

Predição de tempo restante para conclusão de processos de negócio utilizando aprendizado profundo

Ronildo Oliveira da Silva¹, Regis Pires Magalhães¹, Livia Almada Cruz¹,
Criston Pereira de Souza¹, Davi Romero de Vasconcelos¹,
José Antônio Fernandes de Macêdo²

¹Universidade Federal do Ceará - Campus Quixadá
Quixadá – CE – Brasil

²Departamento de Computação
Universidade Federal do Ceará – Fortaleza – CE – Brasil

ronildo.oliveira@alu.ufc.br,

{regismagalhaes, livia.almada, criston, daviromero}@ufc.br,

jose.macedo@dc.ufc.br

Abstract. *This work aims to predict the remaining time to complete a business process instance using deep learning models. Efficiently predicting the remaining time to complete a process instance contributes to preventing uncertain waits, discovering bottlenecks in a processes, and assist alert systems. This paper uses deep learning architectures to predict the remaining time to conclusion a business process, which surpass state-of-the-art solutions. The architectures used are validated with two sets of public data, facilitating the reproducibility of the experiments.*

Resumo. *A predição eficiente do tempo restante para conclusão de uma instância de processo contribui na prevenção de esperas incertas, descoberta de gargalos em processos e assistência em sistemas de alerta. Este trabalho utiliza arquiteturas de aprendizado profundo baseadas em redes recorrentes para prever o tempo restante para conclusão de um processo de negócio, que superam soluções do estado da arte. As arquiteturas utilizadas são validadas com dois conjuntos de dados públicos, facilitando a reprodutibilidade dos experimentos.*

1. Introdução

Uma grande gama de instituições privadas, além de organizações e departamentos de serviços públicos, almejam produzir melhor e em menos tempo, ou executar tarefas da forma mais eficiente possível [Kalenkova et al. 2017]. Devido a esses fatores, a otimização de processos de negócios é uma área de pesquisa que vem se popularizando [Reijers 2021] e comumente trata de problemas reais de monitoramento de processos. Percebendo a intenção atual de se automatizar e otimizar processos, diversos líderes em diversas empresas de produtos digitais já acreditavam que a otimização de processos de negócio seria a prática mais importante sobre o impacto da transformação digital. Do ponto de vista técnico e atual para otimização de processos, técnicas como aprendizado de máquina desempenham o papel de tecnologia central e podem apoiar a criação de novos modelos de negócios ou se adequar muito bem para a realização de manutenções

preditivas. Conforme o estudo realizado por [Paschek et al. 2017], as perspectivas e estimativas para o ano de 2020 mostravam que o aprendizado de máquina para otimização e automação de processos seria uma ferramenta bastante promissora. A relação entre o gerenciamento de processos e a transformação digital é bastante sólida e [Stjepić et al. 2020] levanta uma série de segmentos de negócios suportados por essas tecnologias.

Este trabalho propõe métodos para prever o tempo restante necessário para que um processo seja concluído. A predição de tempo restante para conclusão de forma satisfatória, pode evitar problemas como possíveis gargalos ([Castro et al. 2022]), dependências cíclicas no processo de negócio, esperas durante um grande intervalo de tempo, falhas ([Mello et al. 2019, Mello et al. 2020]), além de dar suporte à tomada de decisão para otimização dos processos. Com as arquiteturas utilizadas neste trabalho, obtém menor erro médio em relação às arquiteturas do estado da arte.

Dessa forma, apoiando-se nos propósitos atuais de melhoria da gestão de processos, este trabalho objetiva responder às seguintes questões de pesquisa (QP):

QP 1: Quais arquiteturas de aprendizado profundo baseadas em redes neurais conseguem prever o tempo restante para conclusão de um processo de negócio com menor erro?

QP 2: Qual a influência do estágio dos processos nas predições do tempo restante para conclusão de um processo de negócio?

QP 3: Quais diferenças entre o melhor modelo proposto neste trabalho com os modelos propostos em trabalhos relacionados?

2. Definição do problema

Um processo de negócio é uma sequência de tarefas com uma ordem de execução predefinida que gera um produto ou um serviço como resultado. O processo de negócio deve ser definido com clareza, organização e respeito às dependências entre etapas e à representação do modelo real [Ponsard and Darimont 2019].

Uma instância de processo de negócio é uma ocorrência de um processo de negócio dada por uma sequência de eventos $bp_i = [e_1, e_2, \dots, e_n]$, onde e_i associado a um único identificador, um rótulo de tempo da sua execução e uma tarefa relacionada.

As etapas executadas em uma instância de processo de negócio são registradas pelas aplicações como *logs* de eventos. A Tabela 1 apresenta um exemplo de um *log* de eventos de um processo. Cada linha do *log* corresponde a um evento identificado pelo atributo *Atividade* e inicia em determinado instante no tempo (*Instante da criação*). Um evento está associado a uma atividade (*Atividade*) como, por exemplo, “Pegar o bilhete”, “Resolver solicitação”, etc. A instância de processo é representada com um identificador único (*ID do processo*). A finalização de uma instância de processo de negócio é dada quando um estado de finalização é alcançado, neste caso, o atributo *Atividade* do processo recebe o valor “Fechar”. Na Tabela 1, as atividades “Fechar” nas linhas 5 e 9 especificam, respectivamente, que as instâncias *ID 1* e *ID 2* foram finalizadas.

