

Combining Residual and LSTM Recurrent Networks for Transportation Mode Detection Using Multimodal Sensors Integrated in Smartphones

Profesor: Hans Lobel

Alumno: Vicente del Río

I. Contexto

- Importancia modo de transporte:
 - Predicción y planificación del trafico
 - Calculo huella de carbono
 - Extracción de patrones de flujos de la población
 - Optimización de redes
- Teléfonos inteligentes
- Propuesta: LSTM con redes residuales que utiliza múltiples sensores ligeros integrados en teléfonos inteligentes

II. Trabajos relacionados

- Uso de GPS: Alto consumo de energía y la falta de disponibilidad en los entornos sin GPS limitan su amplio uso
- Aprendizaje automático con sensores ligeros
- Aprendizaje profundo:
 - Dabiri S: Utilizó una red neuronal convolucional
 - Francisco: Presentó una red híbrida que combina una red convolucional profunda y una red LSTM
 - Kalatian: Diseñó una red residual utilizando señales Wi-fi

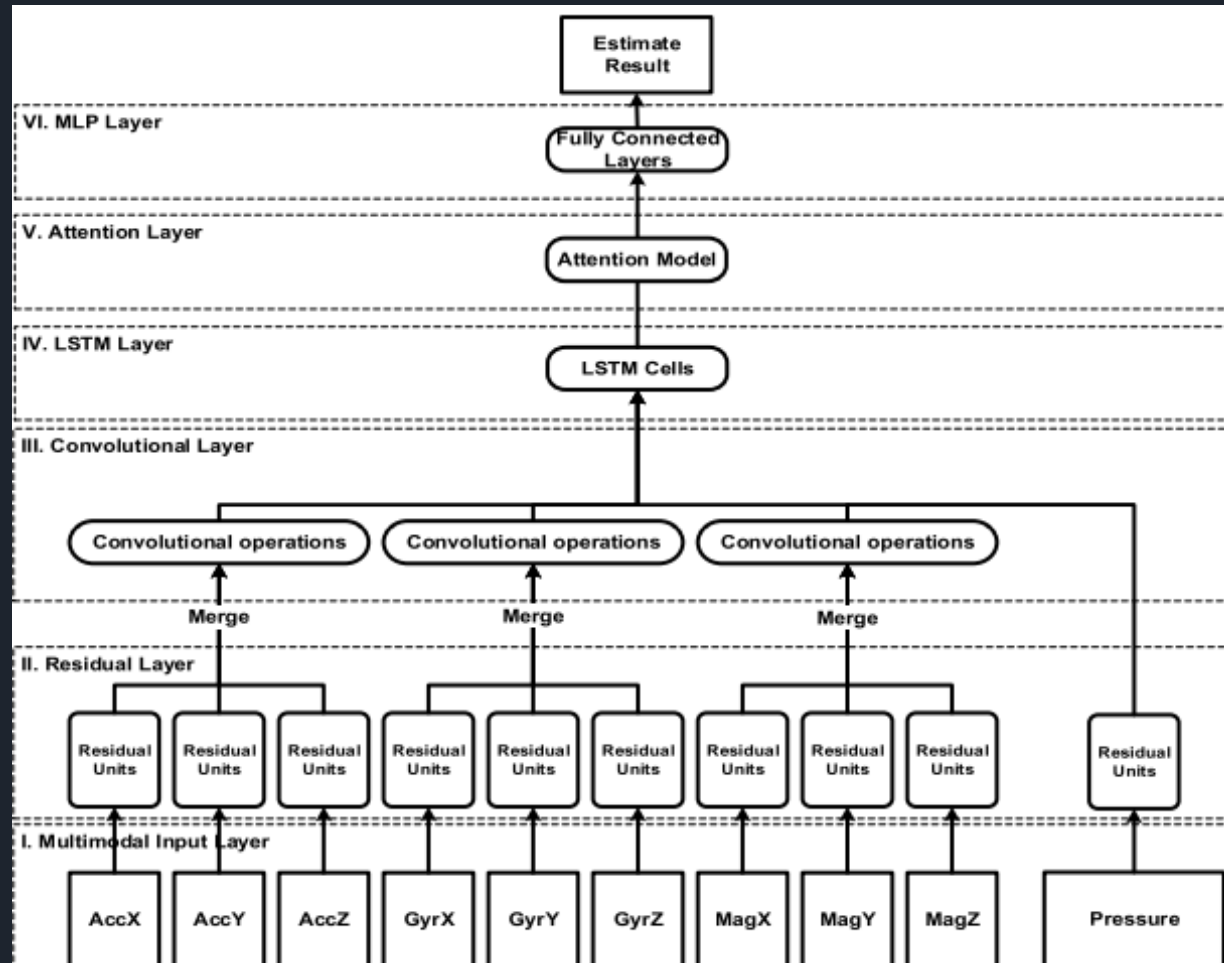
III. Propuesta

“A diferencia de los métodos antes mencionados, en este documento, propone un LSTM basado en la atención residual para la detección del modo de transporte utilizando múltiples sensores de peso ligero integrados en los teléfonos inteligentes básicos. La introducción de la unidad residual en la red neuronal profunda no solo acelera el proceso de entrenamiento, sino que también reduce la influencia de degradación de múltiples capas ocultas”

III. Algoritmo

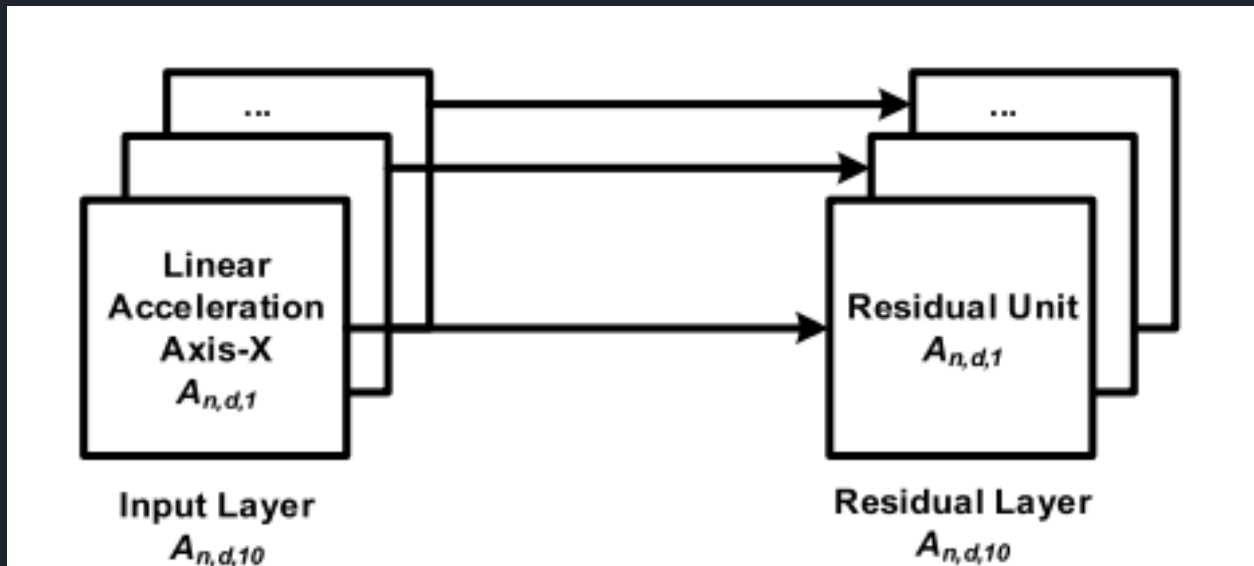
- El algoritmo de aprendizaje profundo propuesto para la detección del modo de transporte incluye tres pasos principales:
 - Preprocesamiento de datos
 - Entrenamiento del modelo MSRLSTM
 - Predicción del modo de transporte

