



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
MESTRADO ACADÊMICO EM COMPUTAÇÃO

RONILDO OLIVEIRA DA SILVA

**PREDIÇÃO DE TEMPO RESTANTE PARA CONCLUSÃO DE PROCESSOS DE
NEGÓCIO UTILIZANDO APRENDIZADO PROFUNDO**

QUIXADÁ

2023

RONILDO OLIVEIRA DA SILVA

PREDIÇÃO DE TEMPO RESTANTE PARA CONCLUSÃO DE PROCESSOS DE NEGÓCIO
UTILIZANDO APRENDIZADO PROFUNDO

Dissertação apresentada ao Curso de Mestrado Acadêmico em Computação do Programa de Pós-Graduação em Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Regis Pires Magalhães

Coorientadora: Prof^a. Dr^a. Livia Almada Cruz

QUIXADÁ

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S583p Silva, Ronildo Oliveira da.
Predição de tempo restante para conclusão de processos de negócio utilizando aprendizado profundo /
Ronildo Oliveira da Silva. – 2023.
74 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Campus de Quixadá, Programa de Pós-
Graduação em Computação, Quixadá, 2023.

Orientação: Prof. Dr. Regis Pires Magalhães.

Coorientação: Profa. Dra. Lívia Almada Cruz.

1. Processo de negócio. 2. Predição de tempo restante. 3. Aprendizagem profunda. I. Título.

CDD 005

RONILDO OLIVEIRA DA SILVA

PREDIÇÃO DE TEMPO RESTANTE PARA CONCLUSÃO DE PROCESSOS DE NEGÓCIO
UTILIZANDO APRENDIZADO PROFUNDO

Dissertação apresentada ao Curso de Mestrado Acadêmico em Computação do Programa de Pós-Graduação em Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Computação. Área de Concentração: Ciência da Computação

Aprovada em: ___/___/___

BANCA EXAMINADORA

Prof. Dr. Regis Pires Magalhães (Orientador)
Universidade Federal do Ceará (UFC)

Prof^ª. Dr.^a. Lívia Almada Cruz (Coorientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Davi Romero de Vasconcelos
Universidade Federal do Ceará (UFC)

Prof. Dr. José Antônio Fernandes de Macêdo
Universidade Federal do Ceará (UFC)

Prof. Dr. Francesco Lettich
Consiglio Nazionale delle Ricerche (Pisa - Itália)

Este trabalho é dedicado ao meu pai, “Seu Joaquim”, minha referência de honestidade, responsabilidade e respeito.

AGRADECIMENTOS

À minha família que ficou satisfeita pelo ingresso no mestrado, que se fosse um sonho meu chegar mais próximo do doutorado, seguisse, mas que se eu quisesse dar um tempo nos estudos e ficasse uns meses quieto e começasse depois, também ficariam satisfeitos.

Meus amigos (eu ia colocar o nome de cada um, mas são poucos e eu ia ficar constrangido) que também disseram que eu fosse descansar e fizesse uma pós depois, pois ia dar trabalho e eu não estava bem. Infelizmente vou ter de admitir que vocês estavam certos. Agora tá tudo equilibrado e daqui pra frente terá mais forró que processo seletivo.

A Iris Bonfim, Valdemir, Cosmo e Wanda que me ajudaram a colocar minha cabeça no lugar, dado uns anos muito difíceis para o planeta Terra.

Aos professores: Paulo Rego, Davi Romero, Marciel Barros e André Meireles pelos tantos “Vamos trabalhar com a gente?”.

Ao professor Joel que há anos vem me respondendo qualquer dúvida que eu tenha em estatística e que sempre me mandou confiar nos meus resultados.

Ao professor Regis que há anos vem me orientado muito bem em trabalhos e projetos desde a graduação, ao Professor Criston que sempre me deu ótimas ideias de como melhorar meus trabalhos. Por último, mas não menos importante, à professora Lívia que chegou para contribuir com este trabalho num momento em que eu não tinha mais ideias e conseguimos concluir este trabalho.

Este trabalho é parcialmente financiado pelos projetos Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) e Fundação de Apoio a Serviços Técnicos, Ensino e Fomento a Pesquisas (FASTEF) 04772314/2020, 04772420/2020 e 04772551/2020 e fez parte do Projeto Cientista Chefe que objetiva unir o meio acadêmico e a gestão pública para identificar soluções de ciência, tecnologia e inovação que podem ser implantadas para melhorar os serviços e, desta forma, dar mais qualidade de vida para a população cearense.

“... eu ia explodir, mas eles não vão ver os meus
pedaços por aí ...

Hoje eu desafio o mundo sem sair da minha casa;
hoje eu sou um homem mais sincero e mais justo
comigo.

Podem os homens vir que não vão me abalar, os
cães farejam o medo, logo não vão me encontrar
...”

(Marcelo Yuka - *in memoriam*)

RESUMO

A análise de processos de negócios — *Business Process* (BP) é uma área que faz parte do estudo em mineração de processos — *Process Mining* (PM), área que abrange o monitoramento preditivo de processos — *Predictive Process Monitoring* (PPM).

Em linhas gerais, PPM visa fazer tanto previsões sobre instâncias de processos individuais, como: qual a próxima instância que deve ser executada dado eventos anteriores, o tempo restante para sua conclusão de uma instância de processo, quanto se aplica em modelos de processos gerais que não incluem necessariamente o foco em negócios.

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 propõe novas arquiteturas de aprendizado profundo baseadas em redes recorrentes para prever o tempo restante para conclusão de um processo de negócio, que superam soluções estado da arte. As arquiteturas utilizadas são validadas com dois conjuntos de dados públicos e três privados, facilitando a reprodutibilidade dos experimentos conforme a disponibilidade dos dados.

