



ConBRepro

XV CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO

03 a 05 de dezembro de 2025

As Engenharias na Transformação Digital das PMEs

Previsão de tempo remanescente de ordens de produção: uma abordagem híbrida

Wesley Gabriel de Lacerda Souza Alves

PPGMNE - UFPR

Eduardo Alves Portela

PPGMNE - UFPR

Resumo: Prever o tempo remanescente de ordens de produção permite que os gestores acompanhem desvios no plano de produção, facilitando a tomada de decisões em tempo real proativamente. Esse trabalho propõe um modelo que integra mineração de processos e redes neurais de aprendizagem para estimar o tempo remanescente por etapa de produção. A partir de logs de eventos reais é construído matrizes de treinamento que representam a sequência de máquinas e a quantidade de produtos a serem fabricados, diante desse panorama utiliza-se como alvo o tempo remanescente por unidade. Avalia-se duas estratégias (i) modelos MLP treinados por produto e (ii) um modelo MLP universal treinado com os dados de todos os produtos. As métricas de avaliação utilizadas foram MSE e MAE. Nos experimentos com 12 produtos, o modelo universal obteve o melhor desempenho, principalmente em produtos com poucas ordens de produção. A abordagem mostra potencial para suportar decisões em tempo real em ambientes de produção.

Palavras-chave: Previsão do tempo remanescente, mineração de processos, monitoramento de processos, Redes neurais de Aprendizado, Indústria 4.0.

Prediction of remaining time of orders: an hybrid approach

Abstract: Predicting the remaining time of production orders enables managers to monitor deviations from the production plan and to make proactive real-time decisions. In this work we propose a model that integrates process mining and neural network models to estimate the remaining time at each production stage. From real event logs we build training matrices that represent the sequence of machines and the quantity of products to be manufactured; in this setting we use the remaining time per unit as the target. We evaluate two strategies: (i) MLP models trained per product, and (ii) a universal MLP model trained on data from all products. The evaluation metrics used were MSE and MAE. In experiments with 12 products, the universal model achieved the best performance, particularly for products with few production orders. The approach shows potential to support real-time decision making in production environments

Keywords: Remaining time prediction, process mining, process monitoring, Artificial Neural Networks, Industry 4.0.

1. Introdução

Monitorar o andamento das ordens de produção é essencial para equilibrar as discrepâncias entre o cronograma original e o processo produtivo real (FANG et al., 2020; SERRANO-RUIZ et al., 2024). Prever o tempo restante para as ordens de produção permite que os gestores acompanhem os desvios no plano de produção, facilitando a tomada de decisões em tempo real para ajustes no agendamento. Idealmente, o plano de produção original estaria perfeitamente alinhado com o processo de manufatura (MÜLLER-ZHANG et al., 2023; CHENG et al., 2018). No entanto, esse alinhamento é frequentemente interrompido por eventos inesperados, como falhas de máquinas (ALVES et al., 2022), a chegada de pedidos urgentes ou volumes significativos de produtos em processo defeituosos, os quais levam a flutuações substanciais no tempo restante das ordens.

As técnicas de mineração de processos têm se destacado ao possibilitar a análise e descoberta de processos com base em logs de eventos (VAN DER AALST, 2016). Utilizando essas informações, algoritmos de descoberta podem gerar um modelo do processo. Com o modelo em mãos, análises adicionais tornam-se possíveis — por exemplo, verificar a conformidade com modelos predefinidos, identificar gargalos, e explorar possíveis extensões e inferências do processo para melhorias futuras (VAN DER AALST et al., 2012; LORENZ et al., 2021). O monitoramento preditivo de processos é uma família de técnicas que utiliza logs de eventos para prever o estado de uma execução do processo (RIZZI et al., 2020). O uso da mineração de processos para prever e monitorar tempos restantes em processos é uma tendência atual (VERENICH et al., 2019b).

