Evaluación de modelos neuronales y estrategias de aumentación de datos para la identificación de trastornos del habla

Iara Kemmerer, Matías F. Gerard, y Leandro D. Vignolo

Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional, sinc(i), FICH-UNL/CONICET
Ciudad Universitaria UNL, 4º piso FICH, Santa Fe (3000), Argentina.
{ikemmerer,mgerard,ldvignolo}@sinc.unl.edu.ar

Abstract. Los trastornos del neurodesarrollo afectan las habilidades de percepción y procesamiento del lenguaje en niños de diferentes edades. Las herramientas de diagnóstico asistido por computadora son de vital importancia para la detección temprana de trastornos del habla y el lenguaje en niños. En este trabajo se explora el uso de modelos de aprendizaje profundo para la detección del Trastorno Específico del Lenguaje a partir de la voz. Para esto se comparan dos enfoques que se emplean para alimentar los modelos neuronales: uno en el que se emplean las señales de audio sin procesar, y otro que se alimenta con espectrogramas derivados de esas señales. Se proponen tres modelos neuronales para realizar la clasificación, dos para el primer enfoque de preprocesamiento de datos y uno para el segundo. Los tres modelos emplean un bloque basado en redes neuronales convolucionales para extraer características. El primero usa una capa completamente conectada como clasificador, mientras que los otros dos procesan la información secuencial mediante redes recurrentes y luego clasifican con una red completamente conectada. A su vez, se exploran estrategias de aumentación de datos, como la adición de ruido, estiramiento temporal, corrimiento temporal y cambio de tono, para analizar su impacto en el desempeño de estas propuestas. Los resultados muestran que los modelos que utilizan las señales sin procesar alcanzan las mejores métricas. Además, el uso de las estrategias de aumentación mejora el desempeño de los modelos propuestos.

Keywords: Modelos neuronales profundos · Trastorno del habla · Aumentación de datos.

1 Introducción

Los trastornos del neurodesarrollo, que afectan diversas funciones esenciales para el día a día de los niños y pueden persistir hasta la edad adulta, incluyen el deterioro específico del lenguaje (SLI, del inglés: Specific Language Impairment), también conocido como disfasia del desarrollo. Este trastorno está estrechamente vinculado con las habilidades de percepción y procesamiento del lenguaje en

niños de diferentes edades. Afecta aproximadamente entre el 2% y el 11% de la población infantil y se presume que tiene un componente genético, lo que se ha demostrado en diversos estudios [10,14]. El SLI se diagnostica cuando un niño muestra un retraso o desorden en el desarrollo del habla sin una causa evidente. A pesar de tener inteligencia no verbal normal y carecer de problemas neurológicos, de audición, comportamiento, emocionales o sociales, estos niños no logran adquirir su lengua materna de manera adecuada o completa. Las manifestaciones típicas incluyen un vocabulario reducido y dificultades para manipular las reglas lingüísticas, lo que a menudo conduce al uso incorrecto de estructuras sintácticas [7,14].

Investigaciones recientes han revelado que los infantes diagnosticados con SLI exhiben patrones vocales singulares, divergentes de aquellos encontrados en niños con desarrollo lingüístico típico [12]. A raíz de esto se plantea la posibilidad de emplear el análisis de diversas características acústicas como herramienta para la detección automática de este trastorno. En respuesta a esta necesidad, se han propuesto abordajes que se enfocan en la caracterización de los contornos de entonación a través de la utilización de descriptores prosódicos de bajo nivel [12]. También se han propuesto recientemente enfoques de aprendizaje profundo que emplean una variante de redes neuronales convolucionales (CNN, del inglés: Convolutional Neural Networks) basada en la arquitectura ResNet, la cual ha demostrado ser eficaz en la detección automática de trastornos del lenguaje, aprovechando su capacidad para procesar eficientemente datos secuenciales como el habla [11]. Otros enfoques emplean modelos que combinan redes convolucionales y recurrentes, utilizando señales de habla sin procesar como entrada [15]. Estos modelos han demostrado alcanzar tasas de precisión notables en la detección de patologías vocales. Paralelamente, se han investigado métodos de extracción de características como los coeficientes cepstrales de frecuencia Mel (MFCC) y el espectrograma, evidenciando su eficacia en la identificación de anomalías en las señales de voz [4].

En un trabajo reciente se propuso un enfoque de aprendizaje profundo para identificar sujetos potenciales de SLI directamente a través de expresiones de habla sin procesar [15]. En base a esto, se diseñaron modelos de aprendizaje automático utilizando las señales de voz cruda, así como los espectrogramas, con el fin de comparar su eficacia en la detección de SLI. Considerando la escasez de datos, los conjuntos disponibles suelen ser limitados en tamaño y diversidad, lo que dificulta la capacidad de entrenar modelos de aprendizaje automático de manera efectiva. Esto se agrava aún más en el caso de modelos de aprendizaje profundo, que requieren grandes cantidades de datos para alcanzar su máximo potencial debido a su alta complejidad y cantidad de parámetros. Para abordar esta limitación, se desarrollaron estrategias de aumento de datos. Estas técnicas incluyen la manipulación de la velocidad de reproducción, la adición de ruido artificial, la variación en la amplitud o tono, entre otras. Su implementación busca enriquecer la diversidad del conjunto de entrenamiento y mejorar la capacidad del modelo para generalizar y detectar patrones relevantes en las expresiones de habla asociadas con el SLI. Este trabajo busca ofrecer una evaluación más

completa de las técnicas de detección de SLI, abordando así un aspecto crucial para el desarrollo de herramientas más precisas y efectivas en este campo.