Palavras-chave: processo de negócio; predição de tempo restante; predição de tempo restante; auto-atenção.

ABSTRACT

Business process analysis belongs to process mining study area that covers the predictive monitoring process and aim to do predictions over both individuals process like: what the next process instance will be executed when provide past events, the remaining time to process instance conclusion when it applies in general process models that is not necessarily includes a business focus.

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 processes, and assist alert systems. This paper proposes new architectures of deep learning with recurrent networks 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 and another three private one, facilitating the reproducibility of the experiments.

Keywords: business process; remaining time prediction; deep learning; self-attention.

LISTA DE FIGURAS

Figura 1 – Influência do aprendizado de máquina e Inteligência Artificial (IA) como ferramentas de apoio à gestão de processos.	16
Figura 2 – Número de estudos publicados ao longo do tempo.	16
Figura 3 – Visão geral do monitoramento preditivo de processos.	22
Figura 4 – Desvios de temperatura global média anual (1900-1997) em graus centígrados	23
Figura 5 – Etapas em que normalmente se encontram a engenharia de <i>features</i> no fluxo de trabalho (<i>pipeline</i>) do aprendizado de máquina	24
Figura 6 – No painel superior esquerdo, um conjunto de dados de treinamento cuja fonte de dados são informações de preços de brinquedos de dez empresas associadas a valores de receita. No painel superior direito, um modelo linear é ajustado aos dados. Esta linha de tendência modela a trajetória geral dos pontos e pode ser usada para previsão no futuro, conforme mostrado no painel inferior esquerdo e inferior direito	26
Figura 7 – Esquema de uma árvore de decisão	27
Figura 8 – Esquema do algoritmo de floresta aleatória.	28
Figura 9 – Descrição gráfica do <i>Mean Absolute Error</i> (MAE). A linha verde representa as previsões de um modelo e os pontos azuis representam os dados reais . . .	29
Figura 10 – Esquema de funcionamento de um modelo de aprendizado profundo	31
Figura 11 – <i>Recurrent Neural Network</i> (RNN) e a representação sequencial do seu <i>loop</i> de sua computação. Os neurônios artificiais recebem informações de outros neurônios em intervalos de tempo anteriores.	31
Figura 12 – Diferença entre as arquiteturas de RNN e <i>Long short-term memory</i> (LSTM).	32
Figura 13 – Arquitetura de um bloco de auto-atenção.	33
Figura 14 – Bloco de auto-atenção (ao centro) compondo arquitetura de um modelo de <i>transformer</i>	34
Figura 15 – Estrutura da arquitetura do modelo	36
Figura 16 – Arquitetura apresentada em trabalho	37
Figura 17 – Estrutura da arquitetura apresentada nesse trabalho	39
Figura 18 – Fluxo de treinamento do <i>Robust and generalizable</i> (RoGen)	40

Figura 19 – Arquitetura de <i>Graph neural network</i> (GNN) para predição de tipo de atividade e predição <i>timestamp</i> . O valor de n na última camada denota o número de classes para o preditor de eventos e 1 para o preditor de tempo.	41
Figura 20 – Esquema da sequência de etapas do processo metodológico.	43
Figura 21 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados do <i>BPI 12</i>	48
Figura 22 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados <i>Helpdesk 17</i>	49
Figura 23 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados <i>A_25</i>	50
Figura 24 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados <i>A_50</i>	51
Figura 25 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados <i>A_75</i>	52
Figura 26 – Arquiteturas de redes neurais usadas: (A) SA_BILSTM; (B) SA_DENSE; (C) DENSE; (D) BILSTM_SA; (E) BILSTM; (F) TRANSFORMER_ENCODER; (G) (TAX); (H) LSTM (NAVARIN); (I) TKN_TRANSF (BUKSHSH)	56
Figura 27 – <i>BPI 12</i> - MAE por estágio do processo (<i>SA_DENSE</i>).	65
Figura 28 – Conjunto de dados <i>Helpdesk 17</i> - MAE por estágio do processo (<i>SA_DENSE</i>).	66
Figura 29 – Conjunto de dados <i>A_25</i> - MAE por estágio do processo (<i>BiLSTM</i>).	66
Figura 30 – Conjunto de dados <i>A_50</i> - MAE por estágio do processo (<i>TKN_TRANSF Bukshsh et al. (2021)</i>).	67
Figura 31 – Conjunto de dados <i>A_50</i> - MAE por estágio do processo (<i>BiLSTM_SA</i>).	68
Figura 32 – Conjunto de dados <i>A_75</i> - MAE por estágio do processo.	68

LISTA DE TABELAS

Tabela 1 – Estrutura de um <i>log</i> de evento.	21
Tabela 2 – Comparativo dos trabalhos relacionados.	42
Tabela 3 – Descrição das características do conjunto de dados do <i>Helpdesk 17</i> utilizadas neste trabalho	45
Tabela 4 – Números de instâncias de processo e de eventos relacionados ao particionamento dos conjuntos de dados <i>BPI 12</i> , <i>Helpdesk 17</i> , <i>A_25</i> , <i>A_50</i> e <i>A_75</i>	46
Tabela 5 – Quantidade média de eventos por processo.	46
Tabela 6 – Quantidade de atividades distintas por subconjunto utilizado no experimento.	47
Tabela 7 – Intersecção das atividades distintas entre os conjuntos de treino, validação e teste.	50
Tabela 8 – Tamanho do maior processo encontrado nos conjuntos de dados.	51
Tabela 9 – Amostras do conjunto de dados <i>Helpdesk 17</i>	52
Tabela 10 – Prefixos na amostra do conjunto de dados <i>Helpdesk 17</i>	53
Tabela 11 – Desvio padrão (em dias) das <i>labels</i> dos conjunto de treino, validação e teste do <i>BPI 12</i> , <i>Helpdesk 17</i> , <i>A_25</i> , <i>A_50</i> e <i>A_75</i>	57
Tabela 12 – Média (em dias) das <i>labels</i> dos conjunto de treino, validação e teste do <i>BPI 12</i> , <i>Helpdesk 17</i> , <i>A_25</i> , <i>A_50</i> e <i>A_75</i>	57
Tabela 13 – Desempenho dos modelos usando a métrica MAE.	59