Modelos preditivos menos robustos podem sofrer com a premissa de mudanças, no tempo de conclusão de um ciclo de produção, tais mudanças podem estar relacionadas a fatores internos como a degradação do desempenho de máquinas, sendo um bom indicador da necessidade de manutenção. Assim, este artigo propõe um modelo para a previsão do tempo remanescente, capaz de lidar com essas sutilezas. A mineração de processos será utilizada para extrair logs de eventos dos sistemas de informação de manufatura. A partir desses logs, serão extraídas características e tempos de ciclo de produção para servir o modelo de rede neural artificial.

2. Referencial Teórico

2.1 Mineração de processos

Segundo (VAN DER AALST, 2016) a mineração de processos compreende um conjunto de técnicas para a descoberta, verificação de conformidade e melhorias utilizando o logs de eventos. Um log de eventos é composto por eventos, cada evento possui um identificador de caso (Case ID), o nome da atividade (Activity) e as marcações temporais do início ou fim da atividade (Timestamp), podendo conter informações adicionais, como os recursos envolvidos, quantidade de peças a serem produzidas (dentro de um contexto industrial), dentre outras.

A descoberta de processos a partir de logs, torna-se a matéria prima que combinada com algoritmos de descobertas pode reconstruir o modelo real. Sendo possível averiguar a conformidade de modelos pré-definidos, identificar gargalos, além de explorar extensões e inferências para melhorias futuras. Dentro do contexto de predição do tempo remanescente, a mineração de processos pode se tornar um alicerce, pois suas técnicas permitem a extração de informações dos registros de eventos para modelos preditivos, como é realizado no trabalho de (CHOUEIRI et al., 2020; BOTELHO et al., 2025). No capítulo de metodologia vamos descrever um pouco mais sobre como os Logs são apresentados e como os convertemos em features para o modelo de rede neural.

2.2 Predição do tempo remanescente

Dentro do contexto industrial, existem trabalhos direcionados para a predição de processos, como pode ser observado em (EVERMANN et al., 2017) e (NEU et al., 2022), porém começou-se a ter uma preocupação não só com a previsão do próximo processo mas também com o tempo envolvido neles, a literatura de (WANG et al., 2018) e (MEIDAN et al., 2011) apresentam métodos para prever o cycle time - tempo total de produção - utilizando técnicas de machine learning.

O tempo de ciclo não é apenas um valor de tempo, ele também pode ser visto como um indicador do processo fabril, entender como ele funciona ajuda os gestores a cumprirem metas, agendar recursos, averiguar falhas, etc. Por outro lado um outro tipo de predição temporal surge quando se trata de produção, essa é conhecida como a predição do tempo remanescente, como mostra (CECI et al., 2014) é possível usar a mineração de processos para construir um modelo de treinamento capaz de aprender o tempo remanescente das ordens de produção de um processo fabril.

Ao analisar o trabalho de (CHOUEIRI et al, 2020) depara-se com uma boa abordagem que ao invés de olhar para "árvores dos principais traces" como ocorre em (CECI et al., 2014), é construído uma matriz que é capturada na etapa de produção dos produtos analisados e posteriormente pode ser usada para o treinamento de um modelo de predição. Esse método evolui mais ainda com o trabalho de (BOTELHO et al., 2025) que adapta essa representação matricial para treinar um modelo mais robusto de predição do tempo remanescente. O presente trabalho será inspirado nas literaturas de (CHOUEIRI et al, 2020) e (BOTELHO et al., 2025). Será utilizado as técnicas apresentadas para desenvolver um modelo de Rede Neural Artificial (RNA) que consiga aprender como funciona o tempo remanescente dentro de uma fábrica utilizando seu log de eventos.