El resto del artículo se ordena de la siguiente manera. En la Sección 2 se presentan las arquitecturas empleadas y los datos utilizados. También se describe el procesamiento aplicado a las señales antes de alimentar los modelos. En la Sección 3 se describen las estrategias de aumentación aplicadas para aumentar la cantidad de datos. En la Sección 4 se describen los experimentos realizados para evaluar los modelos, se presentan los resultados obtenidos y se realiza la discusión. Finalmente, se presentan las conclusiones y líneas de trabajo futuro en la Sección 5.

2 Materiales y Métodos

2.1 Arquitecturas neuronales empleadas

Redes convolucionales. Las CNN se utilizan comúnmente para tareas de clasificación de imágenes, pero también son muy utilizadas para el reconocimiento de voz y otras aplicaciones basadas en el habla. Al aplicar filtros convolucionales a la señal de habla de entrada, una CNN puede aprender a identificar características importantes, como fonemas y prosodia, que son útiles para el reconocimiento de voz [2].

Una CNN es un modelo matemático compuesto típicamente por tres tipos de capas o bloques de construcción: capas de convolución, de agrupación y totalmente conectadas. Las capas de convolución y de agrupación extraen características de los datos, mientras que las capas totalmente conectadas mapean esas características a una salida final, como la clasificación. Las capas de convolución son fundamentales en las CNN, ya que esta operación lineal permite extraer patrones y características presentes en los datos de entrada. En el contexto de señales de audio, donde los datos están representados secuencialmente en el tiempo, las CNN aplican un filtro, también conocido como kernel, a cada punto de la secuencia temporal de la señal de audio. Este kernel actúa como un extractor de características, analizando fragmentos de la señal en diferentes momentos en busca de patrones significativos [16].

Redes Neuronales Recurrentes. Las redes neuronales (LSTM, del inglés: Long Short-Term Memory) son un tipo especializado de Redes Neuronales Recurrentes (RNN, del inglés: Recurrent Neural Networks) diseñadas para capturar dependencias a largo plazo en secuencias de datos. Esto las hace particularmente útiles para tareas como el procesamiento de lenguaje natural y el reconocimiento de voz.

La arquitectura LSTM se basa en la celda LSTM, que está controlada por tres compuertas principales: la compuerta de entrada (i_t), la compuerta de olvido (f_t) y la compuerta de salida (o_t). Estas compuertas regulan el flujo de información dentro y fuera de la celda [5].

El proceso de la LSTM en cada instante t involucra las operaciones que se detallan a continuación. La compuerta de entrada (i_t) se define como:

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i), \tag{1}$$

donde σ es la función sigmoidea que convierte los valores en un rango entre 0 y 1, W_i es la matriz de pesos específica para la compuerta de entrada, $[h_{t-1}, X_t]$ denota la concatenación de la salida de la celda en el instante anterior h_{t-1} y la entrada actual X_t, y b_i es el sesgo asociado a la compuerta de entrada.

Esta compuerta determina qué nueva información se añadirá al estado de la celda. Al combinar la entrada actual X_t y la salida anterior h_{t-1} a través de la función sigmoidea σ, la compuerta de entrada regula la cantidad de nueva información que se incorporará al estado de la celda en el instante t. Esto permite que el modelo decida qué aspectos de la nueva información son relevantes para el estado actual.

La compuerta de olvido (f_t) se expresa mediante la ecuación:

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f), \tag{2}$$

donde σ es la función sigmoidea que mapea los valores en el intervalo [0, 1], W_f es la matriz de pesos para la compuerta de olvido, $[h_{t-1}, X_t]$ representa la concatenación de h_{t-1} y X_t, y b_f es el sesgo asociado a la compuerta de olvido.

Esta compuerta decide qué parte del estado anterior C_{t-1} debe ser olvidada. La función sigmoidea produce valores que determinan la importancia de cada componente del estado anterior. Un valor cercano a 1 indica que se debe retener la información, mientras que un valor cercano a 0 sugiere que se debe olvidar. Esto es crucial para permitir que la red LSTM ignore información no relevante y se enfoque en datos nuevos más significativos.

La generación de la nueva información candidata (N_t) se describe con la ecuación:

$$N_t = \tanh(W_n[h_{t-1}, X_t] + b_n), \tag{3}$$

donde tanh es la función tangente hiperbólica que restringe los valores al rango [-1, 1], W_n es la matriz de pesos utilizada para generar la nueva información candidata, $[h_{t-1}, X_t]$ indica la concatenación de h_{t-1} y X_t, y b_n es el sesgo para la generación de la nueva información candidata. La Ec. 3 produce los nuevos valores candidatos que podrían actualizar el estado de la celda. La función tangente hiperbólica (tanh) asegura que estos valores estén dentro del rango [-1, 1], proporcionando una escala adecuada para la integración en el estado de la celda.