III.1 Arquitectura



III.1 Arquitectura

1. Capa de entrada multimodal



$A_{n,d,k}$

n : número de muestras totales

d : longitud de la ventana deslizante

K : número total de elementos de todos los sensores

Figura 1: Estructura de la capa de entrada

III.1 Arquitectura

2. Capa residual

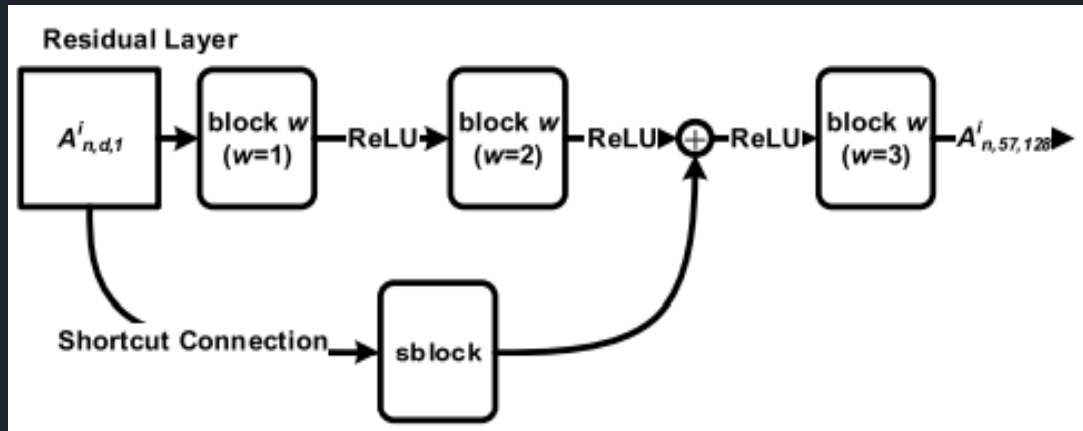


Figura 2: Arquitectura de la unidad residual

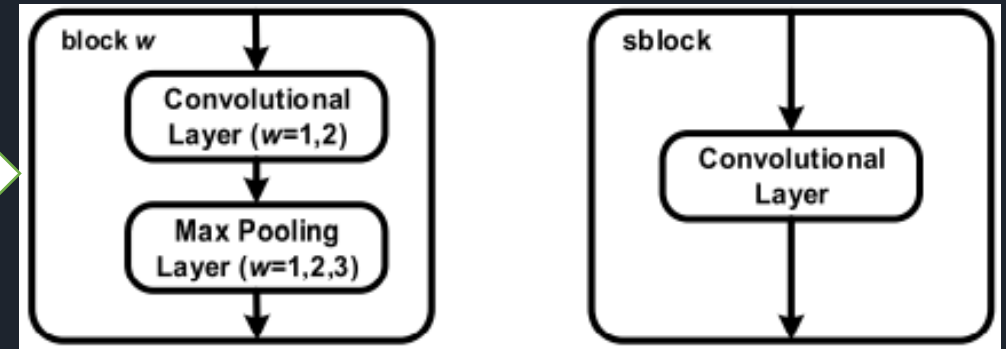


Figura 3: El diseño de bloque para la unidad residual

III.1 Arquitectura

3. Capa convolucional:

- Reducir el tamaño de los datos
- Acelerar el proceso de formación
- Mejorar el ajuste del modelo
- Aprendizaje de característica (aceleración, giroscopio, magnetismo y datos de presión barométrica)