3. Metodologia

O método proposto emprega mineração de processos como técnica básica para predição do tempo remanescente. Na terminologia de mineração de processos, um evento é definido por diversos atributos, tais como timestamp, um recurso que identifica o executor, custos associados, entre outros, (VAN DER AALST, 2016). Cada evento deve pertencer a um caso específico. Quando os eventos de um caso são ordenados cronologicamente, formam uma trilha (trace): uma sequência finita e não vazia de eventos, em que cada evento ocorre apenas uma vez e os timestamps são não decrescentes. É importante notar que múltiplos casos podem compartilhar o mesmo trace, embora cada caso permaneça único. Um log de eventos é uma coleção de traces. A priori, qualquer processo com dimensão temporal pode ser representado como um log de eventos, incluindo atividades de manufatura. A Tabela 1 mostra um fragmento de um log de eventos relacionado a um sistema industrial de manufatura (LEVY, 2014).

Tabela 1 – Exemplo de um log de eventos de manufatura industrial

Case ID	Activity	Start Timestamp	Complete Timestamp
Case 1	Turning & Milling - Machine 4	2012/01/30 06:59:00.000	2012/01/30 07:21:00.000
Case 1	Turning & Milling - Machine 4	2012/01/30 07:21:00.000	2012/01/30 10:58:00.000
Case 1	Turning & Milling Q.C.	2012/01/31 13:20:00.000	2012/01/31 14:50:00.000
Case 1		2012/02/01 08:18:00.000	2012/02/01 08:27:00.000

Fonte: Adaptado de (Levy, 2014)

Para entender esse fragmento, primeiramente examina-se a coluna Case ID ela é um identificador único para cada ordem de produção, na primeira linha identificamos que o produto em fabricação estar passando pela máquina 4 na atividade *Turning & Milling* que se iniciou as 06h59min e finalizou as 07h21min. De forma análoga, cada linha seguindo a ordem de cima para baixo traz informações sobre cada etapa de fabricação do produto em questão.

3.1 Construção da matriz de treinamento

Constata-se que o log registra informações como ID do caso, tempo e atividades. Com base nesses dados, realiza-se a descoberta de processo para entender o fluxo da linha de produção. Nesta etapa, durante a mineração de processos, os produtos são segregados de acordo com suas características ou ordem de produção, formando um sistema de transição. para calcular o tempo remanescente por produto para o i -ésimo estágio de produção, utiliza-se a fórmula:

$$\tau_i = \frac{T - t_i}{Q} \quad (4.1)$$

Sendo t_i o tempo acumulado desde o primeiro até o i -ésimo estágio de produção, Q a quantidade de itens requerida na ordem de produção e T o tempo total (do início ao fim da produção, incluindo os intervalos entre uma máquina e outra). A equação (4.1) foi inspirada no trabalho de (CHOUEIRI et al, 2020), assim como o fragmento do log sintético da tabela 2 que será utilizada para ilustrar a construção da matriz de treinamento.

Tabela 2 – Fragmento do log sintético

Order ID	T_s	T_f	Product	Machine	Qty
01	28-02-2019: 11:05	28-02-2019: 11:50	A	1	20
01	28-02-2019: 12:10	28-02-2019: 12:30	A	2	20
01	28-02-2019: 12:35	28-02-2019: 13:55	A	3	20
02	28-02-2019: 14:00	28-02-2019: 16:30	B	1	30
02	28-02-2019: 17:00	28-02-2019: 19:00	B	2	30
02	28-02-2019: 19:30	28-02-2019: 22:30	B	3	30
02	28-02-2019: 22:30	29-02-2019: 02:30	B	4	30
03	29-02-2019: 03:00	29-02-2019: 03:22	A	1	10
03	29-02-2019: 03:25	29-02-2019: 03:50	A	2	10
03	29-02-2019: 03:55	29-02-2019: 04:30	A	3	10
04	29-02-2019: 05:00	29-02-2019: 05:20	B	1	5
04	29-02-2019: 05:22	29-02-2019: 05:50	B	2	5
04	29-02-2019: 05:55	29-02-2019: 18:25	B	3	5
04	29-02-2019: 18:25	29-02-2019: 18:45	B	5	5

Fonte: Adaptado de (Choueiri et al. (2020))

O fragmento da tabela 2 ilustra a produção de produtos (A e B), que passam por cinco máquinas denominadas (1, 2, 3, 4 e 5), Qty representa a quantidade de peças a serem produzidas, T_s e T_f o Timestamp de início e conclusão, respectivamente. Observe que dentro desse fragmento há quatro ordens de produção denominadas de *Order ID* (01,02,03 e 04), que seria equivalente ao Case ID. A próxima etapa consiste em segregar os produtos, assim o produto A é composto pelas (Order ID 1 e 3) e o produto B (Order ID 2 e 4). A tabela 3 mostra as ordens relacionadas ao produto A.