La actualización del estado de la celda (C_t) se realiza según la ecuación:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot N_t, \tag{4}$$

donde f_t es el valor de la compuerta de olvido en el tiempo t, C_{t-1} es el estado de la celda en el tiempo $t-1$, i_t es el valor de la compuerta de entrada en el tiempo t y N_t es la nueva información candidata generada en el tiempo t.

El estado de la celda se actualiza combinando dos componentes: la parte del estado anterior que se ha decidido mantener ($f_t \cdot C_{t-1}$) y la nueva información candidata ajustada por la compuerta de entrada ($i_t \cdot N_t$).

La compuerta de salida (o_t) se define por la ecuación:

$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o), \tag{5}$$

donde σ es la función sigmoidea que mapea los valores en el intervalo [0, 1], W_o es la matriz de pesos para la compuerta de salida, $[h_{t-1}, X_t]$ representa la concatenación de h_{t-1} y X_t, y b_o es el sesgo asociado a la compuerta de salida. Esta compuerta regula qué parte del estado de la celda será utilizada para la salida. La función sigmoidea combina la información de la entrada actual y la salida anterior para decidir la proporción del estado de la celda que se emitirá como salida.

La salida de la celda (h_t) se calcula mediante la siguiente ecuación:

$$h_t = o_t \cdot \tanh(C_t), \tag{6}$$

donde, o_t es el valor de la compuerta de salida en el tiempo t y $\tanh(C_t)$ es la transformación tangente hiperbólica del estado de la celda C_t.

La salida de la celda se obtiene multiplicando el valor de la compuerta de salida (o_t) por la transformación tangente hiperbólica del estado de la celda actual (C_t). Este producto proporciona la salida final de la celda LSTM en el tiempo t, que se utilizará tanto en el siguiente instante temporal como en la capa de salida de la red.

De este modo, las Ec. 1-6 permiten a la LSTM gestionar de manera eficaz la información en secuencias largas, reteniendo o descartando datos según sea necesario para capturar dependencias temporales.

2.2 Conjunto de datos empleados

La evaluación experimental se llevó a cabo empleando la base de datos LANNA [7]. Los datos corresponden a grabaciones de voz de niños de 4 a 12 años con SLI y grupos de control. Las grabaciones son de un único canal, almacenadas en formato wav, muestreadas a 44,1 kHz y con una resolución de 16 bits. Del conjunto de datos original, se seleccionaron los audios de los niños pronunciando las vocales "a" y "o". Cada niño pronunció ambas vocales, obteniendo un total de 97 ejemplos para cada una. Del total de ejemplos (194 audios), se seleccionaron 43 grabaciones de niños con Desarrollo Típico (TD, por sus siglas en inglés: Typical Development) y 54 de niños con Trastorno Específico del Lenguaje (SLI, del inglés: Specific Language Impairment). Este estudio se centra en las fonaciones de las vocales “a” y “o”, en base a investigaciones anteriores que respaldan su utilidad para detectar patologías de voz [14].

El procesamiento de los datos se basa en la metodología de estudios previos, las señales de voz fueron remuestreadas a 16 KHz, ya que esto permite reducir las operaciones durante el entrenamiento del modelo sin perder información relevante para el problema. Dado que la duración de diferentes fonaciones de la

misma vocal puede variar, y considerando que las CNN requieren una entrada de tamaño fijo para su funcionamiento, el número de muestras en cada segmento de voz se estandarizó en 16000 muestras. Esto se hizo teniendo en cuenta que las señales de audio originales están en el rango [3000, 34000] muestras para una frecuencia de muestreo de 16 KHz. Así, las 16000 muestras abarcan un segundo de señal. Investigaciones anteriores respaldan esta elección como un compromiso adecuado entre la calidad de la señal de entrada y el costo computacional [15]. Toda forma de onda cuya longitud excede el valor establecido se recortó para que tuviera exactamente esa longitud, mientras que si la longitud de la forma de onda era menor que el tamaño especificado, se rellenó con ceros al final de la señal. Luego, en los casos donde se utilizaron los espectrogramas de las señales de voz para alimentar a los modelos, estas se transforman en espectrogramas mediante la transformada de Fourier de tiempo corto (STFT) [9]. Teniendo en cuenta las características de las señales de voz, se ha utilizado una ventana de 30 ms del tipo Hanning con un desplazamiento del 50%. De esta manera se obtienen espectrogramas con 241 valores de frecuencia y 67 ventanas de tiempo.

2.3 Modelos propuestos

El primer enfoque de aprendizaje profundo utilizado en este estudio se basa en el uso de señales de voz sin procesar para la clasificación de niños en categorías de TD o SLI. Dichas señales de voz son preparadas para su posterior alimentación a un modelo de CNN unidimensional (CNN1D o un modelo híbrido que combina CNN1D con LSTM), como se ilustra en la Figura 1. En esta arquitectura, la CNN tiene el objetivo de extraer características que pongan en evidencia la información útil para la clasificación, a la vez de reducir las dimensiones. Mientras que la LSTM le agrega al modelo la capacidad de capturar la dinámica y dependencias temporales en la señal. Se propone además un segundo enfoque en el cual se utilizan los espectrogramas de las señales de voz, que alimentan también al modelo híbrido que combina CNN1D con LSTM, como se muestra en la Figura 2. Esta propuesta se fundamenta en que el espectrograma destaca características importantes de la señal de voz, como formantes, armónicos y otros patrones espectrales. Las CNN pueden aprender a reconocer y extraer estas características de las representaciones más fácilmente del espectrograma, lo que favorece la discriminación entre diferentes sonidos y permite mejorar el desempeño de clasificación. Por otra parte, el dominio frecuencial puede ser más robusto al ruido y a las variaciones del entorno acústico que la señal de audio cruda.