III.1 Arquitetura

4. Capa LSTM

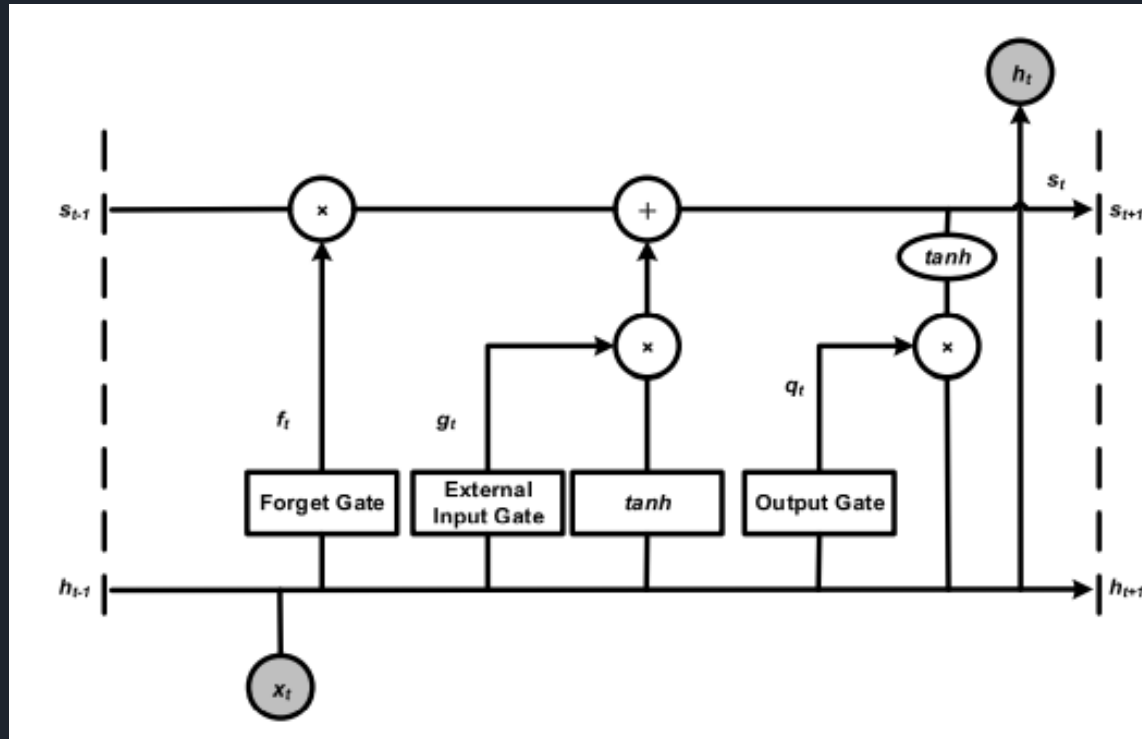


Figura 4: Arquitetura celular LSTM

III.1 Arquitectura

5. Capa de atención

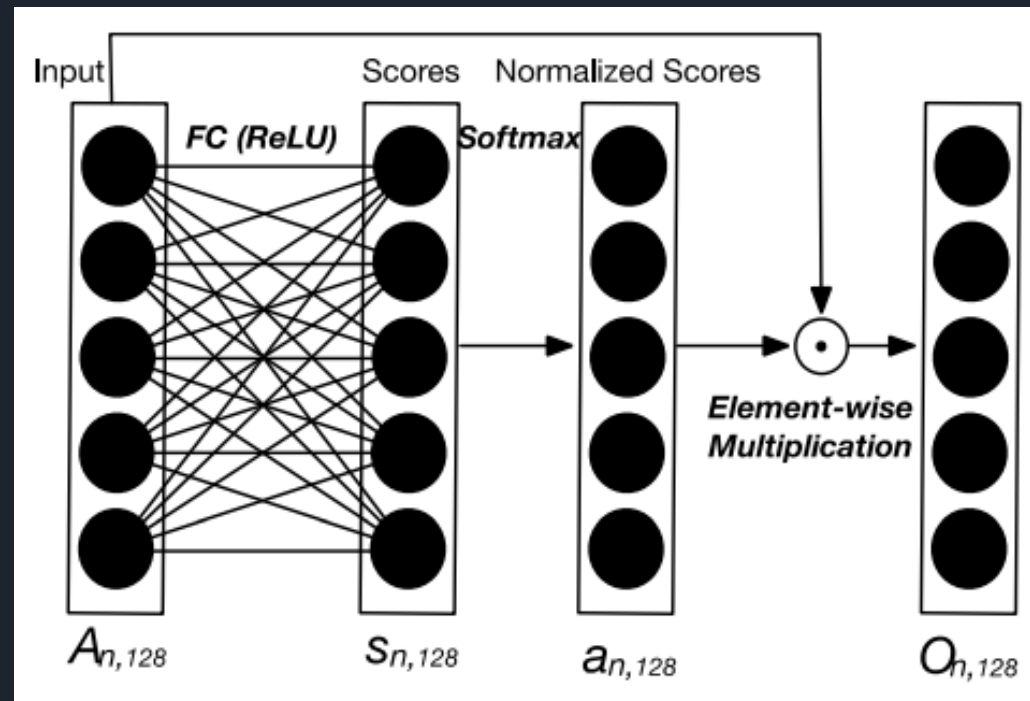


Figura 5: Arquitectura capa de atención

III.1 Arquitectura

6. Capa MLP:

- Dropout 0.2
- Numero capas ocultas:
 - 128 ReLU
 - 256 ReLU
 - 512 ReLU
 - 1024 ReLU
 - 8 Softmax