Tabela 3 – Fragmento do log sintético - produto A

Order ID	T_s	T_f	Product	Machine	Qty
01	28-02-2019: 11:05	28-02-2019: 11:50	A	1	20
01	28-02-2019: 12:10	28-02-2019: 12:30	A	2	20
01	28-02-2019: 12:35	28-02-2019: 13:55	A	3	20
03	29-02-2019: 03:00	29-02-2019: 03:22	A	1	10
03	29-02-2019: 03:25	29-02-2019: 03:50	A	2	10
03	29-02-2019: 03:55	29-02-2019: 04:30	A	3	10

Fonte: Adaptado de (Choueiri et al. (2020))

A seguir, para cada case ID, começando pela Order ID 01, é calculado os tempos remanescentes por produto de cada etapa, dado pela equação (4.1):

$$\tau_1 = \frac{T-t_1}{Q} = \frac{170-0}{20} = 8,5$$

Realizando o mesmo raciocínio encontramos $\tau_2 = 5,25$ e $\tau_3 = 4,00$. Dessa forma cria-se a matriz de treinamento dada pela tabela 4

Tabela 4 – Matriz de treino do produto A - Case 01

M_1	M_2	M_3	M_4	M_5	Q	τ_i
1	0	0	0	0	20	8.50
0	1	0	0	0	20	5.25
0	0	1	0	0	20	4.00

Fonte: Autor

As colunas M_i com $i = 1, 2, 3, 4, 5$ representam as máquinas utilizadas na etapa de produção, utiliza-se os binários 1 ou 0 para afirmar se uma máquina foi ou não utilizada durante a etapa de produção, por exemplo, na primeira linha (primeira etapa de produção) o número 1 indica que uma máquina M_1 foi utilizada enquanto o 0 indica que as máquinas M_2 até M_5 não foram utilizadas. As últimas duas colunas são formadas pela quantidade de produtos $Q = 20$ a serem produzidos e o tempo remanescente 8,50. De forma análoga, forma-se outras linhas da tabela. O mesmo raciocínio de construção será aplicado no case ID (Order ID 03), como mostra a tabela 5.

Tabela 5 – Matriz de treino do produto A - Case 01

M_1	M_2	M_3	M_4	M_5	Q	τ_i
1	0	0	0	0	10	8.99
0	1	0	0	0	10	6.50
0	0	1	0	0	10	3.50

Fonte: Autor

Generalizando o raciocínio, para cada produto, agrupa-se os eventos por Case ID, e calcula-se τ_i via equação (4.1) criando uma matriz de treinamento

$$X \in R^{n.(k+2)}, \quad (4.2)$$

sendo n o número de estágios de produção e $k + 2$ é o número de colunas. Essas colunas incluem as k máquinas utilizadas, seguidas pela penúltima coluna de Q peças produzidas e na última coluna o tempo remanescente τ_i . Portanto:

$$X_{i,j} = \begin{cases} 1, & \text{se a etapa } i \text{ usa a máquina } j, \quad 1 \leq j \leq k, \\ 0, & \text{se a etapa } i \text{ não usa a máquina } j, \quad 1 \leq j \leq k, \\ Q, & \text{se } j = k + 1, \\ \tau_i, & \text{se } j = k + 2. \end{cases}$$

Alternativamente, podemos escrever na forma matricial:

$$X = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,k} & Q & \tau_1 \\ m_{2,1} & m_{2,2} & \dots & m_{2,k} & Q & \tau_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,2} & \dots & m_{n,k} & Q & \tau_n \end{bmatrix}$$

3.2 Modelo MLP - Multilayer Perceptron

Após vários testes com diferentes meta-heurísticas observadas na literatura de (SIQUEIRA, 2021), foi utilizado uma Rede Neural Artificial (RNA) para identificar os padrões das matrizes de treinamento e estimar o tempo remanescente. Para cada etapa i , as features $(m_{i,1}, m_{i,2}, \dots, m_{i,k}, Q)$ têm como alvo τ_i . Em outras palavras, será usado a sequência de maquinários e a quantidade de peças a serem produzidas na esperança de que o modelo aprenda a estimar o tempo remanescente.