LISTA DE ABREVIATURAS E SIGLAS

BiLSTM	<i>Bidirectional Long Short Term Memory</i>
BP	<i>Business Process</i>
BPI	<i>Business Process Improvement</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CF-e	Cupons Fiscais Eletrônicos
CTC-ES	Conselho Técnico-Científico da Educação Superior
DL	<i>Deep Learning</i>
EFD	Escrita Fiscal Digital
FASTEF	Fundação de Apoio a Serviços Técnicos, Ensino e Fomento a Pesquisas
FUNCAP	Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico
GNN	<i>Graph neural network</i>
IA	Inteligência Artificial
ID	Identificador
IRM	<i>Invariant Risk Minimization</i>
LRCN	<i>Long-term Recurrent Convolutional Networks</i>
LSTM	<i>Long short-term memory</i>
MAE	<i>Mean Absolute Error</i>
NF-e	Notas Fiscais Eletrônicas
PM	<i>Process Mining</i>
PPM	<i>Predictive Process Monitoring</i>
RNN	<i>Recurrent Neural Network</i>
RoGen	<i>Robust and generalizable</i>
SA	<i>Self-Attention</i>
SBBD	Simpósio Brasileiro de Bancos de Dados
SBC	Sociedade Brasileira de Computação
SEFAZ-CE	Secretaria da Fazenda do Estado do Ceará
TF-IDF	<i>Term Frequency – Inverse Document Frequency</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Questões de pesquisa	17
1.2	Contribuições e Publicações	17
1.3	Objetivos	18
1.4	Estrutura do trabalho	18
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	Processo de Negócio	20
2.1.1	<i>Registros (logs) de evento</i>	20
2.2	Monitoramento preditivo de processos	22
2.3	Série temporal	22
2.4	Engenharia de características	23
2.5	Aprendizado de máquina	24
2.5.1	<i>Aprendizado supervisionado</i>	25
2.6	Aprendizado profundo	29
2.6.1	<i>Redes Neurais Recorrentes</i>	30
2.6.1.1	<i>Long Short Term Memory</i>	32
2.6.2	<i>Mecanismo de auto-atenção</i>	32
3	TRABALHOS RELACIONADOS	35
3.1	Trabalho Relacionado - LSTM para Regressão e Classificação	35
3.2	Trabalho Relacionado - LSTM Para Regressão	36
3.3	Trabalho Relacionado - <i>Embeddings e Transformers</i>	38
3.4	Trabalho Relacionado - LRCN	38
3.5	Trabalho Relacionado LSTM com IRM	39
3.6	Trabalho Relacionado - GNN	41
4	METODOLOGIA	43
4.1	Conjuntos de Dados	43
4.2	Particionamento dos Dados	45
4.3	Transformação e codificação	50
4.4	Arquiteturas propostas	53
4.5	Execução dos experimentos	55

4.5.1	<i>Metodologia de avaliação</i>	56
5	RESULTADOS	58
5.1	Discussão de resultados	59
5.1.1	<i>SA_BiLSTM</i>	59
5.1.2	<i>DENSE</i>	60
5.1.3	<i>BiLSTM_SA</i>	60
5.1.4	<i>BiLSTM</i>	61
5.1.5	<i>SA_DENSE</i>	62
5.1.6	<i>TRANSFORMER_ENCODER</i>	62
5.1.7	<i>LSTM Tax et al. (2017)</i>	63
5.1.8	<i>LSTM Navarin et al. (2017)</i>	63
5.1.9	<i>TKN_TRANSF Bukhsh et al. (2021)</i>	63
5.2	Discussão orientada às questões de pesquisa.	63
5.2.1	<i>QP1 — Que arquiteturas de aprendizado profundo conseguem prever o tempo restante para conclusão de um processo de negócio com menor erro?</i>	64
5.2.2	<i>QP2 — Qual a influência do estágio dos processos nas previsões do tempo restante para conclusão de um processo de negócio?</i>	64
5.2.2.1	<i>Análise sobre estágios do BPI 12</i>	65
5.2.2.2	<i>Análise sobre estágios do Helpdesk 17</i>	65
5.2.2.3	<i>Análise sobre estágios do A_25</i>	66
5.2.2.4	<i>Análise sobre estágios do A_50</i>	66
5.2.2.5	<i>Análise sobre estágios do A_75</i>	67
5.2.3	<i>QP3 — Como se compara o melhor modelo proposto neste trabalho com os modelos propostos em trabalhos relacionados?</i>	68
6	CONCLUSÕES E TRABALHOS FUTUROS	70
	REFERÊNCIAS	71

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 isso, a otimização de processos de negócios é uma área de pesquisa que vem crescendo Reijers (2021) e demonstra aplicação em problemas reais de monitoramento de processos. Percebendo a intenção atual que se direciona à automatização e otimização de 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. Segundo Paschek *et al.* (2017), 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.

