

Báo cáo Nghiên cứu Paper SOTA

Nguyễn Nam Khánh

1/4/2025

1 Giới thiệu về bài báo 2 Tóm tắt nghiên cứu của bài báo

Bài báo: **DeepHealth: A Self-Attention Based Method for Instant Intelligent Predictive Maintenance in Industrial Internet of Things**

Tác giả: *Weiting Zhang, Student Member, IEEE, Dong Yang, Member, IEEE, Youzhi Xu, Xuefeng Huang, Jun Zhang, and Mikael Gidlund, Senior Member, IEEE*

Bài báo được xuất bản vào tháng 8/2021 trên *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 8, August 2021.

Mục đích của bài báo là nghiên cứu về sự bền vững của thiết bị IoT trong công nghiệp bằng cách xây dựng mô hình AI để dự đoán và đánh giá tình trạng của thiết bị. Việc này rất quan trọng trong bối cảnh công nghiệp hóa, hiện đại hóa, khi mà các thiết bị ngày càng được tự động hóa cao nhưng cũng dễ gặp lỗi hơn. Do đó, bài báo hướng đến việc xây dựng mô hình AI có thể dự đoán và đánh giá khả năng gặp lỗi của thiết bị trong tương lai.

Trong báo cáo này, em sẽ xây dựng lại mô hình của bài báo bằng phiên bản Python và TensorFlow mới nhất, đồng thời tìm cách cải thiện hiệu suất của mô hình.

Bài báo sử dụng một máy rửa tự động (Automatic Washing Machine) làm thiết bị thí nghiệm.

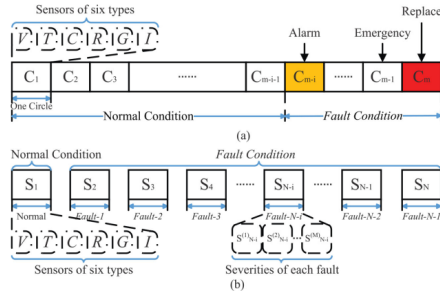
Dữ liệu được lấy từ hai tập dữ liệu: AWE và CRWU. Tuy nhiên, do đường dẫn tải tập dữ liệu CRWU bị xóa, em chỉ có thể sử dụng tập dữ liệu AWE.

Tập dữ liệu AWE được chia thành 7 tệp khác nhau, trong đó:

- 6 tệp thể hiện tình trạng của thiết bị qua cảm biến.
- 5 tệp tương ứng với 5 lỗi khác nhau.
- 1 tệp thể hiện tình trạng bình thường.

Tác giả đề cập đến hai phương pháp thu thập dữ liệu:

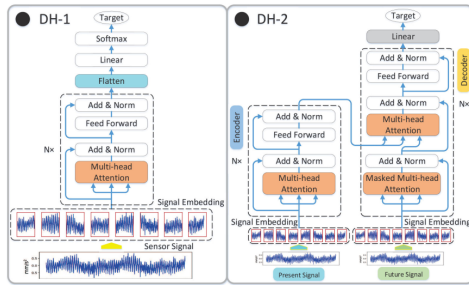
1. **Run to Fail:** Là phương pháp phổ biến nhưng có thể gây mất cân bằng dữ liệu, làm khó khăn trong bước tiền xử lý và xây dựng mô hình.
2. **Destructive Scheme:** Tạo lỗi nhân tạo bằng cách thêm các chips vào trục quay để làm lệch trục quay, giúp tạo ra tập dữ liệu cân bằng hơn và dễ dàng xây dựng mô hình.



Hình 1: Mô tả 2 phương pháp trích xuất data ((a) Run to fail, (b) Destructive Scheme)

3 Mô hình dự đoán

Bài báo xây dựng mô hình dự đoán dựa trên hai mô hình con DH-1 và DH-2:



Hình 2: Cấu trúc của 2 sub model

3.1 Mô hình DH-1

- Chức năng: Phân loại để xác định tình trạng của thiết bị.
- Đầu vào: Chuỗi tín hiệu từ cảm biến.
- Quá trình:

1. Dữ liệu được chia thành các khối nhỏ.
2. Qua lớp Attention: giúp mô hình tập trung vào nhiều vùng dữ liệu

khác nhau của đầu vào, giúp mô hình hóa ngữ cảnh đối với chuỗi dữ liệu thời gian.

3. Qua lớp Feed Forward: giúp mô hình học được các biểu diễn phức tạp hơn của dữ liệu.
4. Tiếp tục qua bước Flatten, Linear.
5. Cuối cùng là Softmax để phân lớp tình trạng thiết bị.

3.2 Mô hình DH-2

- Chức năng: Dự đoán tín hiệu tiếp theo của cảm biến.
- Cấu trúc tương tự DH-1 nhưng có thêm bước Masked Attention, đảm bảo mô hình chỉ nhìn thấy thông tin từ quá khứ, không thể nhìn được trước giá trị trong tương lai, giúp mô hình tạo ra chuỗi theo từng bước.
- Dữ liệu dự đoán từ DH-2 sẽ được sử dụng làm đầu vào cho DH-1 để dự đoán tình trạng tương lai của thiết bị.