La arquitectura del primer modelo propuesto consiste en una CNN unidimensional (CNN1D) de cinco capas. La primera capa convolucional (CONV1) tiene 10 filtros de 1×160 con un stride de 80. Se aplica la función de activación ReLU (del inglés, Rectified Linear Unit) en cada unidad de la red neuronal. Se utiliza la normalización por lotes (del inglés, Batch Normalization) después de la capa de entrada y cada capa convolucional para acelerar el entrenamiento y reducir el sobreajuste. Además, se incorporaron capas de agrupación máxima (del inglés, Max Pooling) después de las capas CONV2-CONV4 para reducir la

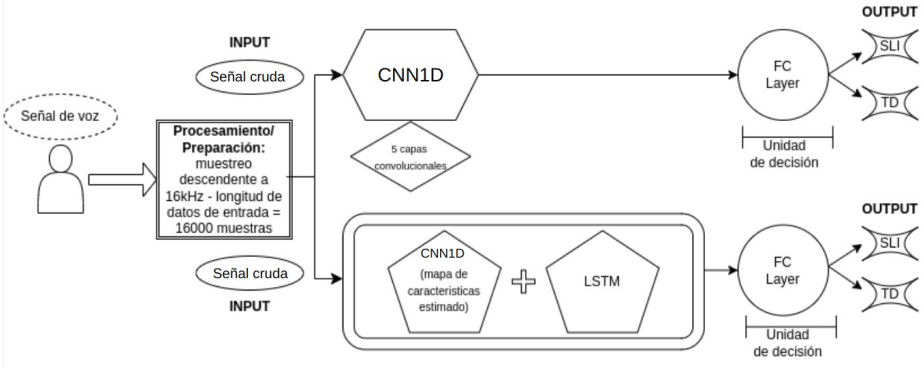

Fig. 1. Diagrama del primer enfoque de aprendizaje profundo empleado en este estudio, un modelo neuronal para el procesamiento de señales de voz crudas. Comienza con la señal de voz capturada y preprocesada a 16 kHz, produciendo segmentos de 16000 muestras. El modelo ofrece dos flujos: uno con una Red Neuronal Convolucional de una dimensión (CNN1D) de cinco capas para extraer características, y otro que combina la CNN1D con una red LSTM para capturar patrones temporales. Ambos flujos terminan en una capa completamente conectada (FC Layer) para la clasificación final en categorías como "SLI" o "TD".

dimensionalidad de los datos mientras se mantienen las características importantes. El mapa de características obtenido de la capa CONV5 alimenta a una capa completamente conectada (FC). Para este trabajo, se utilizó una sola capa FC con 2 salidas. La capa FC, utiliza la función de activación softmax para predecir la etiqueta de clase de salida (TD o SLI). Además, se presenta una arquitectura híbrida de CNN1D con una LSTM. Empleando la misma estructura que en el modelo CNN1D descrito, los mapas de características obtenidos en la última capa (CONV5) alimentan la entrada de la estructura LSTM, que tiene el objetivo de capturar la información de la dinámica temporal de la señal que sea relevante para la clasificación. La LSTM posee una única capa con 50 unidades ocultas. A su vez, el modelo emplea dropout como regularizador, con probabilidad de 0.2 [15].

En el segundo enfoque, el modelo híbrido propuesto utiliza dos capas convolucionales. La primera capa convolucional (CONV1) opera sobre los espectrogramas de entrada, utilizando filtros de convolución con un kernel de longitud 4 y un stride de 2, lo que produce 121 mapas de características. Se aplica la función de activación ReLU para introducir no linealidades en el modelo, seguida de normalización por lotes (Batch Normalization) para mejorar la estabilidad y acelerar la convergencia. La segunda capa convolucional (CONV2) sigue el mismo patrón, generando 60 mapas de características que representan las características de alto nivel de los datos de entrada. Posteriormente, después de la segunda capa convolucional (CONV2), la salida es aplanada y pasada a través de la misma capa LSTM para modelar las dependencias temporales. La arquitectura de la LSTM, así como su configuración de dropout, se mantienen iguales

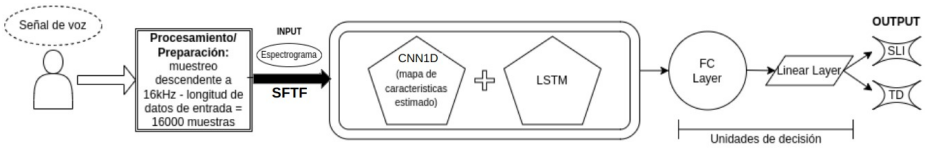

Fig. 2. Diagrama del segundo enfoque de aprendizaje profundo empleado en este estudio, un modelo neuronal que utiliza espectrogramas de señales de voz para la clasificación. La señal de voz preprocesada se convierte en un espectrograma mediante STFT. Este espectrograma se procesa con una CNN1D seguida de una LSTM, capturando tanto características espaciales como temporales de la señal. La salida se clasifica a través de una capa completamente conectada y una capa lineal para decidir entre "SLI" o "TD".

que en el primer enfoque. La LSTM, con células de memoria que mantienen y actualizan su estado interno en función de las entradas actuales y anteriores, permite capturar dependencias a largo plazo en los datos de secuencia, crucial en tareas de procesamiento de voz. Finalmente, la salida de la capa LSTM se pasa a través de la misma capa FC del primer enfoque para realizar la clasificación final. Esta capa FC, al igual que en el primer enfoque, consta de dos neuronas para problemas de clasificación binaria, como la identificación de comandos de voz, y utiliza una función de activación de salida adecuada, como la función softmax.