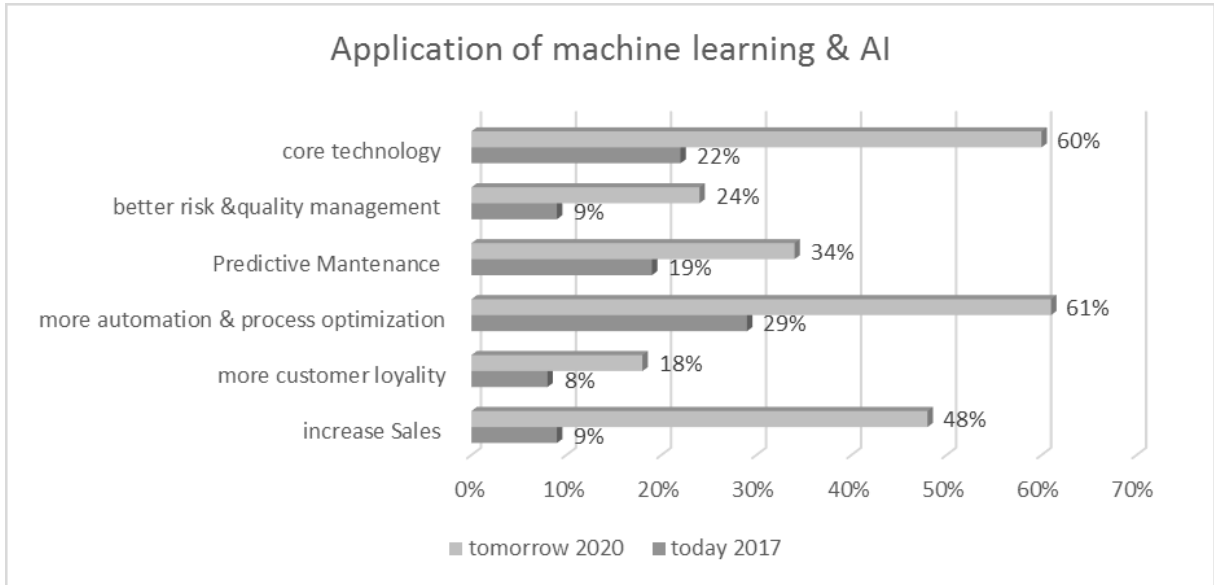
Conforme o estudo realizado por Paschek *et al.* (2017), relatou que as perspectivas e estimativas para o ano de 2020 mostravam uma posição fortalecida do aprendizado de máquina para otimização e automação de processos. A relação entre o gerenciamento de processos e a transformação digital é sólida. Líderes em diversas empresas de produtos digitais na Alemanha já acreditavam que a otimização de processos de negócio seria a prática mais importante sobre o impacto da Transformação Digital. Conforme os participantes submetidos a uma entrevista realizada pelo autor do trabalho, técnicas como aprendizado de máquina desempenham um papel de “tecnologia central” e pode apoiar a criação de novos modelos de negócios ou se adequar muito bem para a realização de manutenção preditiva. Segundo o autor, as perspectivas e estimativas para o ano de 2020 mostravam uma posição fortalecida do aprendizado de máquina, campo de pesquisa e produção que está presente quando se deseja mais otimização de processos e mais automação utilizando técnicas pertencentes a área da inteligência artificial (PASCHEK *et al.*, 2017).

A Figura 1 apresenta uma clara evolução do interesse da aplicação de inteligência artificial e aprendizado de máquina no contexto de melhoria na gestão de processos.

Stjepić *et al.* (2020) levanta uma série de segmentos de negócios suportados por tecnologias que apoiam a transformação digital. A Inteligência Artificial se configura como uma das mais promissoras.

Este trabalho propõe métodos para predizer o tempo restante necessário para que um processo seja concluído. A predição de tempo restante para conclusão de forma satisfatória,

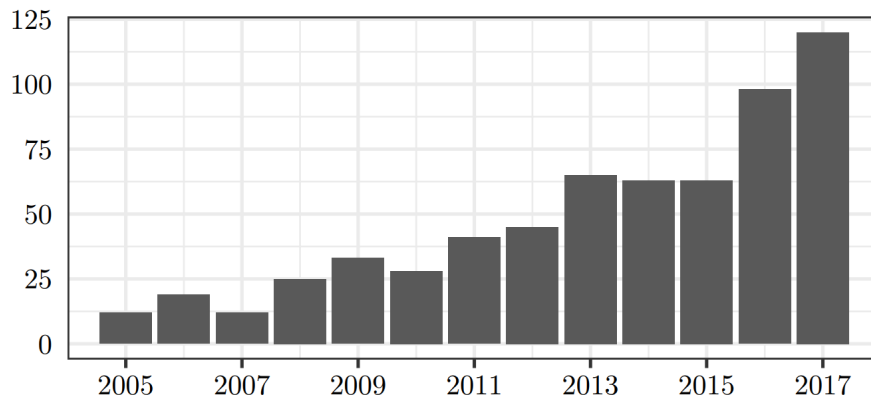
Figura 1 – Influência do aprendizado de máquina e IA como ferramentas de apoio à gestão de processos.



Fonte: Paschek *et al.* (2017).

pode evitar problemas como possíveis gargalos, dependências cíclicas no processo de negócio, esperas durante um grande intervalo tempo, além de dar suporte à tomada de decisão para otimização dos processos. Não somente houve a intenção vinda de órgãos e gestores, mas também segue existindo um crescimento de pesquisas relacionadas a assuntos que envolvem predição, otimização e automatização de processos, como mostra a Figura 2.

Figura 2 – Número de estudos publicados ao longo do tempo.



Fonte: Verenich *et al.* (2019).