Os testes foram realizados no log de eventos de sistema industrial de manufatura (LEVY, 2014), já apresentado anteriormente na tabela 1, após a construção das matrizes de treinamento, adotou-se uma divisão por cases para evitar o vazamento de informações entre treino e teste. Para cada produto, foi definido:

- **Conjunto de treinamento:** primeiros 80% das ordens de produção, dentro desse conjunto, os primeiros 90% serão utilizados para treino e os últimos 10% para realizar a validação do treino, por exemplo, no log da tabela 1 tem um produto chamado Cable Head, que possui 50 ordens de produção, as primeiras 40 (80% das ordens de produção) será dividida em 90% iniciais (36 para aprendizado) e 10% finais (4 para validação);
- **Conjunto de teste:** últimos 20% das ordens de produção, no exemplo anterior, utilizou-se 40 das 50 ordens de produção, restou apenas as 10 últimas para teste final que consiste na predição do tempo remanescente de cada uma delas.

Duas abordagens foram testadas, a primeira usa um modelo de rede neural para cada produto, já a segunda cria um modelo universal, treinado com todos os produtos. As métricas de avaliação utilizadas foram o Erro Quadrático médio (MSE) e o Erro Absoluto Médio (MAE).

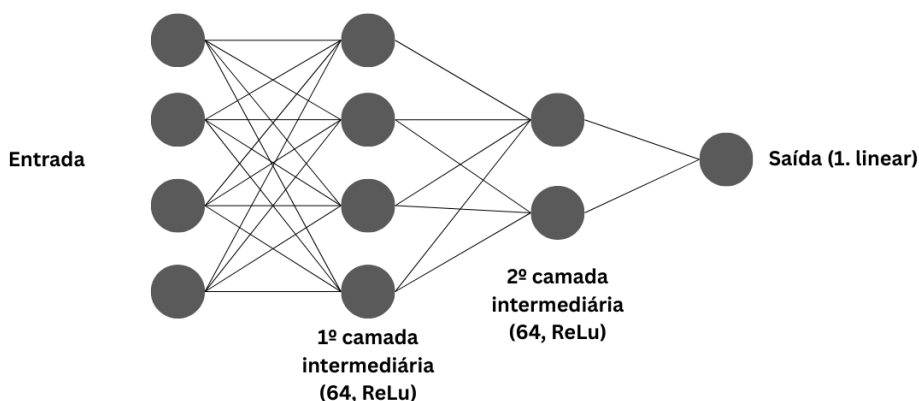
$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (4.3)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (4.4)$$

sendo y_i é o tempo remanescente real e $\hat{y}_i$ é a predição. O MAE está na mesma unidade de tempo do tempo remanescente, enquanto o MSE é sensível a outliers pois ao elevar ao quadrado, erros muito grandes, são penalizados.

A MLP possui uma entrada de $k + 1$ neurônios que passam por duas camadas intermediárias ocultas, a primeira com 64 neurônios e a segunda com 32 neurônios ambas com a ativação ReLU (Rectified Linear Unit) que transforma valores negativos em zero e mantém os positivos. Por fim passa por uma camada de saída de 1 neurônio com ativação linear. Utiliza-se como função de perda o MSE da equação (4.3), mas também utilizamos o MAE da equação (4.4) com uma métrica auxiliar, ela não influencia diretamente o modelo, mas será usada no fim para averiguar os erros ao longo do treino e testes. De forma análoga temos uma arquitetura semelhante à MLP universal, exceto pela camada de entrada que terá p neurônios a mais dado pela identificação one-hot dos p produtos que serão utilizados no modelo. A figura 1 ilustra essa arquitetura.