III.1 Arquitectura

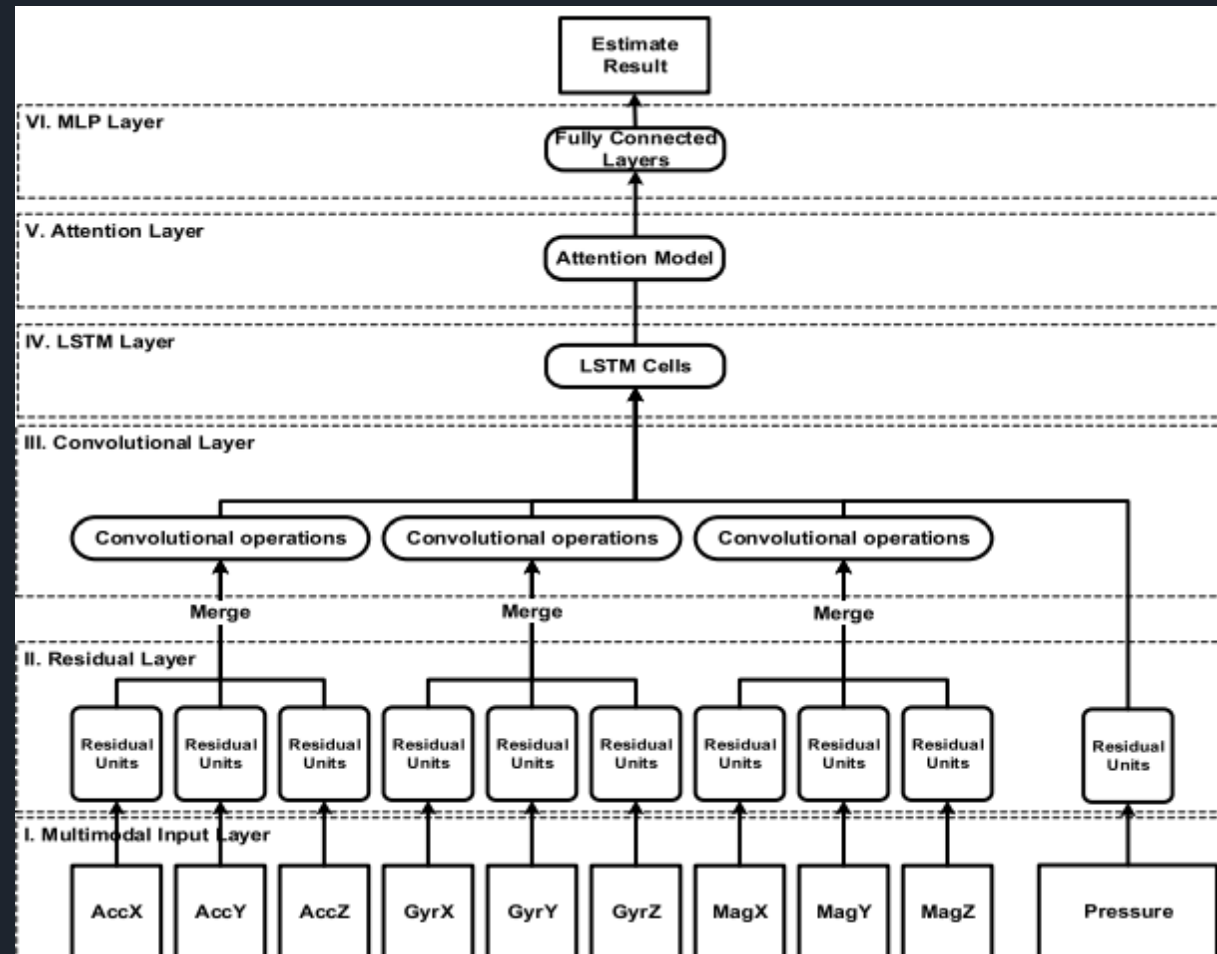


Figura 6: Arquitectura MSRLSTM

IV. Evaluación experimental

1. Conjunto de datos

- Tres conjuntos de datos: SHL, HTC y TIC
- Todos los conjuntos de datos se transforman en matrices uniformes y luego se procesan con lote normalización y segmentación mediante una ventana deslizante para reducir la influencia de los ruidos ambientales.
- Los datos preprocesados se separaron en dos partes: set de entrenamiento (80%) y set de prueba (20%)

IV. Evaluación experimental

2. Procesamiento previo de datos

- Eliminación de datos sucios
- Normalización: Para manejar la gran diferencia de rango de valores
- Segmentación basada en ventanas deslizantes superpuestas