Adicionalmente, existe uma importante iniciativa chamada Cientista Chefe, do Governo do Estado do Ceará, que acumula bilhões de registros de dados de diferentes naturezas. A Secretaria da Fazenda do Estado do Ceará (SEFAZ-CE), especificamente, possui registros das Notas Fiscais Eletrônicas (NF-e), dos Cupons Fiscais Eletrônicos (CF-e), da Escrita Fiscal Digital (EFD) entre diversos outros. Esse enorme volume de informação,

associado à sua variedade, e à rapidez com que esses dados são gerados, os classificam como Big Data. O processamento desses dados para o direcionamento de políticas públicas requiere formas inovadoras de análise, e o seu grande valor encontra-se exatamente nas possibilidades de desenvolvimento de medidas preventivas, corretivas em “tempo real”, bem como no fornecimento de informações precisas importantes para o planejamento de políticas públicas.

Esse projeto, que também financiou esta pesquisa, objetiva um ponto muito importante na área de monitoramento preditivo de processos, que propõe utilizar técnicas avançadas e desenvolver novas ferramentas de análise que consigam caracterizar, através do cruzamento entre estes dados de naturezas distintas, as correlações intrínsecas existentes entre eles no tempo e no espaço, visando a detecção preditiva de comportamentos emergentes e/ou sincronizados, relações de causalidade e a determinação de medidas de resiliência interdependentes, com foco na proposição de políticas públicas¹.

1.1 Questões de pesquisa

Dessa forma, apoiando-se nos propósitos atuais de melhoria da gestão de processos e na evolução das pesquisas nessa área, este trabalho objetiva também contribuir nessa evolução e responder às seguintes questões de pesquisa (QP):

QP 1:

Quais arquiteturas de aprendizado profundo conseguem prever o tempo restante para conclusão de um processo de negócio com menor erro absoluto médio?

QP 2:

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

QP 3:

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

1.2 Contribuições e Publicações

A solução desenvolvida neste trabalho também entrega três principais contribuições:

C1:

¹ <<https://www.funcap.ce.gov.br/cientista-chefe-descricao-dos-programas/>>

Realizar novas experimentações com arquiteturas de aprendizado profundo baseadas em redes recorrentes para prever o tempo restante para conclusão de um processo de negócio, que superam soluções estado da arte;

C2:

Validar as arquiteturas propostas com dois conjuntos de dados públicos e três conjuntos de dados privados;

C3:

Disponibilizar os experimentos realizados para possam ser reproduzidos.

Os resultados utilizando os conjuntos de dados públicos foram publicados no seguinte artigo:

- “Predição de tempo restante para conclusão de processos de negócio utilizando aprendizado profundo” nos Anais do XXXVIII Simpósio Brasileiro de Bancos de Dados (SBBD), realizado pela Sociedade Brasileira de Computação (SBC) em 2023. O trabalho de Silva *et al.* (2023) possui extrato B2 segundo a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e o Conselho Técnico-Científico da Educação Superior (CTC-ES) (BARATA, 2017).

1.3 Objetivos

Predizer o tempo de conclusão de uma instância de processo, sendo essa com precisão, contribui na prevenção de esperas incertas, descoberta de gargalos em processos e a assistência a sistemas de recomendação.

1.4 Estrutura do trabalho

Este trabalho se organiza da seguinte forma: O Capítulo 2 diz respeito à fundamentação teórica. Nele é possível identificar e entender os conceitos-chave utilizados neste trabalho. No Capítulo 3, são descritos os trabalhos que serviram de embasamento teórico para fundamentação e comparação com este trabalho. Em seguida, no Capítulo 4, tem-se a abordagem de métodos de coleta de dados, engenharia de características e procedimentos metodológicos. Seguindo o trabalho, no Capítulo 5, tem-se os resultados obtidos após a execução das etapas descritas no capítulo anterior. Por fim, no Capítulo 6, tem-se uma revisão geral de todo o trabalho desenvolvido e pontuações de possíveis melhorias e utilização de outras abordagens para a

solução do problema apresentado.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são explanados os principais conceitos que auxiliarão na melhor compreensão da abordagem tratada neste trabalho, como: processos de negócio e seus componentes e representações, organização de etapas de um processo de negócio, aprendizado de máquina, aprendizado profundo, séries temporais, modelos preditivos, arquiteturas de redes neurais recorrentes, blocos de autoatenção, métodos e métricas de avaliação.

2.1 Processo de Negócio

Um processo de negócio, é um conjunto de um ou mais procedimentos, tarefas, instâncias de processo ou atividades executadas de acordo com uma ordem predefinida em que a etapa atual depende da conclusão etapa anterior e que possa gerar um produto ou um serviço como resultado. Um processo de negócio também pode ser entendido como um fluxo de trabalho. Independente da forma em que um processo se apresenta, ou até mesmo a organização de suas de atividades, o que é de fato importante, é que em todas as definições mencionadas, o propósito do processo de negócio é de se chegar a um objetivo conhecido por meio de etapas definidas com clareza, organização e respeito às dependências entre etapas e à representação abstrata do modelo real (PONSARD; 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 do momento (tempo) da sua execução e uma tarefa relacionada.

2.1.1 Registros (logs) de evento

“Avanços recentes na pesquisa de mineração de processos permitem descobrir, analisar e melhorar os processos de negócios com base em dados de eventos. Atividades executadas por pessoas, máquinas e software deixam rastros nos chamados registros de eventos.” (AALST, 2012)

As etapas executadas em uma instância de processo de negócio são geralmente registradas pelas aplicações como *logs* de eventos. A Tabela 1 apresenta um exemplo de um registro (*log*) de eventos de um processo. Cada linha do registro corresponde a um evento identificado pelo atributo *Atividade* que iniciou 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 (ID) do processo. A finalização de uma instância de processo de negócio é dada quando um estado “Fechado” é alcançado. E 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 para conclusão de um processo 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 t que falta para atingir a atividade “Fechar”

Tabela 1 – Estrutura de um *log* de evento.