3 Aumentación de datos

Con el objetivo de disponer de un mayor conjunto de datos de entrenamiento, y mejorar la robustez de los modelos, se evaluaron 4 estrategias de aumentación: adición de ruido blanco (*noise*), estiramiento (*stretching*), corrimiento temporal (*time-shifting*) y cambio de tono (*pitch-shifting*). La elección de estas técnicas se basa en estudios recientes que han demostrado su eficacia en la aumentación de datos para señales de audio [1,13]. Estas estrategias se aplican sobre las señales de audio crudo, previamente a ingresar a los modelos neuronales, de manera de producir nuevas señales de audio que contengan ligeras distorsiones.

La estrategia *noise* consiste en introducir aleatoriedad controlada en una señal existente mediante la adición de ruido blanco. Este proceso implica la superposición de una señal de ruido blanco, que tiene una distribución uniforme de amplitud en todas las frecuencias, sobre la señal original. La ecuación para realizar este proceso es $y[n] = x[n] + q[n]$, donde $y[n]$ es la señal aumentada, $x[n]$ es la señal original y $q[n]$ es el ruido blanco añadido. Esta técnica es útil para aumentar la robustez de modelos de procesamiento de señales, mejorar la generalización y evitar el sobreajuste en tareas de aprendizaje automático.

La técnica de *stretching* consiste en la modificación de la duración temporal de una señal sin alterar su contenido espectral. Esto se logra mediante la interpolación o decimación de la señal original, siguiendo la ecuación: $y[n] = x[n]$, donde $y[n]$ es la señal estirada, $x[n]$ es la señal original, y es el factor de estiramiento

(o compresión). Esta operación modifica la velocidad del audio sin cambiar el tono.

La técnica de *time shifting* consiste en desplazar temporalmente una señal a lo largo del eje del tiempo sin cambiar su contenido espectral. Esto se logra mediante el retraso o adelanto de la señal en el dominio del tiempo, empleando la ecuación $y[n] = x[n - \delta]$, donde $y[n]$ es la señal desplazada, $x[n]$ es la señal original, y δ es el desplazamiento temporal. En particular, se empleó la alternativa que realiza un corrimiento circular, que implica insertar al inicio o al final de la señal aquellas muestras que quedan por fuera del tamaño de la ventana luego de aplicado el desplazamiento.

La técnica de *pitch-shifting* consiste en modificar el tono de una señal de voz alterando la frecuencia de las componentes espectrales de la señal, aumentando o disminuyendo su frecuencia de acuerdo a la ecuación: $Y[k] = X[k \times \frac{f_s}{f'_s}]$, donde $Y[k]$ es la señal de voz modificada, $X[k]$ es la señal original en el dominio de la frecuencia discreta k, f_s es la nueva frecuencia de muestreo y f'_s es la frecuencia de muestreo original.

4 Resultados y Discusión

En esta sección se presentan los experimentos llevados a cabo con cada modelo neuronal propuesto para clasificar niños con TD o SLI. Primero se describen los experimentos empleando el modelo convolucional (CNN-R) y el modelo híbrido (Hybrid-R) alimentados con las señales crudas, para determinar y ajustar los hiperparámetros utilizados para el entrenamiento de los tres modelos. Luego se evalúa el modelo híbrido de dos capas convolucionales que es alimentado por espectrogramas (Hybrid-S) para determinar el mejor enfoque y arquitectura. Finalmente, se evalúan los modelos con las 4 estrategias de aumentación, para determinar si estas permiten obtener un mejor desempeño de generalización.

4.1 Detalle experimental y métricas de evaluación

La etapa experimental se llevó a cabo empleando un esquema común para todos los experimentos. Se generaron cinco particiones para entrenar y probar los modelos con diferentes ejemplos, siguiendo el esquema definido en [15]. Cada partición consiste en 60 ejemplos de entrenamiento (30 TD y 30 SLI) y 37 ejemplos de prueba (13 TD y 24 SLI). Para evitar el sobre-entrenamiento de los modelos y obtener la mayor capacidad de generalización posible, cada partición de entrenamiento se subdividió en un esquema de 80% para entrenamiento y un 20% para validación. De esta manera se realizó el monitoreo luego de cada época de entrenamiento para conservar el modelo con mejor desempeño en el conjunto de validación.