Dado um conjunto de instâncias de processos de negócios já finalizadas *BP* e uma instância de processo de negócio em execução *p*, o problema de predição de tempo restante consiste em prever o tempo que falta para que *p* seja completado. Por exemplo, considere a instância de processo cujo *ID do processo* é 1 (Tabela 1). Suponha que a

instância do processo esteja na atividade “*Definir prioridade*”. O modelo deve prever o tempo X que falta para atingir a atividade “*Fechar*”

Tabela 1. Estrutura de um log de eventos.

ID do processo	Atividade	Instante da criação
1	Definir prioridade	2012/10/09 14:50:17
1	Pegar o bilhete	2012/10/09 14:51:01
1	Pegar o bilhete	2012/10/12 15:02:56
1	Resolver solicitação	2012/10/25 11:54:26
1	Fechar	2012/11/09 12:54:39
2	Definir prioridade	2012/04/03 08:55:38
2	Pegar o bilhete	2012/04/03 08:55:53
2	Resolver solicitação	2012/04/05 09:15:52
2	Fechar	2012/05/19 09:00:28

3. Trabalhos Relacionados

Alguns trabalhos encontrados na literatura propõem soluções para o problema de predição de tempo restante ou problemas semelhantes no contexto de processos de negócio. O trabalho [Tax et al. 2017] investiga uma abordagem baseada em *LSTM* para construir modelos preditivos relacionados a monitoramento de processos. O modelo proposto é formado por um par de camadas *LSTM* intercaladas por camadas de normalização (*Batch Normalization*) e duas saídas, uma para predição do tempo restante para conclusão e outra para predição da próxima atividade. Essa camada de saída voltada para predição da próxima atividade refere-se a um problema de classificação para o estudo de monitoramento preditivo de processos. Cada um dos tipos de instâncias de processo foi representado como um caractere de 2 bytes, reduzindo assim, a quantidade de memória alocada na solução. Para treinamento e avaliação do modelo, o conjunto de dados *Helpdesk 17* foi utilizado.

O modelo desenvolvido por [Navarin et al. 2017] explora informações como momento de início, momento de fim e dia da semana em que ocorreu o evento, além de representar cada atividade com (*One-hot Encoding*), a fim de produzir uma predição do tempo restante para conclusão de instâncias de processos em execução. Similar ao trabalho de [Tax et al. 2017], o modelo de [Navarin et al. 2017] também utiliza *LSTM* como arquitetura principal e o conjunto de dados *Helpdesk17*. A arquitetura do modelo proposto nesse trabalho é constituída por l camadas compostas por uma camada *LSTM*, n , como o número de neurônios por cada camada definida e o otimizador *NAdam*¹ seguida de uma camada de saída, onde l e n são parâmetros de construção da rede.

[Venkateswaran et al. 2021] prediz o tempo restante para conclusão de um processo de negócio tomando como base um modelo de aprendizado profundo composto por duas camadas *LSTM* empilhadas e uma camada de saída densa. Levando em consideração que modelos de processos do mundo real podem sofrer mudanças ao longo do tempo (por exemplo, processos podem tornar-se mais longos, com menos etapas, com prioridades distintas) e que os logs usados para treinar modelos podem estar associados a diferentes versões de processos modificados, a solução proposta utiliza atributos invariantes que possuem forte correlação entre os atributos e o valor predito. Dessa forma, o objetivo é amenizar distorções no modelo quando a distribuição dos dados muda, garantindo predições

¹<https://keras.io/api/optimizers/Nadam/>

mais precisas no decorrer do tempo. Os atributos são representados como um vetor de *embeddings* que possui um bom desempenho quando se leva em consideração a ordem dos dados. A solução foi avaliada com o conjunto de dados *Helpdesk 17* e *BPI 12W*, um subconjunto do *BPI 12*, limitado apenas à eventos de itens de trabalho, que contém eventos executados manualmente.

[Venugopal et al. 2021] apresenta uma solução para o problema de predição de tempo restante baseado em redes neurais de grafos (*GNNs*). A representação matricial do grafo é dada por vértices como estados (*e*) (*Actividades*) e funções de transição (*t*) desse estado a outro sob uma janela de tempo, como as arestas (valores $A_{e,t}$ de uma matriz A) do grafo. Internamente, as camadas ocultas tratam de operações matriciais que envolvem vetores que representam o número de atividades distintas multiplicados pelas matrizes de adjacências que representam o modelo de negócio. A rede neural apresentada é simples com duas camadas densas e uma camada de ativação *ReLU* entre elas. Adicionalmente, uma função de ativação linear com o otimizador *Adam*. A solução foi avaliada com o conjunto de dados *Helpdesk 17* e *BPI 12W*. [Park and Song 2020] desenvolve um método para prever o desempenho futuro de um processo de negócios, objetivando também o suporte às ações proativas para melhorar o processo de negócios. O modelo proposto é híbrido e baseado em *CNN* e *LSTM*, denominado *Long-term Recurrent Convolutional Networks (LRCN)*, que realiza uma combinação de extração de atributos de matrizes em sequências temporais. Os atributos são representados como matrizes que contém informações sobre os desempenhos no processo de negócios. Os conjuntos de dados utilizados são principalmente das competições *BPI* e ao *Helpdesk 17*.

