo de la serie de inundación, subestimando de forma considerable ambas series. Esto podría deberse principalmente a la complejidad del comportamiento hidrodinámico de la bahía de Coquimbo, lo que dificulta la relación entre los registros GNSS y la inundación. Esta dificultad se ve acentuada por el ruido presente, lo

que refuerza la necesidad de explorar formas de filtrado o depuración de estos registros para que el modelo funcione de manera más confiable en condiciones reales.

En conclusión, este trabajo evidencia que, a pesar de los desafíos, los modelos de redes neuronales tienen un enorme potencial para predecir inundaciones por tsunamis a partir de registros GNSS. El enfoque presentado podría convertirse en una herramienta clave para la gestión de riesgos en comunidades vulnerables, ofreciendo una alternativa accesible y eficaz para enfrentar eventos de alta peligrosidad con rapidez y confianza.

REFERENCIAS

Caruana, R., S. Lawrence, and L. Giles (2001), Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, *Advances in neural information processing systems*, pp. 402–408.

Catalán, P. A., A. Gubler, J. Cañas, C. Zuñiga, C. Zelaya, L. Pizarro, C. Valdés, R. Mena, E. Toledo, and R. Cienfuegos (2020), Design and operational implementation of the integrated tsunami forecast and warning system in chile (sipat), *Coastal Engineering Journal*, 62(3), 1–16, doi:10.1080/21664250.2020.1727402.

Davies, G. (2019), Tsunami variability from uncalibrated stochastic earthquake models: tests against deep ocean observations 2006–2016, *Geophysical Journal International*, 218(3), 1939–1960.

Fauzi, A., and N. Mizutani (2020), Machine Learning Algorithms for Real-time Tsunami Inundation Forecasting: A Case Study in Nankai Region, *Pure and Applied Geophysics*, 177(3), 1437–1450, doi:10.1007/s00024-019-02364-4.

Green, G., et al. (1838), On the motion of waves in a variable canal of small depth and width, *Transactions of the Cambridge Philosophical Society*, 6, 457.

He, K., X. Zhang, S. Ren, and J. Sun (2015), Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Hochreiter, S., and J. Schmidhuber (1997), Long short-term memory, *Neural computation*, 9(8), 1735–1780.

Hochreiter, S., Y. Bengio, P. Frasconi, J. Schmidhuber, et al. (2001), Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Ioffe, S., and C. Szegedy (2015), Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *International conference on machine learning*, pp. 448–456, PMLR.

Jordan, M. (1986), Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986, *Tech. rep.*, California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science.

Kamigaichi, O., M. Saito, K. Doi, T. Matsumori, S. Tsukada, K. Takeda, T. Shimoyama, K. Nakamura, M. Kiyomoto, and Y. Watanabe (2009), Earthquake early warning in japan: Warning the general public and future prospects, *Seismological Research Letters*, 80(5), 717–726.

Kingma, D. P., and J. Ba (2015), Adam: A method for stochastic optimization, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun.

Kiranyaz, S., T. Ince, R. Hamila, and M. Gabbouj (2015), Convolutional neural networks for patient-specific ecg classification, in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2608–2611, IEEE.

Kiranyaz, S., O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman (2021), 1d convolutional neural networks and applications: A survey, *Mechanical Systems and Signal Processing*, 151, 107,398, doi:<https://doi.org/10.1016/j.ymssp.2020.107398>.

LeCun, Y., Y. Bengio, and G. Hinton (2015), Deep learning, *nature*, 521(7553), 436–444.

Lipton, Z. C., J. Berkowitz, and C. Elkan (2015), A critical review of recurrent neural networks for sequence learning, *arXiv preprint arXiv:1506.00019*.

Maas, A. L., A. Y. Hannun, A. Y. Ng, et al. (2013), Rectifier nonlinearities improve neural network acoustic models, in *Proc. icml*, vol. 30, p. 3, Atlanta, GA.

Macías, J., M. J. Castro, S. Ortega, C. Escalante, and J. M. González-Vida (2017), Performance Benchmarking of Tsunami-HySEA Model for NTHMP’s Inundation Mapping Activities, *Pure and Applied Geophysics*, 174(8), 3147–3183, doi:10.1007/s00024-017-1583-1.

Makinoshima, F., Y. Oishi, T. Yamazaki, T. Furumura, and F. Imamura (2021), Early forecasting of tsunami inundation from tsunami and geodetic observation data with convolutional neural networks, *Nature communications*, 12(1), 1–10.

Melgar, D., R. J. LeVeque, D. S. Dreger, and R. M. Allen (2016), Kinematic rupture scenarios and synthetic displacement data: An example application to the cascadia subduction zone, *Journal of Geophysical Research: Solid Earth*, 121(9), 6658–6674, doi:10.1002/2016JB013314.

- Mulia, I. E., A. R. Gusman, and K. Satake (2020), Applying a deep learning algorithm to tsunami inundation database of megathrust earthquakes, *Journal of Geophysical Research: Solid Earth*, *125*(9), e2020JB019,690.
- Nair, V., and G. E. Hinton (2010), Rectified linear units improve restricted boltzmann machines, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, p. 807–814, Omnipress, Madison, WI, USA.
- Núñez, J., P. A. Catalán, C. Valle, N. Zamora, and A. Valderrama (2022), Discriminating the occurrence of inundation in tsunami early warning with one-dimensional convolutional neural networks, *Scientific reports*, *12*(1), 1–20.
- Poulos, A., M. Monsalve, N. Zamora, and J. C. de la Llera (2019), An Updated Recurrence Model for Chilean Subduction Seismicity and Statistical Validation of Its Poisson Nature, *Bull. Seismol. Soc. of Amer.*, *109*(1), 66–74, doi:10.1785/0120170160.
- Rim, D., R. Baraldi, C. M. Liu, R. J. LeVeque, and K. Terada (2022), Tsunami early warning from global navigation satellite system data using convolutional neural networks, *Geophysical Research Letters*, *49*(20), e2022GL099,511.
- Rim, D., S. Suri, S. Hong, K. Lee, and R. J. LeVeque (2024), A stability analysis of neural networks and its application to tsunami early warning, *Journal of Geophysical Research: Machine Learning and Computation*, *1*(4), e2024JH000,223.
- Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014), Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, *15*(56), 1929–1958.
- Sutskever, I., O. Vinyals, and Q. V. Le (2014), Sequence to sequence learning with neural networks, in *Advances in neural information processing systems*, pp. 3104–3112.
- Tan, M., and Q. Le (2019), Efficientnet: Rethinking model scaling for convolutional neural networks, pp. 6105–6114.
- Thomas, A., D. Melgar, S. N. Dybing, and J. R. Searcy (2023), Deep learning for denoising high-rate global navigation satellite system data, *Seismica*, *2*(1).
- Zamora, N., P. A. Catalán, A. Gubler, and M. Carvajal (2021), Microzoning Tsunami Hazard by Combining Flow Depths and Arrival Times, *Frontiers in Earth Science*, *8*, 1–18, doi:10.3389/feart.2020.591514.

Zhu, X., and M. Bain (2017), B-cnn: branch convolutional neural network for hierarchical classification, *arXiv preprint arXiv:1709.09890*.