Dado que el conjunto de test empleado presenta desbalance de clases, el desempeño de los modelos se evaluó empleando la medida F1-score [3]. Esta medida se calcula como la media armónica entre precisión p, y recall r, de acuerdo

a la ecuación:

$$F1 = \frac{2 \cdot p \cdot r}{p + r}. \tag{7}$$

Esta medida toma valores en el rango $[0, 1]$, siendo máxima cuando todas las predicciones son correctas.

Para cada modelo analizado se realizó una etapa preliminar de optimización de hiperparámetros. Para seleccionar y ajustar los hiperparámetros de entrenamiento de los modelos propuestos, se realizó una exploración tomando como punto de partida la configuración utilizada en [15]. En dicha configuración, el entrenamiento se realiza en 30 épocas, empleando gradiente descendente con mini-batch, una tasa de aprendizaje de 0.01 y un término de momento de 0.9. Los pesos de los modelos fueron inicializados empleando el método de Glorot [11]. Al finalizar un entrenamiento, se mantuvo el modelo con el conjunto de pesos que logró el máximo valor de F1-score con los datos de validación. Para la capa de normalización de batch se fijó un valor $\varepsilon = 1 \times 10^{-5}$, un valor de escala $\gamma = 1$ y un valor de compensación $\beta = 0$. Se empleó la entropía cruzada para evaluar el error, y un valor de $\lambda = 0.0001$ como parámetro L2 de regularización.

Se utilizó el modelo CNN-R para realizar un conjunto de experimentos preliminares, con el objetivo de optimizar la configuración base, empleando el conjunto de datos original sin aumentación. Para mitigar los efectos de variabilidad asociados a la inicialización, se realizaron cinco repeticiones de cada experimento. En primer lugar, se evaluó de esta manera el desempeño del modelo con la configuración base. Para este experimento se realizó el entrenamiento y la evaluación del modelo para cada una de las 5 particiones de datos, considerando el promedio y el desvío del F1-score. El mismo experimento se repitió, aumentando la cantidad de épocas a 100, observando que no había mejoras significativas. Seguido, manteniendo las épocas en 100, se disminuyó la tasa de aprendizaje a 0.0001. Se observó que al disminuir la tasa de aprendizaje, el modelo obtiene un mejor desempeño. De esta forma, se establecieron los siguientes hiperparámetros para entrenar los modelos: 30 épocas de entrenamiento, optimizador SGD, mini-batch de 5, una tasa de aprendizaje de 0.01, un término de momento de 0.9 y un valor de $\lambda = 0.0001$ como parámetro L2 de regularización. Para la inicialización de los pesos de los modelos se mantuvo el método de Glorot [6] para la capa FC, y para las capas convolucionales se utilizó el método de He [8], ya que este presentó mayor estabilidad en las pruebas preliminares. Para la capa de normalización de batch se empleó un valor de (ϵ) igual a 1×10^{-5} para estabilizar el cálculo de la varianza durante la normalización; el valor de (γ) igual a 1 para controlar la amplitud de las activaciones normalizadas permitiendo el ajuste de escala; y el factor de desplazamiento (β) igual a 0. Como función de pérdida se empleó la entropía cruzada.

4.2 Evaluación de los modelos con el conjunto de datos original

Primero se evaluaron los modelos CNN-R, Hybrid-R y Hybrid-S empleando el conjunto de datos original sin aumento. En cada experimento se entrenó un modelo independiente para cada vocal considerada. El entrenamiento se realizó

Tabla 1. Resultados obtenidos de la métrica F1-score para las 5 particiones sin aumento de datos. Se presentan los promedios de 5 repeticiones por partición y el promedio general de las 5 particiones. Se resaltan en negrita: el mejor desempeño de F1-score para cada partición entre los modelos de cada vocal y los modelos con mejor desempeño promedio para cada vocal.

	"a"			"o"		
	CNN-R	**Hybrid-R**	**Hybrid-S**	**CNN-R**	**Hybrid-R**	**Hybrid-S**
Partición 1	**81.90**	68.49	81.15	**82.34**	79.87	77.74
Partición 2	**82.24**	78.75	75.95	73.19	**78.36**	72.67
Partición 3	83.22	**84.44**	73.37	74.51	77.91	**80.11**
Partición 4	72.90	**81.73**	71.97	69.59	**80.17**	69.32
Partición 5	**73.05**	71.44	57.75	**73.8**	71.53	68.07
Promedio	**78.66**	76.97	72.04	74.68	**77.57**	73.58

con las cinco particiones de datos, realizando 5 repeticiones para estimar el desempeño independientemente de la inicialización de los pesos. Cada repetición fue identificada empleando el término sesión.

La Tabla 1 muestra el desempeño de la métrica F1-score obtenido en los experimentos sin aumento de datos. Se muestra el valor promedio de las 5 repeticiones por partición, modelo y vocal. Como se puede observar, el modelo CNN-R obtiene el mejor desempeño promedio para la vocal "a" (F1-score promedio de 78.66). Aunque el modelo Hybrid-R logra mejor desempeño en dos de las cinco particiones, el modelo CNN-R exhibe mayor consistencia, lo que sugiere una mejor estabilidad. En el caso de la vocal "o" se observa la situación inversa, el modelo Hybrid-R logra el mejor desempeño general (F1-score promedio de 77.57) y resultados más estables a lo largo de las particiones. Esto sugiere que la información secuencial procesada por la LSTM contribuye, en este último caso, a mejorar el desempeño. El modelo Hybrid-S, por su parte, obtiene un desempeño inferior en ambos casos.