4. Dados e Métodos

Esta seção apresenta os conjuntos de dados utilizados neste trabalho, bem como os pré-processamentos e transformações realizados neles. Trata também das arquiteturas de aprendizado profundo propostas.

4.1. Conjuntos de Dados

Este trabalho utiliza dois conjuntos de dados amplamente conhecidos na literatura, o *BPI 12* e o *Helpdesk 17*, descritos a seguir.

BPI 12: O conjunto de dados *Business Process Intelligence 2012* (*BPI 12*)[van Dongen 2012] foi originalmente disponibilizado na *International Conference on Business Process Management (BPM)* em 2012. O conjunto de dados possui um *log* de eventos de uma instituição financeira holandesa. O *log* diz respeito a processos de solicitação de empréstimos pessoais e contém 156.424 eventos em 13.087 instâncias. Cada instância de processo possui 11,95 etapas em média. Esse conjunto de dados é composto por processos de negócio executados entre 10 de janeiro de 2011 a 4 de março de 2012.

Helpdesk 17: O conjunto de dados *Helpdesk 17* compreende dados de *logs* de eventos de um sistema de gerenciamento de *tickets* projetado para o *helpdesk* de uma empresa de *software* italiana. O *log* de eventos contém 4.580 instâncias com 21.340 eventos. Em média, cada instância de processo possui 4,45 etapas. Esse conjunto de dados é composto por processos de negócio executados entre 13 de janeiro de 2010 a 3 de janeiro de 2014 [Polato 2017]. Uma descrição resumida dos atributos do conjunto de dados *helpdesk* escolhidos para este experimento é apresentada na Tabela 2.

Tabela 2. Descrição das características do conjunto de dados do *BPI 12* e *Helpdesk 17* utilizadas neste trabalho.

Atributo	Descrição
<i>Case ID</i>	Nome do identificador da instância do processo.
<i>Atividade</i>	Nome da Atividade.
<i>Timestamp</i>	O instante em que o evento foi iniciado.

4.1.1. Particionamento dos Dados

Os dados foram particionados em conjuntos de treino, validação e teste. Visto que um processo trata-se de uma sequência temporal, no particionamento realizado, o conjunto de treino contém eventos anteriores aos eventos do conjunto de validação e o conjunto de validação, eventos ocorridos antes dos eventos do conjunto de teste. Apenas processos que finalizaram foram utilizados para treinamento, ou seja, processos em que a atividade do tipo *Fechar* é a última etapa. Como resultado desse filtro, o conjunto de dados do *BPI 12* e *Helpdesk 17* possuem 13.087 e 4.557 instâncias de processos, respectivamente. Eventualmente, a média de etapas por processo se altera, resultando em 3,65 no conjunto de dados do *Helpdesk 17* e se mantendo o mesmo no *BPI 12*, pois esse conjunto de dados é formado apenas de processos que chegaram à sua finalização.

Tabela 3. Números de instâncias de processo e de eventos relacionados ao particionamento dos conjuntos de dados *BPI 12* e *Helpdesk 17*.

Conjunto	BPI 12		Helpdesk 17	
	Instâncias de processo	Eventos	Instâncias de processo	Eventos
Treino	7.852	85.963	2.745	10.254
Validação	2.618	26.865	896	3.417
Teste	2.617	30.509	916	3.004
Total	13.087	143.337	4.557	16.675

4.1.2. Transformação e codificação

Nesta etapa foi realizada a tokenização das atividades, na qual uma codificação numérica e única é atribuída para cada tipo de atividade existente. Além disso, é dado um valor indicativo para a etapa em que o evento se encontra. Uma etapa assume um valor entre $1 \dots n$, onde n é o máximo de etapas que o processo possui (*Etapa*). A definição do tempo em dias se deu pelo uso de três atributos derivados do *Instante da criação* do evento: a duração (*Duração*) do evento, o tempo de execução do processo até o evento atual (*Tempo passado*) e o tempo restante para a conclusão do processo (*Tempo restante para conclusao*). Esse último, representa o rótulo para os conjuntos de treino, validação e de teste deste experimento. A seguir, alguns itens à respeito ao particionamento dos dados, são listados:

1. A composição dos conjuntos de treino, validação e testes segue a proporção 60%, 20% e 20% respectivamente;
2. Cada processo é representado por várias amostras (eventos) nos conjuntos de dados;

3. Os processos são completos. Todas as suas amostras estão contidas em todos os três conjuntos de dados. Não existe parte de um mesmo processo em dois ou mais conjuntos;
4. As linhas que possuem etapas 5 e 4 (Tabela 4) onde os valores da coluna **Tempo restante para conclusão** são iguais a 0, logo, devem ser removidas do conjuntos de treino, validação e teste. Uma vez que o processo encerra, não se vê necessidade de prever nenhuma outra etapa após esse estado.