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

Fonte: Elaborado pelo autor.

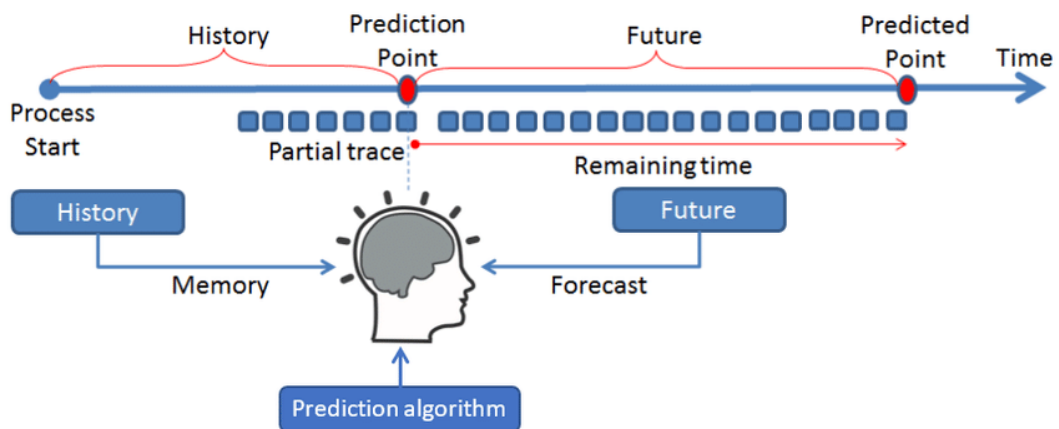
Ainda evidenciando a organização que um *log* de evento deve ter, é de grande importância que os *logs* estejam ordenados, da primeira até a última atividade. Tal ordenação garante o melhor entendimento da execução das etapas de um processo, suas dependências e casualidades. Para isso ocorrer, complementarmente, em diversos exemplos de modelos de processos, existem informações de registro de data e hora em cada instância de processo (AALST, 2016).

Além desses dados, pode-se gerar outras informações em *logs* de eventos, como quando se registra o tempo de duração que uma instância levou para ser encerrada e o tempo que passou desde que a primeira instância do processo foi criada, essas informações também foram utilizadas neste trabalho.

2.2 Monitoramento preditivo de processos

Dada uma representação da execução parcial de uma instância de processo (*trace*), é possível prever um comportamento dos próximos eventos de um processo no futuro. Um exemplo é a predição do tempo restante para conclusão de uma instância de processo. Esta abordagem está esboçada na Figura 3. Um ponto de previsão é o momento em que a previsão ocorre. Um ponto previsto é um ponto no tempo no futuro onde ocorre a previsão. Uma previsão é, portanto, baseada no conhecimento do modelo ou algoritmo preditor sobre o histórico da execução do processo até o ponto de previsão e do futuro até o ponto previsto. O histórico é garantido pela memória do preditor e o último ponto (*predicted point*) é baseado na previsão do preditor, que tem como função, prever o futuro com base na tendência e sazonalidade, ou seja, apoia-se na análise de padrões. Finalmente, a previsão é realizada com base em um método de raciocínio. Como nos processos da realidade a quantidade de incerteza aumenta com o tempo, a tarefa de previsão torna-se mais difícil e geralmente menos precisa.

Figura 3 – Visão geral do monitoramento preditivo de processos.



Fonte: Verenich *et al.* (2019).