Figura 1 – Arquitetura MLP



Fonte: Autor

Os modelos foram implementados utilizando a linguagem de programação Python. As features foram padronizadas com o StandardScaler da biblioteca **scikit-learn**, o treinamento dos modelos foram realizados com a biblioteca **Tensorflow 2.19.0** (que implementa e treina redes neurais) em ambiente CPU (não foi utilizada GPU), utilizou-se o otimizador Adam (que ajuda a atualizar os pesos da rede durante o treino) com taxa de aprendizado padrão de 0,001, com *batch_size* = 32 e foi executado por até 200 épocas com *early stopping* de 8 épocas (o treinamento para após 8 épocas, se a função de perda não melhorar), monitorando a perda de validação quando possível.

4. Resultados e discussões

Os resultados são apresentados nas tabelas 6 e 7.

Tabela 6 – Desempenho por produto (MLP)

Produto	80% treino	20% teste	MSE treino	MAE treino	MSE teste	MAE teste
Cable Head	40	10	540,3772	8,4169	161,2374	10,4823
Spur Gear	20	6	60,7436	4,4154	18,3861	3,7623
Ballnut	46	12	308,3138	7,5007	11,8873	2,2082
Housing	7	2	26,6063	2,8755	6,5270	1,6526
Bearing	8	3	37,6441	4,3774	0,0689	0,1716
Clamp	2	1	76,9817	8,0901	15,3671	2,8067
Plug	4	2	16,9072	2,9208	830,5091	25,9194
Spring	2	1	0,0000	0,0027	0,1473	0,3468
Piston	4	1	40,6034	3,5033	0,1031	0,3211
Drill	2	1	13,9455	3,2702	0,0755	0,2425
Punch Holder	2	1	0,0136	0,0740	30,5040	5,4776
Gauge	3	1	93,0556	6,7876	43,7208	6,2313

Fonte: Autor

Tabela 7 – Desempenho da MLP global

Produto	80% treino	20% teste	MSE treino	MAE treino	MSE teste	MAE teste
Cable Head	40	10	453,6819	8,8319	503,6544	21,0955
Spur Gear	20	6	54,3852	5,1620	41,0574	4,9075
Ballnut	46	12	188,5070	7,1234	142,3721	10,0462
Housing	7	2	47,9386	4,7935	27,0033	2,2576
Bearing	8	3	13,3663	2,5012	34,3532	5,0973
Clamp	2	1	13,0989	3,1817	20,5098	3,4444
Plug	4	2	697,3004	24,7747	231,4172	13,4015
Spring	2	1	0,0001	0,0100	14,8782	3,8537
Piston	4	1	13,8330	3,0618	164,3005	12,8180
Drill	2	1	2,4811	1,0373	0,0412	0,1684
Punch Holder	2	1	1,4300	0,7917	1,2121	0,9446
Gauge	3	1	18,7255	3,8947	23,0642	4,0996

Fonte: Autor

Nota-se que alguns produtos como **Clamp, Spring, Drill, Punch Holder** e **Gauge** possuem poucas ordens de produção, e por essa razão não podemos tirar conclusões olhando somente pelo modelo MLP por produto. Poucas ordens de produção na MLP por produto, gera resultados que a priori parecem bons, mas existe a possibilidade do modelo ter apenas memorizado uma porção de tempo remanescente, por exemplo, no caso do produto **Clamp**, é utilizado 2 ordens de produção para treino e 1 para teste, nesse caso o modelo pode não ter dados o suficiente para capturar as nuances entre a sequência de máquinas utilizadas, quantia de peças a serem produzidas e tempo remanescente.

Por outro lado, poucas ordens de produção são interessantes no caso da MLP universal, pois nesse último caso será testado a capacidade que o modelo tem de generalizar as próximas ordens de produção com base em padrões observados na produção de todos os produtos. Por fim, durante os testes, foram considerados apenas os produtos com ordem de produção igual ou superior a 3.