4 Kết quả nghiên cứu

4.1 Kết quả từ bài báo

TABLE 1
DESCRIPTION OF THE TRAINING, EVALUATING, AND TESTING PROCESS OF DEEPHEALTH

Dataset	Sequence length	Data volume	Health Perception (DH-1)					Sequence Prediction (DH-2)				
			Time consumption		Metrics (Testing)			Dataset descriptions		Time consumption		
			Train	Test	Accuracy	Precision	Recall	F1-score	Train/Val/Test	Total volume	Train	Test
Our (AWE)	128	187500	5980.5 s	1.292 s	69.52%	70.30%	69.52%	69.54%	172500/7499/7499	187498	5594.7 s	1.719 s
	256	93750	4505.9 s	1.165 s	95.87%	95.80%	95.87%	95.86%	84374/4687/4687	93748	6971.1 s	1.573 s
	512	46872	4109.6 s	0.983 s	98.78%	98.77%	98.78%	98.77%	42184/2343/2343	46870	5565.4 s	1.371 s
	1024	23436	4019.6 s	0.887 s	98.81%	98.80%	98.80%	98.80%	21092/1171/1171	23434	5602.2 s	1.275 s
	2048	11718	11874.8 s	1.047 s	98.29%	98.29%	98.29%	98.28%	10546/585/585	11716	17589.8 s	1.473 s
CWRU	4096	5859	21606.2 s	1.238 s	91.96%	91.95%	91.95%	91.96%	5270/292/292	5854	25411.1 s	1.794 s
	128	37760	1484.1 s	0.450 s	81.25%	81.45%	81.25%	81.26%	33984/1887/1887	37758	2520.9 s	0.774 s
	256	18880	906.2 s	0.352 s	85.80%	86.40%	85.80%	85.85%	16992/943/943	18878	1285.9 s	0.660 s
	512	16100	1399.2 s	0.441 s	93.41%	93.61%	93.41%	93.38%	8496/471/471	9438	1093.9 s	0.650 s
	1024	16080	2801.7 s	0.672 s	98.32%	98.32%	98.32%	98.32%	4248/235/235	4718	1149.6 s	0.615 s
	2048	16050	15816.8 s	1.253 s	99.56%	99.57%	99.56%	99.56%	2124/1171/1171	2358	6122.7 s	0.647 s
	4096	15980	55860.2 s	3.253 s	99.88%	99.88%	99.87%	99.87%	1062/58/58	1178	10533.9 s	0.704 s

Hình 3: Kết quả của model do bài báo công bố

Đối chiếu với Hình 4, có thể thấy model do em code lại có sự cải thiện nhẹ về thời gian train và test nhưng cũng có sự sụt giảm nhẹ về accuracy.

4.2 Kết quả thực nghiệm của em

	DH-1				LSTM		
	Accuracy	Training time	Testing time	Accuracy	Training time	Testing time	
1	97,396	1640,839	0,652	99	651,075	0,158	
2	97,951	1646,501	0,512	98,76	543,246	0,153	
3	97,183	1598,725	0,492	98,76	509,333	0,156	
4	97,354	1616,689	0,503	98,46	530,845	0,157	
5	97,567	1813,383	0,51	99,19	628,995	0,182	

Length of input: 1024

Length of input: 1024

Hình 4: Bảng so sánh hiệu suất của LSTM và DH-1

- Kết quả chạy lại code không đạt độ chính xác bằng bài báo, thường kém khoảng 1% accuracy (Hình 4).
- Mô hình truyền thống LSTM lại cho thấy sự ổn định hơn và có độ chính xác nhỉnh hơn một chút so với DH-1.
- LSTM có thời gian train nhanh hơn gấp 3 lần và thời gian test nhanh hơn gần 5 lần so với mô hình DH-1.
- Điều này cho thấy LSTM vượt trội hơn về thời gian phản hồi so với mô hình của bài báo.
- Kết quả MAE và RMSE trên tập AWE không được bài báo công bố chính xác, nhưng với model DH-2 của em cũng đã đưa ra được MAE và RMSE khá tốt.
- Đối với model DH-2 em đã thử chạy trên 2 framework khác nhau để so sánh độ hiệu quả (Hình 5 dòng màu đỏ tương ứng với TensorFlow, màu đen tương ứng với Pytorch).

tương ứng với Pytorch). Ở bảng có thể thấy TensorFlow chạy chậm hơn so với Pytorch nhưng đều cho ra kết quả không quá khác biệt, vậy nên đối với model cuối em quyết định sử dụng TensorFlow để có thể đồng bộ load model và xây dựng được model cuối IPdM như bài báo nhắc tới.

	RMSE	MAE	Training time	Testing time
1	0.0236	0.0193	5024.338	0.591
2	0.0242	0.0198	4257.084	0.725
3	0.0277	0.185	3957.851	0.395
4	0.0248	0.0203	4196.397	0.441
5	0.0239	0.0196	4599.419	0.557
6	0.0241	0.0194	7333.717	2.82

Hình 5: Thống kê hiệu suất model DH-2

5 Kết luận

Bài báo đưa ra một phương pháp mới dựa trên Self-Attention để dự đoán lỗi trong thiết bị IoT công nghiệp. Tuy nhiên, qua thử nghiệm của em, mô hình truyền thống LSTM vẫn có ưu thế về tốc độ và độ chính xác trong một số trường hợp. Trong tương lai, cần nghiên cứu thêm các cách cải thiện hiệu suất của DH-1 và DH-2 để đạt kết quả tốt hơn. Đồng thời tìm ra cách cải thiện cấu trúc của model truyền thống để thay thế submodel DH-1 phức tạp trên.