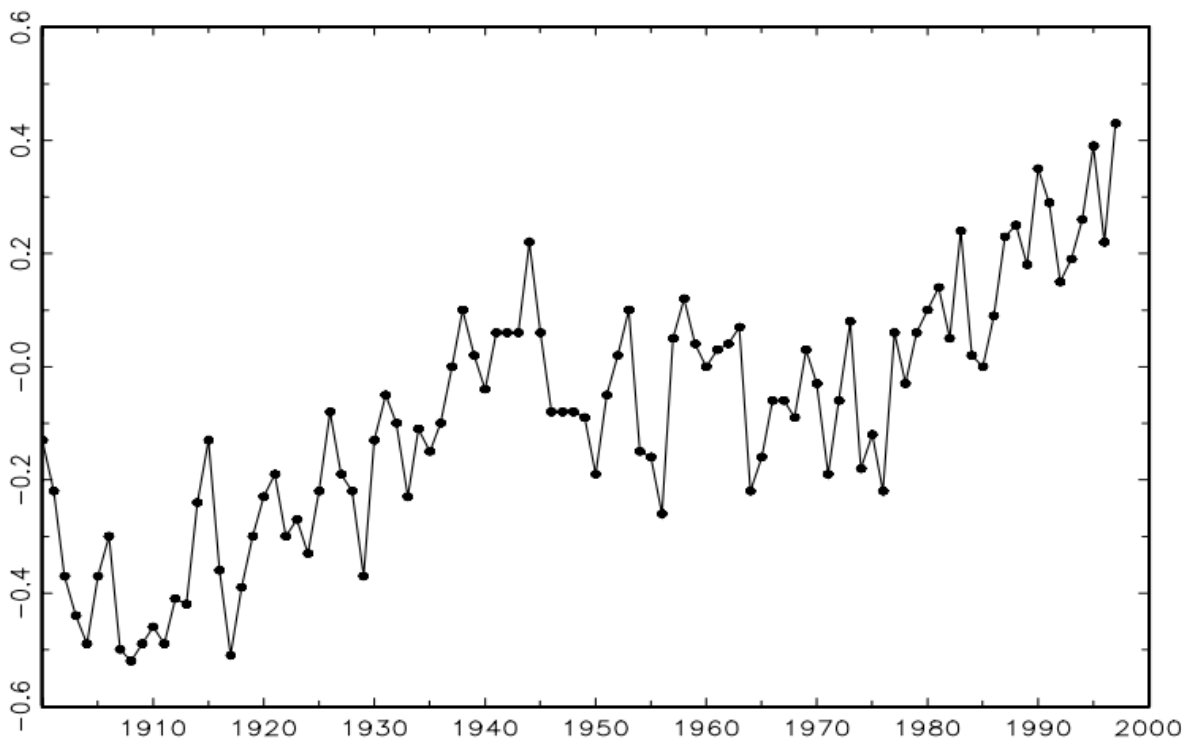
2.3 Série temporal

Uma série temporal é uma sequência de observações registradas ao longo do tempo. Quase invariavelmente, as observações que estão em períodos aproximados são mais fortemente correlacionadas do que observações amplamente separadas. O uso no tempo t de observações disponíveis de uma série temporal para prever seu valor em algum tempo futuro $t + l$ pode fornecer uma base para planejamento econômico e de negócios, planejamento de produção, controle de estoque e produção, e controle e otimização de processos industriais. Uma série

temporal S é uma coleção ordenada de L pares de uma medida e um instante de tempo, tal que, $S = (s_1, t_1), (s_2, t_2), \dots, (s_L, t_L)$, onde $s_i \in \mathbb{R}$, e t_1 até t_L são instantes de tempo para alguma medida s_1 até s_L (MAINDONALD; BRAUN, 2006; BOX *et al.*, 2015; TAN *et al.*, 2021).

Para cada registro de tempo, tem-se um valor de característica vinculado a ele. Analisando o comportamento desses dados como mostra a Figura 4, é possível fazer inferências a respeito do comportamento futuro dos dados. Por exemplo, o aumento da temperatura do planeta Terra ao longo dos últimos anos.

Figura 4 – Desvios de temperatura global média anual (1900-1997) em graus centígrados



Fonte: Shumway *et al.* (2000).

2.4 Engenharia de características

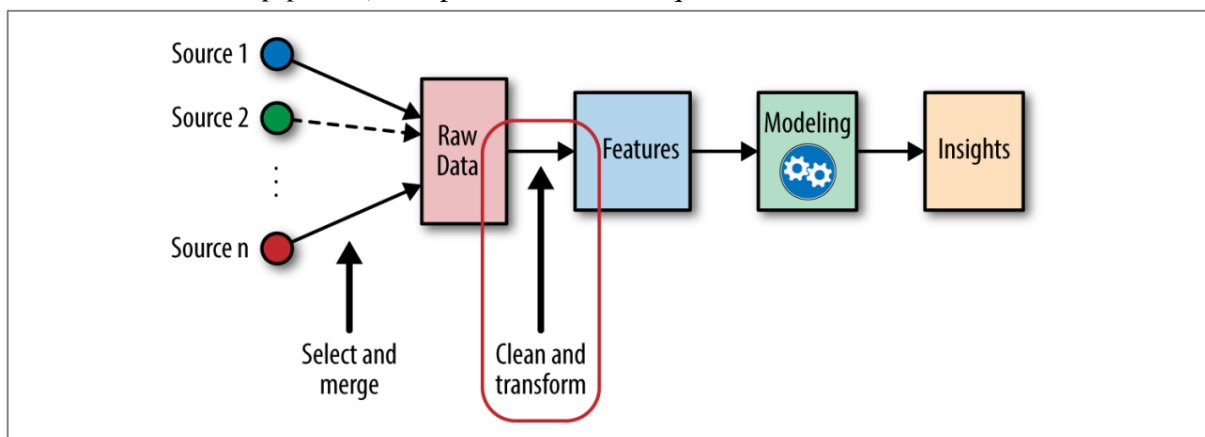
Uma característica é uma representação numérica de dados brutos, seja um valor como preço, altura, coordenada geográfica ou até mesmo um dado categórico. Há muitas maneiras de transformar dados brutos em dados numéricos, e é por isso que as características podem representar muitas coisas. Naturalmente, as características devem derivar do tipo de dados que está disponível, uma característica numérica representando um dado.

É interessante entender bem que as características também estão vinculadas ao modelo; alguns modelos de aprendizado de máquina são mais apropriados para alguns tipos de

características e vice-versa. As características corretas são relevantes para solucionar o problema em questão e devem ser claras para que o modelo de aprendizado de máquina possa ingeri-los. A engenharia de características é o processo de formular as características mais apropriadas conforme os dados, o modelo e o problema a ser resolvido. A quantidade de características também é importante. Se não houver recursos informativos suficientes, o modelo não conseguirá resolver um problema eficientemente. Se houver muitas características ou se a maioria delas forem irrelevantes, o modelo geralmente será mais difícil de treinar, irá “aprender” menos, ou seja, algo pode dar errado no processo de treinamento e isso afeta o desempenho do modelo.

De modo geral, as características e os modelos de aprendizado de máquina ficam “entre” os dados brutos e as intuições desejadas, conforme esquematizado na Figura 5.

Figura 5 – Etapas em que normalmente se encontram a engenharia de *features* no fluxo de trabalho (*pipeline*) do aprendizado de máquina



Fonte: Zheng e Casari (2018).