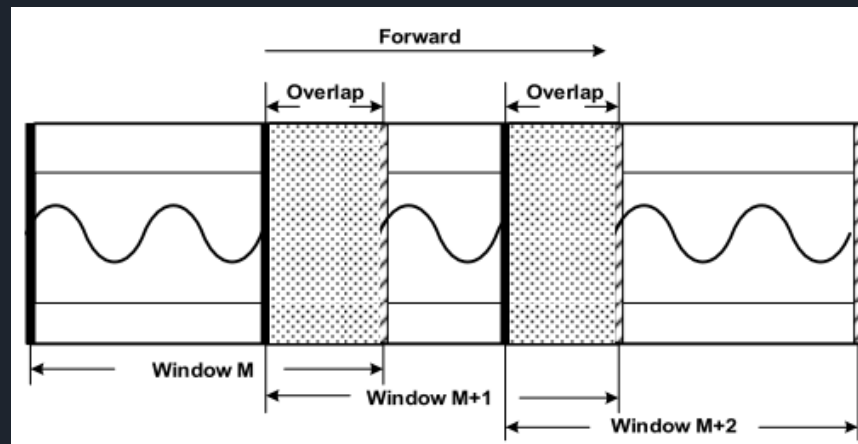


Figura 7: Segmentación de datos de sensores basada en ventana deslizante.

IV. Evaluación experimental

3. Configuración experimental

- Optimizador Adam
- La función de pérdida de entropía cruzada se aprovecha con una tasa de aprendizaje: $\text{rate} = 0,001$, $\text{beta1} = 0,9$, $\text{beta2} = 0,999$
- Epoch de 100 y batch se establece en 1024

IV. Evaluación experimental

4. Resultados

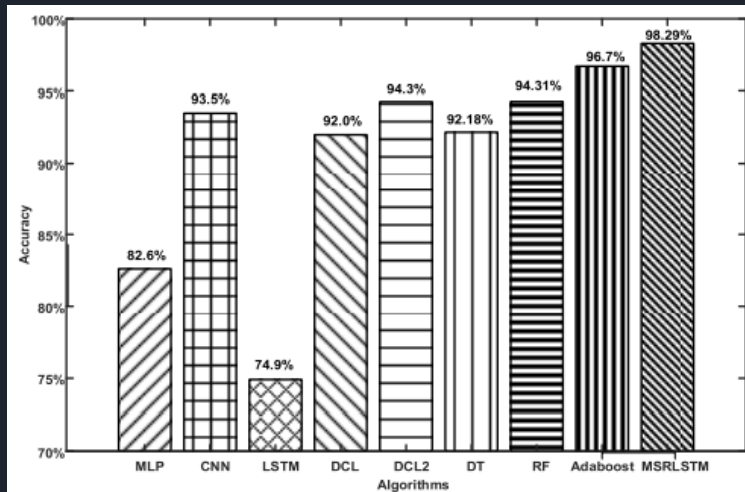


Figura 8: Resultados SHL.

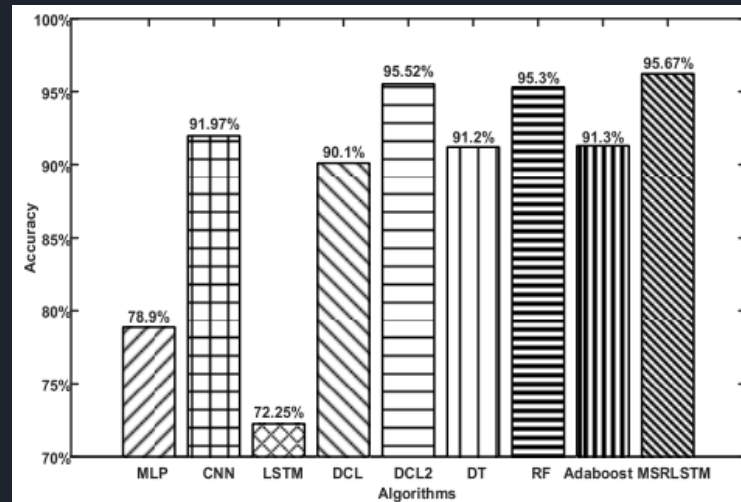


Figura 9: Resultados HTC.