4.3 Evaluación del impacto de las estrategias de aumentación

Con el objetivo de analizar el impacto en el entrenamiento y la capacidad de generalización de los modelos, se realizó la evaluación empleando las técnicas de aumentación de datos descritas en la Sección 3. La aumentación de datos se realizó sobre los datos de entrenamiento de cada partición, manteniendo los conjuntos de validación y prueba originales (37 ejemplos para todas las particiones). Para cada tipo de transformación se realizaron dos sesiones, generando nuevos ejemplos para incrementar en 5 y 10 veces el tamaño del conjunto de datos, respectivamente.

Para aplicar la técnica de *time shifting* se utilizó un desplazamiento de 10 ms en la primera sesión y de 20 ms en la segunda sesión. Para la técnica de *stretching*, se utilizó un factor aleatorio, que en este contexto es un valor multiplicador, extraído de una distribución uniforme, denotada como $U(a, b)$, en el intervalo $[0.8, 1.2]$. Este factor sirve para comprimir (si es menor que 1) o estirar (si es

Tabla 2. Resultados obtenidos de F1-score para la vocal "a" utilizando técnicas de aumentación de datos durante el entrenamiento. Se resaltan: en negrita, los valores que superan el F1-score obtenido en los experimentos sin aumento de datos y en sombreado gris, el mejor desempeño de cada modelo entre todas las técnicas de aumentación.

Vocal "a"									
	S/A	**Time Shift**		**Noise**		**Stretch**		**Pitch Shift**	
		5	**10**	**5**	**10**	**5**	**10**	**5**	**10**
CNN-R	78.66	**79.13**	74.09	77.56	73.64	76.36	**79.76**	**79.71**	**80.17**
Hybrid-R	76.97	76.23	74.23	75.27	72.28	**78.89**	**77.31**	**81.72**	**80.28**
Hybrid-S	72.04	**77.02**	**73.30**	**75.36**	**78.13**	69.19	71.68	**83.25**	**81.46**

Tabla 3. Resultados obtenidos de F1-score para la vocal "o" utilizando técnicas de aumentación de datos durante el entrenamiento. Se resaltan: en negrita, los valores que superan el F1-score obtenido en los experimentos sin aumento de datos y en sombreado gris, el mejor desempeño de cada modelo entre todas las técnicas de aumentación.

Vocal "o"									
	S/A	**Time Shift**		**Noise**		**Stretch**		**Pitch Shift**	
		5	**10**	**5**	**10**	**5**	**10**	**5**	**10**
CNN-R	74.68	**74.79**	65.87	**76.71**	67.91	72.29	50.27	**81.70**	**77.89**
Hybrid-R	77.57	72.20	74.17	74.18	**80.61**	70.48	77.40	**78.26**	**79.67**
Hybrid-S	73.58	69.03	72.94	70.30	70.07	67.83	**76.79**	**75.43**	**77.51**

mayor que 1) la señal original. Para la técnica *noise* se utilizó una distribución normal con media 0 y varianza 1. El ruido añadido a la señal se moduló mediante un factor de ganancia aleatorio, calculado como $G = 0.05 \times L \times M$, donde L es un valor aleatorio de una distribución uniforme $U(0, 1)$ y M es el máximo valor absoluto de la señal. Para la técnica de *pitch shifting* se consideraron 12 pasos por octava, y para el desplazamiento se utilizó un número entero de pasos aleatorio, también obtenido de una distribución uniforme $U(-3, 4)$.

La Tabla 2 muestra los valores de la métrica F1-score, correspondientes a la vocal "a", obtenidos empleando las técnicas de aumentación en el entrenamiento de los tres modelos. La Tabla 3, de la misma manera, presenta los resultados para la vocal "o". Estas tablas muestran el desempeño de los tres modelos comparados: CNN-R, Hybrid-R y Hybrid-S, entrenados con y sin técnicas de aumentación. Como se puede observar, las técnicas de aumentación permiten en general lograr un mejor desempeño de los modelos, de manera similar para ambas vocales. Particularmente, la técnica de *pitch shifting* permite mejorar los resultados de los tres modelos entrenados sin aumentación, y considerando ambos factores de aumentación. Puntualmente, para la vocal "a", la técnica de *pitch shifting* permite al modelo Hybrid-S, que mostraba el peor desempeño sin aumentación, obtener el mejor desempeño global. Las otras técnicas, por otro lado, permiten obtener mejoras en ciertos casos en particular. Por ejemplo, para el modelo Hybrid-S, las técnicas *time shifting* y *noise* permiten mejorar el desempeño en el caso de la

vocal "a". Para la misma vocal, además, la técnica de *stretching* permite obtener mejoras en los dos modelos para señales sin procesar, CNN-R y Hybrid-R.

5 Conclusiones y trabajos futuros

En este trabajo se exploró el uso de modelos de aprendizaje profundo para la detección de SLI, a partir de grabaciones de fonaciones de vocales de niños sanos y con patologías del habla. Nuestro estudio se centró en comparar diferentes modelos neuronales, contemplando dos estrategias de preprocesamiento de datos, y evaluando el impacto de distintas técnicas de aumentación de datos en el desempeño de detección automática de SLI.