Tabela 4. Amostras do conjunto de dados Helpdesk 17.

Atividade	Etapas	Duração	Tempo passado	Tempo restante para conclusão
1	1	0,0000	0,0000	31,0087
12	2	16,0084	0,0000	15,0003
9	3	0,0001	15,0002	15,0002
2	4	15,0002	15,0003	0,0000
2	5	0,0000	31,0087	0,0000
1	1	0,0000	0,0000	30,9822
12	2	5,8750	25,1053	25,1072
9	3	0,0019	25,1072	25,1053
2	4	25,1053	30,9822	0,0000

Uma maneira de resumir e reduzir a complexidade dos dados relacionados a atividades, é juntar atividades que se repetem em sequência em uma mesma instância de processo, colapsando-as em uma única atividade, e ajustando seus tempos para acumular os valores das amostras colapsadas. Isso reduz a quantidade de etapas de representação do processo, do trecho que se encontra sem mudança de estado e ajuda o modelo recorrente a entender melhor as possíveis transições, já que uma transição para a mesma *Atividade* está descartada. A Tabela 4 apresenta amostras do conjunto de dados *Helpdesk 17* (o conjunto BP1 12 possui a mesma estrutura, porém valores diferentes). A repetição em sequência de atividades nas etapas (*Etapas*) 4 e 5 da instância de processo de *ID 1* (Tabela 1), permite colapsá-las e ajustar os dados das colunas *Duração*, *Tempo passado* e *Tempo restante para conclusão* com a soma dos valores das linhas colapsadas.

Além disso, pode-se representar essa sequência de atividades (linhas 1 a 3 da Tabela 4) como uma sequência de prefixos na forma matricial (Tabela 5) onde a quantidade de etapas do maior processo é 10 (o conjunto BP1 12 possui a mesma estrutura com *padding* igual à 55).

Tabela 5. Prefixos na amostra do conjunto de dados Helpdesk 17 sem os dados da última instância.

Prefixo e Padding	Atributos temporais			Rótulo
	Duração	Tempo passado	Etapas	Tempo restante para conclusão
0 1	0,0000	0,0000	1	31,0087
1 12	16,0084	15,0002	2	15,0003
1 129	0,0001	15,0003	3	15,0002

No exemplo acima (Tabela 5), os valores iguais a 0 são usados como preenchimentos (*padding*s), que possibilitam uniformizar o número de colunas na representação das

atividades das instâncias de processos. Essa matriz resultante substitui a coluna *Atividade* (Tabela 4).

4.2. Arquiteturas propostas

As arquiteturas propostas neste trabalho são estruturadas utilizando *Embeddings* para codificação vetorial das atividades, *BiLSTM* (*Bidirectional Long Short Term Memory*) e mecanismos de atenção. A arquitetura visa mesclar o suporte aos dados temporais e sequenciais que o processo de negócio possui, dado pela *BiLSTM*, atrelado à identificação da importância de certas etapas do processo, feita pela camada de atenção. O modelo mais complexo (Figura 1 A)) é um modelo sequencial onde os dados de definição da atividade 4 passam por uma camada de *Embedding* e em seguida são processados por uma camada de autoatenção (*Self-Attention* – SA). Em seguida, a saída dessa camada é concatenada aos dados temporais de entrada. O resultado, é uma nova entrada para cada camada de *BiLSTM* concatenada com a camada de dados temporais. Além disso, são realizados experimentos com *Transformer Encodding*, modelo genérico de ingestão de sequências e manipulação de representações por meio de codificador e decodificador.

Para avaliação da arquitetura, foram obtidos modelos a partir de variações do modelo mais complexo. As variações são baseadas principalmente no uso e na ordem em que as camadas de atenção (*Self-attention*) e a *BiLSTM* são dispostas. Foram avaliados os seguintes modelos:

A) SA_BiLSTM. Este modelo recebe como entrada um par formado pelos prefixos que compõem as etapas do processo e os dados temporais (Tabela 4). A primeira parte da entrada é enviada a uma camada de *Embedding* para que os dados representantes das atividades sejam transformados em vetores. A camada de *Embedding* é entrada da camada de autoatenção e lida com diferentes partes da sequência da entrada inicial. Em seguida, a camada de atenção se conecta a uma camada de *BiLSTM*, que são usadas principalmente para processar sequências de dados em ambas as direções, capturando dependências de longo prazo. Isso pode fornecer um contexto adicional e resultar em um aprendizado mais completo dos dados. A segunda parte do par de entrada corresponde aos dados temporais de cada etapa da execução do processo. Essa, por sua vez é concatenada à camada de *BiLSTM*, que, então, é atrelada a uma camada densa.

B) SA_DENSE. Similar à arquitetura anterior, esta não possui a camada de *BiLSTM*. É uma organização da arquitetura de modo a analisar o comportamento da rede sem as interações entre camadas nas direções *backwards* e *forward*, mas mantendo a influência da camada de autoatenção, que por sua vez, compara todos os elementos (etapas do processo) da sequência de entrada entre si e modifica as posições correspondentes da sequência de saída.