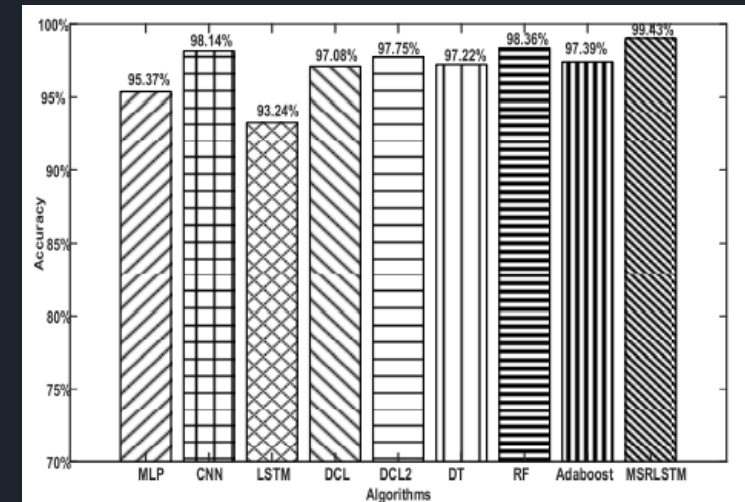


Figura 10: Resultados ICT.

IV. Evaluación experimental

4. Resultados

	MLP	CNN	LSTM	DCL	DCL2	DT	RF	Adaboost	MSRLSTM
Still	88.60%	93.90%	79.90%	94.50%	94.24%	97.03%	97.84%	97.50%	98.64%
Walk	90.60%	96.00%	94.50%	96.80%	96.62%	97.51%	99.76%	99.50%	97.82%
Run	97.60%	98.70%	98.60%	98.90%	99.78%	99.31%	98.95%	99.90%	99.55%
Bike	91.80%	95.90%	88.40%	96.00%	98.16%	99.55%	99.43%	99.90%	99.03%
Car	87.00%	97.30%	69.60%	96.10%	97.57%	96.74%	99.71%	99.10%	99.52%
Bus	75.60%	93.20%	65.50%	92.50%	95.54%	99.77%	93.60%	99.90%	98.49%
Train	68.60%	98.40%	58.50%	82.80%	88.36%	60.87%	70.73%	79.90%	97.04%
Subway	61.40%	61.40%	46.40%	78.30%	84.23%	74.91%	83.72%	80.60%	96.08%

Tabla 1: Precisión diferentes modos de transporte con diferentes algoritmos en el conjunto de datos SHL.

IV. Evaluación experimental

4. Resultados: Importancia esquema de atención

	Precision	Recall	F1-Score
Still	98.64%	98.46%	98.55%
Walk	97.82%	98.69%	98.25%
Run	99.55%	99.61%	99.58%
Bike	99.03%	98.34%	98.69%
Car	99.52%	99.28%	99.40%
Bus	98.49%	98.46%	98.47%
Train	97.04%	97.17%	97.11%
Subway	96.08%	96.15%	96.12%

Tabla 2: Métricas de MSRLSTM con esquema de atención en datos SHL



	Precision	Recall	F1-Score
Still	97.47%	98.29%	97.87%
Walk	98.25%	98.11%	98.18%
Run	99.66%	99.60%	99.63%
Bike	98.95%	98.05%	98.50%
Car	99.44%	99.27%	99.35%
Bus	97.15%	98.78%	97.96%
Train	96.77%	96.14%	96.45%
Subway	95.51%	95.00%	95.26%

