

# Analiză comparativă a arhitecturilor LSTM, GRU și Transformer pentru predicția consumului de energie electrică

Ana-Maria Rusu  
Andrei-George Drudun  
Bogdan-Valeriu Țicu

5 Februarie 2026

# Introducere și Obiective

## 1. Contextul problemei

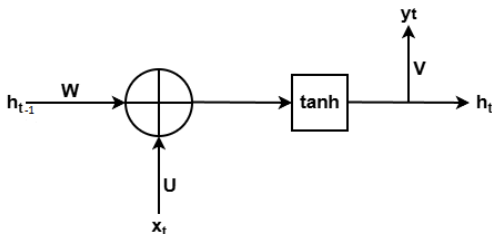
- Miza: Echilibrul cerere-ofertă în Smart Grids.
- Provocarea: Date non-staționare și haotice; energia nu poate fi stocată eficient.
- Soluția: Tranziția de la metode statistice (ARIMA) la Deep Learning.

## 2. Obiectivul studiului

- Analiză comparativă: Arhitecturi (RNN) vs. Transformer.
- Evaluarea compromisului: Acuratețe predicție vs. Cost computațional.

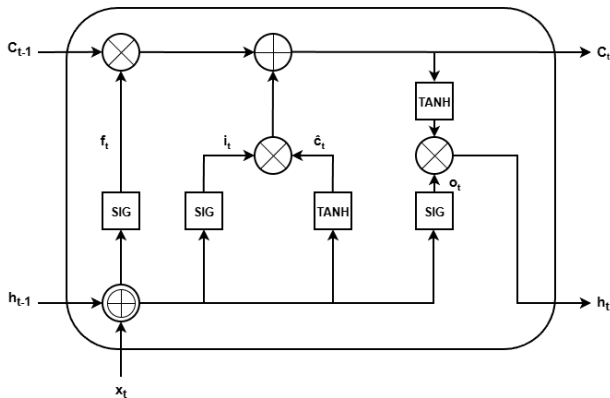
## 3. Metode propuse

- LSTM: Gestionarea dependențelor pe termen lung.
- GRU: Eficiență și viteză sporită de antrenare.
- Transformer: Procesare paralelă prin mecanisme de Self-Attention.

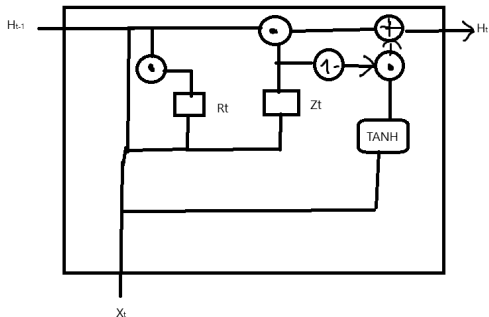


- Rețea neuronală pentru date secvențiale
- Unitate fundamentală  $\rightarrow$  unitatea recurentă
- Utilizează memorie internă (stare ascunsă)  $\rightarrow$  actualizată la fiecare pas
- Antrenare cu Backpropagation Through Time (BPTT)
- Nu reține dependențe lungi  $\rightarrow$  dispariția / explozia gradientului

# LSTM



- Tip de RNN, proiectat pentru secvențe lungi
- Celula LSTM menține 2 stări: starea ascunsă + starea celulei
- Controlează stările prin 3 porți: de uitare, de intrare, de ieșire



- Utilizează doar două porți (actualizare și resetare) și o stare ascunsă unificată.
- Spre deosebire de LSTM, nu menține o stare a celulei separată, ci un singur vector de stare ascunsă ( $h_t$ ).
- Poarta de actualizare decide câtă informație din trecut se păstrează, iar cea de resetare câtă informație se ignoră.

# Transformer - Diagrama Arhitecturii

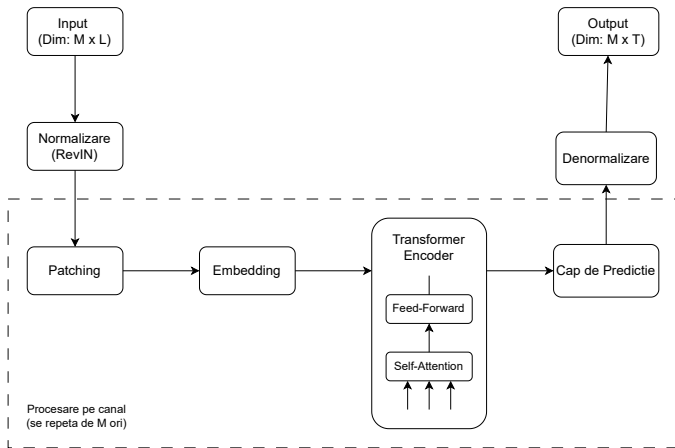


Figura: Arhitectura PatchTST

# Transformer

- **RevIN: Reversible Instance Normalization**

- Normalizare:  $\hat{x}_t = \gamma \odot \left( \frac{x_t - \mu_x}{\sigma_x} \right) + \beta$
- Denormalizare:  $x_t = \sigma_x \odot \left( \frac{\hat{x}_t - \beta}{\gamma} \right) + \mu_x$
- Parametrii  $\gamma, \beta$  sunt învățabili