C) DENSE Seguindo a ideia de reduzir a complexidade do modelo, esta arquitetura é uma das mais simples apresentadas nessa seção. Além da ausência da *BiLSTM*, a camada de atenção também não é usada aqui. Este modelo se assemelha à modelos clássicos de aprendizado de máquina.

D) BiLSTM_SA Esta versão é uma variação da arquitetura A, onde a camada de atenção vem depois da camada de *BiLSTM*. A proposta aqui é realizar o processamento da camada de atenção após o aprendizado da *BiLSTM*.

E) BiLSTM Analogamente, à arquitetura *C*, é removida a camada de atenção, deixando esta arquitetura numa configuração simples de *RNN* com a concatenação dos dados temporais.

F) TRANSFORMER_ENCODER A segunda arquitetura mais simples deste experimento se configura como a arquitetura *C* e utiliza uma camada de transformação e codificação ao invés de uma camada de *Embedding*.

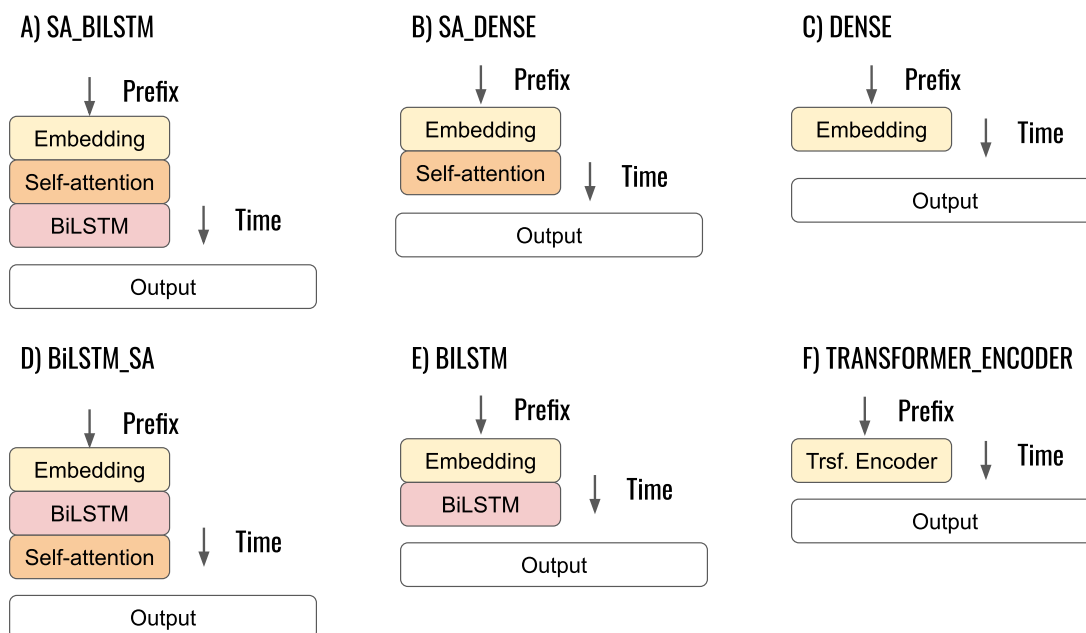


Figura 1. Arquiteturas de redes neurais profundas usadas.

Metodologia de avaliação

Todos os modelos descritos na Figura 1 recebem como entrada, as sequências de prefixos e de definições das características temporais do conjunto de treino.

A métrica de avaliação é baseado no MAE (*Mean Absolute Error*), calculado pela soma dos erros absolutos dividida pelo tamanho da amostra (*n*), onde y_i é o valor observado e \hat{y}_i é o valor previsto:

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

A Tabela 6 organiza os valores de MAE de cada um dos subconjuntos definidos para o experimento e análise de significância de cada partição de dados.

Tabela 6. Desvio padrão (em dias) das labels dos conjunto de treino, validação e teste do BPI 12 e Helpdesk 17.

Conjunto	Desvio Padrão	
	BPI 12	Helpdesk 17
Treino	13,0282	11,1193
Validação	11,9414	9,9398
Teste	8,5743	12,5011

É possível considerar também a avaliação por estágio de processo, ou seja, o quão bem o modelo de aprendizado consegue prever tendo como conjunto de teste apenas atividades que se encontram numa determinada etapa.

5. Experimentos e Resultados

Esta seção exibe e discute os resultados dos experimentos realizados com os modelos de aprendizado preditivos propostos. Como soluções de linha de base foram utilizadas as seguintes estratégias: *Dummy Regressor*: Retorna o valor médio dos valores esperados no conjunto de treinamento, sem observar os valores de entrada; Regressão com Floresta Aleatória, regressão linear, Regressão com Aumento de Gradiente Extremo e leve. A solução proposta e suas variações também foram comparadas a soluções de aprendizado profundo do estado da arte para predição de tempo restante de processos, sendo duas soluções baseadas em LSTM [Navarin et al. 2017, Tax et al. 2017], e uma solução baseada em Transformers [Bukhsh et al. 2021]. A Tabela 7 apresenta os resultados obtidos com a métrica MAE para os conjuntos de dados *BPI 12* e *Helpdesk 17*. Nessa experimentação são consideradas apenas as instâncias de processos que chegaram a sua conclusão.