Se evaluaron tres modelos neuronales: dos basados en el procesamiento de datos crudos (CNN1D y una combinación de CNN1D con LSTM) y uno que emplea el espectrograma de las señales de audio (una combinación de CNN1D con LSTM). Los resultados muestran que mientras el modelo CNN1D presentó un mejor desempeño para reconocer la vocal "a", el modelo híbrido para señales crudas fue superior en la discriminación de la vocal "o". Esto sugiere la presencia de características en las señales crudas que son mejor aprovechadas por los modelos. A su vez, el uso de estrategias de aumentación mejoró el desempeño de los modelos propuestos. En particular, el *pitch shifting* mostró ser la estrategia más efectiva para mejorar la capacidad de generalización de los modelos. Estos resultados permiten así concluir que los modelos de aprendizaje profundo muestran un gran potencial para la detección temprana de trastornos del habla.

En base a estos resultados prometedores, se hace necesario continuar trabajando para reducir la variabilidad observada en el desempeño de los modelos. Para esto se propone evaluar nuevos esquemas de particionado y estrategias de validación que favorezcan el aprendizaje y generalización de los modelos. A su vez, un aspecto que no fue considerado en este estudio, pero que puede presentar un importante efecto en el desempeño de los modelos, es la presencia de sesgo por sexo. Para eso, se planea entrenar y evaluar modelos independientes. En desarrollos futuros se trabajará además en la optimización de la arquitectura de los modelos neuronales. Particularmente, se considerará la utilización de enfoques evolutivos, ya que estos recientemente han emergido como una alternativa con importantes ventajas en el diseño de modelos neuronales.

References

1. Abayomi-Alli, O.O., Damaševičius, R., Qazi, A., Adedoyin-Olowe, M., Misra, S.: Data augmentation and deep learning methods in sound classification: A systematic review. Electronics **11**(22) (2022). https://doi.org/10.3390/electronics11223795
2. Amami, R., Amami, R., Trabelsi, C., Mabrouk, S., Khalil, H.: A robust voice pathology detection system based on the combined bilstm–cnn architecture. MENDEL **29**(2), 202–210 (Dec 2023). https://doi.org/10.13164/mendel.2023.2.202

3. Bekkar, M., Djema, H., Alitouche, T.: Evaluation measures for models assessment over imbalanced data sets. Journal of Information Engineering and Applications **3**, 27–38 (01 2013)
4. Chaiani, M., Selouani, S.A., Boudraa, M., Sidi Yakoub, M.: Voice disorder classification using speech enhancement and deep learning models. Biocybernetics and Biomedical Engineering **42**(2), 463–480 (2022). https://doi.org/10.1016/j.bbe.2022.03.002
5. Er, M.B., Isik, E., Isik, I.: Parkinson's detection based on combined cnn and lstm using enhanced speech signals with variational mode decomposition. Biomedical Signal Processing and Control **70**, 103006 (2021). https://doi.org/10.1016/j.bspc.2021.103006
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research - Proceedings Track **9**, 249–256 (01 2010)
7. Grill, P., Tučková, J.: Speech databases of typical children and children with sli. PLOS ONE **11**(3), 1–21 (03 2016). https://doi.org/10.1371/journal.pone.0150365
8. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1026–1034 (2015). https://doi.org/10.1109/ICCV.2015.123
9. Jeon, H., Jung, Y., Lee, S., Jung, Y.: Area-efficient short-time fourier transform processor for time–frequency analysis of non-stationary signals. Applied Sciences **10**(20) (2020). https://doi.org/10.3390/app10207208
10. K. F. Swaiman, S. Ashwal, D.M.F.N.F.S.R.S.F., Gropman, A.L.: Swaiman's pediatric neurology: Principles and practice (2017)
11. Kotarba, K., Kotarba, M.: Efficient detection of specific language impairment in children using resnet classifier. pp. 169–173 (09 2020). https://doi.org/10.23919/SPA50552.2020.9241289
12. Ringeval, F., Demouy, J., Szaszak, G., Chetouani, M., Robel, L., Xavier, J., Cohen, D., Plaza, M.: Automatic intonation recognition for the prosodic assessment of language-impaired children. IEEE Transactions on Audio, Speech, and Language Processing **19**(5), 1328–1342 (2011). https://doi.org/10.1109/TASL.2010.2090147
13. Salamon, J., Bello, J.P.: Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters **24**(3), 279–283 (2017). https://doi.org/10.1109/LSP.2017.2657381
14. Sharma, Y., Singh, B.K.: Prediction of specific language impairment in children using speech linear predictive coding coefficients. In: 2020 First International Conference on Power, Control and Computing Technologies (ICPC2T). pp. 305–310 (2020). https://doi.org/10.1109/ICPC2T48082.2020.9071510
15. Sharma, Y., Singh, B.K.: One-dimensional convolutional neural network and hybrid deep-learning paradigm for classification of specific language impaired children using their speech. Computer Methods and Programs in Biomedicine **213**, 106487 (2022). https://doi.org/10.1016/j.cmpb.2021.106487
16. Yamashita, R., Nishio, M., Do, R., Togashi, K.: Convolutional neural networks: an overview and application in radiology. Insights into Imaging **9** (06 2018). https://doi.org/10.1007/s13244-018-0639-9