Após uma análise, observa-se que o modelo de MLP por produto se saiu melhor em 8 dos 12 produtos avaliados, ou seja, em 2/3 dos casos. Diante desse panorama, a MLP por produto demonstrou maior capacidade de generalização, produzindo erros mais estáveis. Perceba que nos produtos com maiores ordens de produção (**cable Head, Spur Gear e Ballnut**) os resultados são bem superiores em relação ao MLP universal. Isso alimenta a hipótese de que a MLP por produto se sobressai, quando se tem uma base de dados maior a respeito da produção.

Em suma, o modelo é viável, pois a alternativa de MLP por produto parece criar especialistas que capturam boas variações, já o modelo MLP universal pode ser uma boa alternativa para produtos com poucos dados. Essa combinação é capaz de aprender padrões relevantes do tempo remanescente a partir dos logs reais, entregando previsões com bons erros.

5. Conclusões

Neste trabalho foi proposto e avaliado um modelo para a previsão do tempo remanescente de ordens de produção baseadas em técnicas de mineração de processos e rede neural de aprendizado (MLP). A metodologia consistiu em:

- extração e pré-processamento de logs;
- construção das matrizes de treinamento;
- treinamento da MLP por produto e universal;
- avaliação dos erros.

5.1 Limitações

Apesar dos resultados existem limitações relevantes, a forma como o tempo remanescente é construído supõe que o mesmo é analisado por peça, então existe a possibilidade do modelo não se comportar bem na hipótese de que uma fábrica utiliza processamentos paralelos, ou seja, quando diferentes produtos forem fabricados na mesma máquina durante um dado momento.

Além do que foi citado, alguns produtos apresentam poucas ordens de produção no Log, portanto qualquer interpretação deve ser realizada com cautela. Foi utilizado produtos com 3 ou mais ordens de produção para testar o modelo universal, mas o modelo por produto poderia de alguma forma retornar resultados subótimos, por conta dessas poucas ordens de produção.

5.2 Trabalhos futuros

É natural que ideias vão surgindo durante a construção de um trabalho, parte delas habitam o campo hipotético até o momento de publicação deste estudo. Sugerimos algumas ideias para trabalho futuros:

- **Engenharia de features:** incluir mais informações de cada ordem de produção, como os intervalos de descanso entre o uso das máquinas, recursos humanos e de maquinários envolvidos (pois um produto pode ser produzido por mais de um maquinário ou funcionário simultaneamente), indicadores de estado da máquina, dentre outros.
- **Testar outros modelos de redes neurais:** durante os estudos para o desenvolvimento da fundamentação teórica e metodologia, percebe-se o uso de diversos modelos, embora este trabalho seja uma inspiração da literatura de (CHOUEIRI et al. 2020) e (BOTELHO et al. 2025), não foram utilizados necessariamente todos os modelos contemplados nesses artigos. Pois focamos mais na Rede Neural Artificial do tipo MLP. No entanto em (SIQUEIRA, 2021) existem diversos tipos de modelos de aprendizado inspirados em redes neurais, dentre eles modelos de redes neurais temporais que talvez sejam ótimas aliadas na predição do tempo remanescente.
- **Cálculo do tempo remanescente:** criar ou adaptar uma nova equação de tempo remanescente, por exemplo, a equação 4.1 poderia ser adaptada como:

$$\tau_i = \frac{R(T-t_i)}{Q} \quad (4.5)$$

R seria os recursos envolvidos, que pode ser visto como uma grandeza diretamente proporcional a quantidade de peças produzidas, pois quanto mais máquinas/funcionários disponível, mais peças serão produzidas. portanto é possível usar a matemática básica, mais precisamente o conteúdo de grandezas diretamente ou inversamente proporcionais para criar novas fórmulas que talvez ajude a estabelecer uma relação entre as grandezas envolvidas e o tempo remanescente.