Tabla 3: Métricas de MSRLSTM sin esquema de atención en datos SHL

IV. Evaluación experimental

4. Resultados: Importancia parámetros en la ventana deslizante

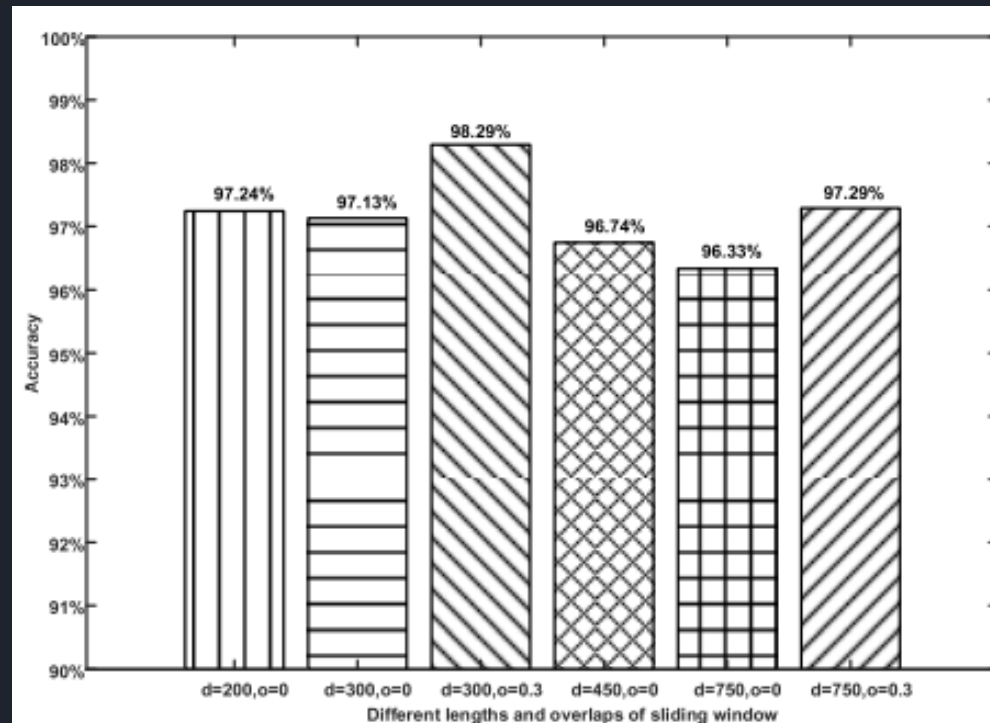


Figura 11: Precisión de diferentes longitudes y relaciones de superposición de la ventana deslizante utilizando el modelo MSRLSTM en el conjunto de datos SHL

IV. Evaluación experimental

4. Resultados: Importancia redes residuales

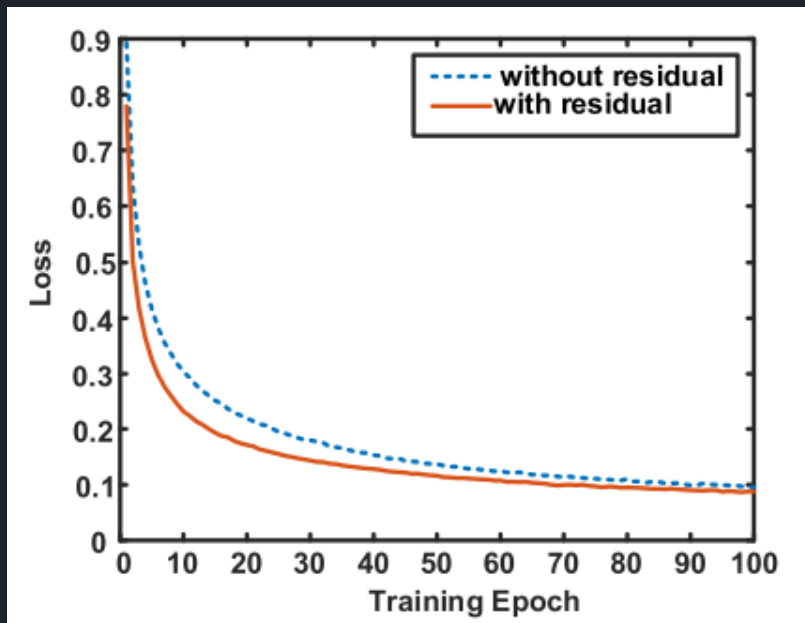


Figura 12: El valor de la pérdida de entrenamiento con diferentes épocas durante el entrenamiento

Algorithm	Platform Type	Training Time	Predicting Time
MLP	GPU	1937s	0.11ms
CNN	GPU	19315s	0.22ms
LSTM	GPU	23467s	2.68ms
DCL	GPU	52132s	4.58ms
DCL2	GPU	49012s	10.5ms
DT	CPU	541.7s	0.04ms
RF	CPU	895.35s	0.07ms
Adaboost	CPU	7292s	121.7ms
MSRLSTM	GPU	37890s	0.68ms

Tabla 4: Tiempo de entrenamiento y testeo en diferentes algoritmos con el conjunto de datos SHL

IV. Evaluación experimental

5. Escalabilidad y generalización de los resultados

- La arquitectura de aprendizaje de características independientes del modelo MSRLSTM puede adaptarse fácilmente a tipos flexibles de datos de entrada
- La precisión del modelo se mejora aproximadamente entre un 3% y un 5% al agregar un sensor. Por lo que se puede aprender más funciones cuando se utilizan más variables de sensor, especialmente el sensor de presión barométrica
- Generalización: Conjunto de datos de TIC obtuvo resultado que superan en todos los modos de transporte el 95% y la precisión total fue del 97,92%



¡MUCHAS GRACIAS!