As métricas apresentadas para os trabalhos relacionados são obtidas a partir da execução de novos experimentos usando um particionamento e filtragem diferentes dos usados nos trabalhos originais. Buscou-se, assim, a mesma filtragem e particionamento para uma comparação mais justa entre todas as abordagens.

Tabela 7. Desempenho dos modelos usando as métricas: MAE.

Modelo	BPI 12	Helpdesk 17
	Valor MAE (95% I.C.)*	Valor MAE (95% I.C.)*
Dummy Regressor	7,64 (7,58 - 7,69)	12,06 (11,77 - 12,32)
Regressão Linear	6,03 (5,98 - 6,08)	12,02 (11,48 - 12,67)
Regressão com Floresta Aleatória	4,89 (4,82 - 4,96)	5,52 (5,34 - 5,69)
Regressão com Aumento de Gradiente Extremo	5,02 (4,95 - 5,09)	5,73 (5,55 - 5,90)
Regressão com Aumento de Gradiente Leve	4,84 (4,78 - 4,90)	5,41 (5,26 - 5,57)
LSTM [Tax et al. 2017]	N.A. **	6,17 (5,50 - 6,33)
LSTM [Navarin et al. 2017]	7,25 (7,17 - 7,34)	8,46 (8,30 - 8,62)
TKN_TRANSF [Bukhsh et al. 2021]	4,91 (4,88 - 4,94)	5,69 (5,53 - 5,86)
SA_BiLSTM	3,80 (3,74 - 3,87)	5,70 (5,50 - 5,89)
DENSE	4,25 (4,18 - 4,31)	7,58 (7,42 - 7,75)
BiLSTM_SA	3,72 (3,65 - 3,79)	5,89 (5,70 - 6,07)
BiLSTM	3,72 (3,65 - 3,79)	5,58 (5,39 - 5,77)
SA_DENSE	3,71 (3,65 - 3,78)	5,22 (5,04 - 5,40)
TRANSF_ENCO	4,24 (4,18 - 4,31)	6,64 (6,48 - 6,80)

* Intervalo de confiança abaixo e sem interseção. ** Não foi possível realizar o experimento completo.

5.1. Discussão dos resultados

A listagem a seguir discorre a respeito das características, resultados de cada uma das arquiteturas apresentadas neste trabalho.

SA_BiLSTM Neste cenário, a ordem em que se configura a camada de atenção importa bastante no resultado, principalmente quando há sequências grandes na execução do pro-

cesso, o qual é o caso do *BPI 12*, produzindo um resultado mediano entre os as arquiteturas apresentadas neste trabalho e difere em 1.44 dias do melhor *baseline* (Regressão com Aumento de Gradiente Leve) apresentado. Ao ter uma sequência menor, produz também resultados semelhantes (*Helpdesk 17*) e configura também uma arquitetura produtora de um resultado mediado (terceiro melhor resultado encontrado) e dista de 0.29 dias a mais de diferença do melhor *baseline*, assim, configurando uma arquitetura pouco viável para sequências menores.

DENSE Apesar dessa arquitetura ser a mais simples e próxima dos modelos de *baseline* do que diz respeito à robustez, uma arquitetura baseada apenas em camadas densas, produzem os resultados com maior erro que os demais modelos de aprendizado profundo apresentados neste trabalho. Além disso, esta arquitetura produz melhores resultados quando se utilizam sequencias maiores em relação aos *baselines* propostos.

BiLSTM_SA Esta arquitetura é uma variação clara da arquitetura SA_BiLSTM (A) e também produz resultados medianos. Quando se tem sequências maiores no *log* de eventos e maiores distribuições das atividades, o erro absoluto médio é um pouco menor em relação aos *baselines* apresentados. Em contrapartida, não é um bom modelo quando as sequências são menores ou com uma distribuição menor de atividades (*Helpdesk 17*). A quantidade de instâncias, o tamanho máximo de atividades por execução do processo é um fator de grande influência na combinação de camadas de auto-atenção com *BiLSTM*.

BiLSTM Modelos de *RNN* são amplamente usados na solução de problemas envolvendo sequências de dados para capturar dependências temporais, como as usadas neste trabalho e nos trabalhos relacionados. Os experimentos demonstram que utilizar apenas uma camada nessa configuração já é capaz de produzir resultados próximos aos melhores resultados obtidos em todas as estratégias analisadas neste trabalho.

SA_DENSE Esta arquitetura obteve os melhores resultados para ambos os conjuntos de dados *BPI 12* e *Helpdesk 17*. Um grande colaborador para esse resultado é que a camada de auto-atenção se conecta em toda a sequência processada, um ponto de melhor performance em relação as *RNN*.