- **Testes com outros logs industriais:** realizar testes do modelo em diversos logs industriais ajuda a testar o modelo e obter melhorias no código ou metodologia.
- **Testes com logs que simulam drifts:** Uma lacuna notável na literatura é a consideração do conceito de drift que se refere a mudanças em um processo (VAN DER AALST, 2016), que podem afetar diversos aspectos, como a sequência (fluxo de controle) ou o tempo de ciclo de atividades específicas. Nota-se que modelos de RNA conseguem captar padrões nos dados, mas diante de um cenário de drift a predição do tempo remanescente é um fenômeno complexo que merece atenção.

Referências

- Alves, F. F., Nogueira, T. H. e Ravetti, M. G. (2022). Learning algorithms to deal with failures in production planning. *Computers & Industrial Engineering*, 169:108231.
- Botelho, J., Santos, E., Choueiri, A. e Junior, J. (2025). Remaining time prediction in manufacturing systems: an approach based on ml and process mining. *Procedia CIRP*, 132:165–170.
- Ceci, M., Lanotte, P., Fumarola, F., Cavallo, D. e Malerba, D. (2014). Completion time and next activity prediction of processes using sequential pattern mining. páginas 49–61.
- Cheng, Y., Chen, K., Sun, H., Zhang, Y. e Tao, F. (2018). Data and knowledge mining with big data towards smart production. *Journal of Industrial Information Integration*, 9:1–13.
- Choueiri, A. C., Sato, D. M. V., Scalabrin, E. E. e Santos, E. A. P. (2020). An extended model for remaining time prediction in manufacturing systems using process mining. *Journal of Manufacturing Systems*, 56:188–201.
- Evermann, J., Rehse, J.-R. e Fettke, P. (2017). A deep learning approach for predicting process behaviour at runtime. páginas 327–338.
- Fang, W., Guo, Y., Liao, W., Ramani, K. e Huang, S. (2020). Big data driven jobs remaining time prediction in discrete manufacturing system: a deep learning-based approach. *International Journal of Production Research*, 58(9):2751–2766.
- Levy, D. (2014). Production analysis with process mining technology.
- Lorenz, R., Senoner, J., Sihn, W. e Netland, T. (2021). Using process mining to improve productivity in make-to-stock manufacturing. *International Journal of Production Research*, 59(16):4869–4880.
- Meidan, Y., Lerner, B., Rabinowitz, G. e Hassoun, M. (2011). Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining. *Semiconductor Manufacturing, IEEE Transactions on*, 24:237 – 248.
- Müller-Zhang, Z., Kuhn, T. e Antonino, P. O. (2023). Towards live decision-making for service-based production: Integrated process planning and scheduling with digital twins and deep-q-learning. *Computers in Industry*, 149:103933.
- Neu, D., Lahann, J. e Fettke, P. (2022). A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artificial Intelligence Review*, 55:1–27.
- Rizzi, W., Di Francescomarino, C. e Maggi, F. M. (2020). Explainability in predictive process monitoring: when understanding helps improving. Em *International Conference on Business Process Management*, páginas 141–158. Springer.
- Serrano-Ruiz, J. C., Mula, J. e Poler, R. (2024). Job shop smart manufacturing scheduling by deep reinforcement learning. *Journal of Industrial Information Integration*, 38:100582.
- Siqueira, P. H. (2021). Metaheurísticas e aplicações. Repositório GitHub, Disponível em: <https://github.com/paulohscwb/metaheuristics>. Acesso em: 10 set. 2025.

van der Aalst, W. (2016). Data science in action. Springer.

Van der Aalst, W., Adriansyah, A. e Van Dongen, B. (2012). Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(2):182–192.

Verenich, I., Dumas, M., Rosa, M. L., Maggi, F. M. e Teinemaa, I. (2019b). Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(4):1–34.

Wang, J., Zhang, J. e Wang, X. (2018). A data driven cycle time prediction with feature selection in a semiconductor wafer fabrication system. *IEEE Transactions on Semiconductor Manufacturing*, 31:1–1.