- **Patching: Agregă pași temporali adiacenți în patch-uri**

- Transformă  $x \in \mathbb{R}^{1 \times L}$  în  $X_p \in \mathbb{R}^{N \times P}$
- Numărul de patch-uri:  $N = \lfloor \frac{L-P}{S} \rfloor + 2$
- Complexitatea redusă de la  $O(L^2)$  la  $O(N^2)$

- **Proiecția Latentă:**

- Transformare liniară pe fiecare patch:  $X_d = X_p W_p + b_p$
- Din dimensiune  $P$  la  $D_{model}$  (ex.  $16 \rightarrow 256$ )

- **Positional Encoding: Adaugă informația de poziție și timp.**

- Sinusoide de frecvențe diferite:

$$PE_{(pos, 2i)} = \sin \left( \frac{pos}{10000^{2i/d_{model}}} \right), \quad PE_{(pos, 2i+1)} = \cos \left( \frac{pos}{10000^{2i/d_{model}}} \right) \quad (1)$$

- Integrare:  $X_{final} = X_d + PE$

- **Scaled Dot-Product Attention:**

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

- Query:  $Q = X_{final} \cdot W^Q$  – "ce caut?"
- Key:  $K = X_{final} \cdot W^K$  – "cum mă găsești?"
- Value:  $V = X_{final} \cdot W^V$  – "informația relevantă"

- **Multi-Head Attention:**

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \cdot W^O$$

- $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

- **Position-wise Feed-Forward Network:**

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

- **Conexiuni Reziduale:**

$$\text{Output} = \text{Layer}(x) + x$$

- **Layer Normalization:** Normalizare pe dimensiunea caracteristicilor



# Experimente

- Metrice folosite: MSE, MAE, MAPE,  $R^2$
- 6 experimente (configurații)
- Orizont fixat ( $h$ )
  - $h = 48$ ,  $\text{win\_len} = 168$
  - $h = 48$ ,  $\text{win\_len} = 336$
  - $h = 48$ ,  $\text{win\_len} = 504$
- Fereastră fixată ( $\text{win\_len}$ )
  - $\text{win\_len} = 336$ ,  $h = 24$
  - $\text{win\_len} = 336$ ,  $h = 48$
  - $\text{win\_len} = 336$ ,  $h = 72$

# Rezultate

## LSTM

$h$	$w = 336$			
	MSE	MAE	MAPE	$R^2$
24	0.547	0.513	3.49	0.910
48	0.822	0.638	4.34	0.865
72	1.202	0.791	5.35	0.801

$w$	$h = 48$			
	MSE	MAE	MAPE	$R^2$
168	0.858	0.659	4.47	0.858
336	0.822	0.638	4.34	0.865
504	0.843	0.654	4.48	0.861

## GRU

$h$	$w = 336$			
	MSE	MAE	MAPE	$R^2$
24	0.492	0.495	3.38	0.919
48	0.798	0.621	4.23	0.869
72	1.251	0.798	5.41	0.793

$w$	$h = 48$			
	MSE	MAE	MAPE	$R^2$
168	0.852	0.637	4.33	0.859
336	0.798	0.621	4.23	0.869
504	0.845	0.645	4.37	0.860

## Transformer

$h$	$w = 336$			
	MSE	MAE	MAPE	$R^2$
24	0.649	0.590	4.02	0.894
48	0.955	0.706	4.79	0.843
72	1.402	0.861	5.78	0.768

$w$	$h = 48$			
	MSE	MAE	MAPE	$R^2$
168	0.933	0.686	4.61	0.846
336	0.955	0.706	4.79	0.843
504	1.034	0.737	4.99	0.830

# Comparații & concluzii

- Creșterea orizontului de predicție  $\Rightarrow$  degradarea performanței
- LSTM și GRU sunt sensibile la dimensiunea ferestrei
- Transformer-ul se degradează odată cu creșterea ferestrei
- LSTM și GRU performanțe similare, domină Transformer-ul
- Transformer-ul poate fi îmbunătățit  $\Rightarrow$  mai multe date, optimizare
- GRU potrivit pentru predicții pe termen scurt, ușor instabil
- LSTM potrivit pentru predicții pe termen mai lung, mult mai stabil
- Cost computațional:  $\text{LSTM} > \text{GRU} > \text{Transformer}$