TRANSFORMER_ENCODER Esta arquitetura deixa de utilizar *embeddings* para ter uma camada de *Transformer Encoding* como parte fundamental do modelo. Este modelo é bastante semelhante ao descrito na Figura 1 C). Ambos os resultados não são tão promissores e são apenas melhores que os resultados obtidos com a arquitetura DENSE. Os resultados obtidos utilizando o tanto os *datasets BPI 12* quanto *Helpdesk 17* produzem o pior resultado dentre as arquiteturas apresentadas neste trabalho.

5.2. Discussão orientada às questões de pesquisa

Esta seção se propõe a responder às questões de pesquisa do trabalho a partir das análises e conclusões sobre os experimentos realizados.

QP1 - Que arquiteturas de aprendizado profundo conseguem prever o tempo restante para conclusão de um processo de negócio com menor erro?

Predizer o tempo restante para a conclusão de um processo de negócio torna-se um desafio, principalmente por conta da variabilidade na execução das instâncias de processos, que não se comportam como uma série temporal única. Por exemplo, as atividades dos processos podem mudar com o passar do tempo em termos de eficiência, nome e até

mesmo existência, visto que algumas atividades podem ser extintas. Pelos resultados da experimentação realizada, observa-se que uso do mecanismo de autoatenção pôde melhorar alguns resultados quando aplicado diretamente na representação vetorial das atividades dada pela saída da camada de *Embedding*, sendo a melhora mais significativa ocorrida ao comparar a arquitetura DENSE com a SA_DENSE em ambos os conjuntos de dados. Além disso, para o conjunto de dados *BPI 12*, os experimentos demonstraram que não houve melhoria no uso da *BiLSTM* com camada de atenção. Finalmente, as arquiteturas *BiLSTM* e SA_DENSE foram eficientes na predição de tempo restante dos processos, quando comparadas aos modelos de linha de base e aos trabalhos de [Tax et al. 2017], [Navarin et al. 2017] e [Bukhsh et al. 2021], obtendo o melhor desempenho entre os modelos avaliados. Mais precisamente, a SA_DENSE mostrou-se melhor para o *BPI 12* e *Helpdesk 17* simultaneamente.

QP2 - Qual a influência do estágio dos processos nas predições do tempo restante para conclusão de um processo de negócio?

Para responder a esta pergunta, variou-se o estágio dos processos como entrada para os modelos e obteve-se o MAE para cada um dos estágios. Nesse experimento, foi utilizado o melhor modelo avaliado para cada conjunto de dados. As Figuras 2 e 3 apresentam o MAE obtido para cada estágio do processo dos conjuntos de dados *BPI 12* e *Helpdesk 17*, respectivamente. Observa-se que o erro na predição tem tendência a ser maior nos primeiros estágios do processo. Após alguns estágios, o erro mantém-se variando e começa a diminuir em estágios posteriores. Tal comportamento pode indicar que se há mais informações sobre a execução do processo e se este se encontra mais próximo ao estado final, então, torna-se mais fácil prever o tempo restante. Este comportamento também pode indicar que os modelos conseguem distinguir melhor as atividades realizadas mais ao final da execução dos processos.

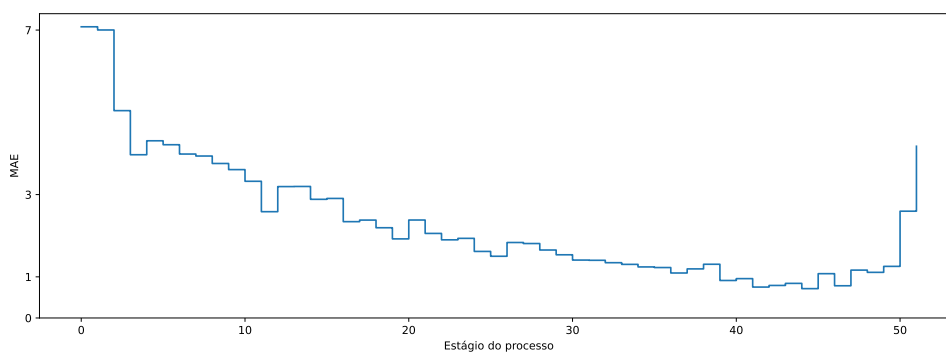


Figura 2. BPI 12 - MAE por estágio do processo. O último estágio representa a média entre todos os estágios.

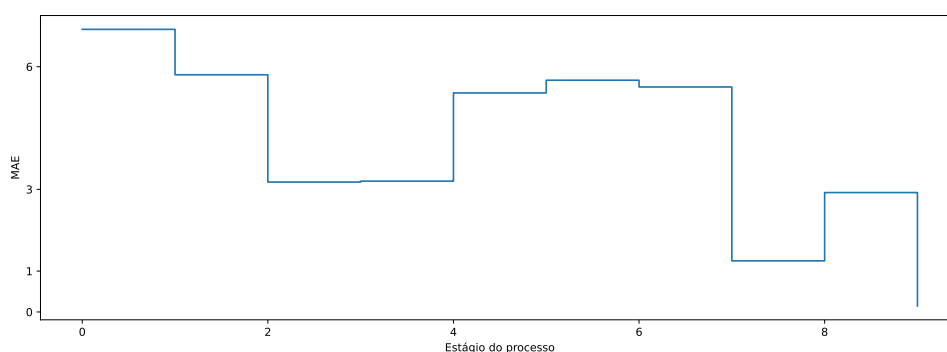


Figura 3. Helpdesk 17 - MAE por estágio do processo. O último estágio representa a média entre todos os estágios.

QP3 - Como se compara o melhor modelo proposto neste trabalho com os modelos propostos em trabalhos relacionados?

Os modelos apresentados neste trabalho produzem resultados semelhantes, mas a utilização de camada de atenção, produz resultados com menor erro absoluto médio. Os modelos apresentados nos trabalhos relacionados diferem na complexidade da RNN ([Tax et al. 2017] e [Navarin et al. 2017]) e na existência de uma *RNN* para suportar as sequências ([Bukhsh et al. 2021]), apesar de ambos os trabalhos utilizarem *embeddings*. O melhor modelo apresentado neste trabalho é mais robusto, utiliza conjuntos de dados de *logs* que finalizam, e consegue produzir resultados com desempenho superior a trabalhos relacionados, bem como a outras estratégias mais tradicionais de aprendizado de máquina.

6. Conclusões e Trabalhos Futuros

Este artigo investigou diferentes abordagens para predição do tempo restante para conclusão de um processo. Dentre as abordagens avaliadas neste trabalho, o uso de aprendizado profundo e camada de autoatenção lidam bem com *logs* de eventos reais, independente de ter instâncias automatizadas e manuais no mesmo conjunto de dados. Como trabalhos futuros, é importante considerar outros conjuntos de dados, explorar *logs* de diferentes competições voltadas à monitoramento preditivo de processos, verificar como arquiteturas mais recentes podem contribuir na solução desse problema, comparar modelos clássicos com os atuais, disponibilizando o repositório² com os experimentos desenvolvidos para dar suporte a novas práticas.

Referências

- Bukhsh, Z. A., Saeed, A., and Dijkman, R. M. (2021). Processtransformer: Predictive business process monitoring with transformer network. *arXiv preprint arXiv:2104.00721*.
- Castro, M. A., Souza Jr, N., Escovedo, T., Lopes, H., and Kalinowski, M. (2022). Mineração de processos aplicada à auditoria interna na marinha do brasil. In *Anais do XXXVII Simpósio Brasileiro de Bancos de Dados*, pages 241–253. SBC.

²https://github.com/RonildoSilva/PTRC_PN_DL

- Kalenkova, A., Ageev, A., Lomazova, I. A., and van der Aalst, W. M. (2017). E-government services: Comparing real and expected user behavior. In *International Conference on Business Process Management*, pages 484–496. Springer.
- Mello, P., Santoro, F., and Revoredo, K. (2020). It incident solving domain experiment on business process failure prediction. *Journal of Information and Data Management*, 11(1).
- Mello, P. O., Revoredo, K., and Santoro, F. (2019). Business process failure prediction: a case study. In *Anais do VII Symposium on Knowledge Discovery, Mining and Learning*, pages 89–96. SBC.
- Navarin, N., Vincenzi, B., Polato, M., and Sperduti, A. (2017). Lstm networks for data-aware remaining time prediction of business process instances. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7. IEEE.
- Park, G. and Song, M. (2020). Predicting performances in business processes using deep neural networks. *Decision Support Systems*, 129:113191.
- Paschek, D., Luminosu, C. T., and Draghici, A. (2017). Automated business process management—in times of digital transformation using machine learning or artificial intelligence. In *MATEC Web of Conferences*, volume 121, page 04007. EDP Sciences.
- Polato, M. (2017). Dataset belonging to the help desk log of an italian company.
- Ponsard, C. and Darimont, R. (2019). Towards goal-oriented analysis and redesign of bpmn models. In *MODELSWARD*, pages 527–533.
- Reijers, H. A. (2021). Business process management: The evolution of a discipline. *Computers in Industry*, 126:103404.
- Stjepić, A.-M., Ivančić, L., and Vugec, D. S. (2020). Mastering digital transformation through business process management: Investigating alignments, goals, orchestration, and roles. *Journal of entrepreneurship, management and innovation*, 16(1):41–74.
- Tax, N., Verenich, I., Rosa, M. L., and Dumas, M. (2017). Predictive business process monitoring with lstm neural networks. In *International Conference on Advanced Information Systems Engineering*, pages 477–492. Springer.
- van Dongen, B. (2012). Bpi challenge 2012.
- Venkateswaran, P., Muthusamy, V., Isahagian, V., and Venkatasubramanian, N. (2021). Robust and generalizable predictive models for business processes. In *Business Process Management: 19th International Conference, BPM 2021, Rome, Italy, September 06–10, 2021, Proceedings*, pages 105–122. Springer.
- Venugopal, I., Töllich, J., Fairbank, M., and Scherp, A. (2021). A comparison of deep-learning methods for analysing and predicting business processes. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.