Em um esquema de modelagem de um processo de aprendizado de máquina, escolhe-se não apenas o modelo, mas também as características. A escolha das características altera o desempenho do modelo e a escolha do modelo afeta o uso das características. Boas características tornam a etapa de modelagem subsequente fácil e o modelo resultante mais capaz de resolver o problema desejado. Características mal escolhidas podem exigir um modelo muito mais complexo para atingir o mesmo nível de desempenho. As questões que envolvem aprendizado de máquina e aprendizado profundo serão discutidas em seguida.

2.5 Aprendizado de máquina

A definição formal de aprendizado de máquina é dita como: “Um programa de computador aprende a partir de uma experiência E com respeito a alguma classe de tarefas T

medida por uma performance P , se esse desempenho na tarefa T medida por P , melhora com a experiência E .” (MITCHELL, 1997).

Em tempos atuais, existe um recurso de grande abundância e constante crescimento presente na tecnologia moderna, esses artefatos são os dados.

A razão principal da rápida expansão da quantidade de dados, é o extenso grau em que eles são criados, compartilhados e utilizados nos últimos tempos. Graças à digitalização, transformação de sinais analógicos em digitais, alcançaram enorme popularidade no início de 1990 e segue em grande velocidade (MAURO *et al.*, 2016). Na metade do último século, não só a quantidade de dados cresceu, mas também, técnicas de IA e poder computacional evoluíram bastante e conseguiram manipular essa quantidade de dados gerados. Uma técnica popular é o aprendizado de máquina, uma área de pesquisa importantíssima da computação, subárea da IA que utiliza como ferramenta, algoritmos de aprendizado automático que por sua vez, gera conhecimento a fim de realizar previsões ao invés de necessitar da ação humana para criar regras manualmente após analisar uma quantidade grande de dados. Técnicas de aprendizado de máquina, em geral, oferecem eficiência na geração de conhecimento, melhoria de modelos preditivos e tomada de decisões (RASCHKA; MIRJALILI, 2017).

Apesar de existirem três tipos de aprendizado de máquina: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço, este trabalho visa utilizar o aprendizado de máquina supervisionado, dado que o cenário proposto no Capítulo 1 se assemelha com a definição do domínio de problema de aprendizado de máquina supervisionado. Segundo Raschka e Mirjalili (2017), os domínios do problema de aprendizado de máquina supervisionado possuem as seguintes características:

- Os dados utilizados possuem rótulos (*labels*) que definem as características dos dados conhecidos;
- Resposta direta. Dado uma entrada, a previsão é feita imediatamente;
- Predição ou aproximação de um resultado decorrente de um conjunto de características que compõem dados conhecidos.

2.5.1 *Aprendizado supervisionado*

Problemas encontrados na área de aprendizado supervisionado ou aprendizado preditivo constituem a maioria das tarefas que o aprendizado de máquina pode ser usado para resolver hoje. Essas problemáticas são encontradas em uma ampla gama de situações e tipos

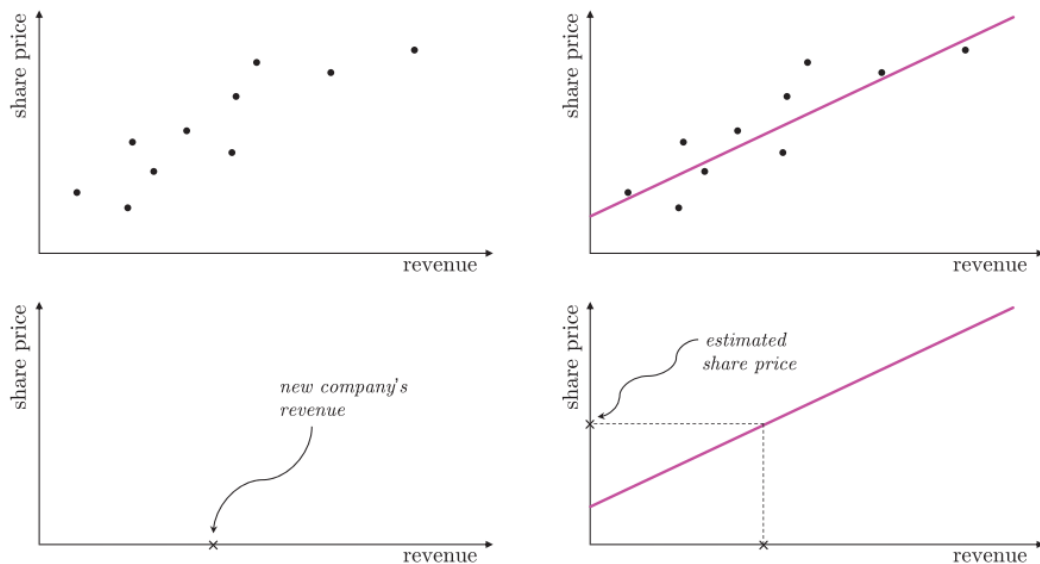
de dados. Existem os dois principais problemas de aprendizagem preditiva: problemas de classificação e problemas de regressão (WATT *et al.*, 2016). Esse último, problemas de regressão, é foco deste trabalho.

Problemas de regressão

“Utilizada na previsão de resultados não discretos, também chamada de análise de regressão. A análise de regressão, recebe uma série de variáveis preditoras (explicativas) e uma variável de resposta contínua (resultado ou objetivo), e tenta encontrar um relacionamento entre aquelas variáveis que permite prever um resultado” (RASCHKA; MIRJALILI, 2017).

A Figura 6 mostra como um algoritmo básico (linear) de regressão funciona. Sempre haverá a relação entre características do dado e um respectivo valor (rótulo). Novos dados que possuem características semelhantes às já existentes, geram valores também semelhantes.

Figura 6 – No painel superior esquerdo, um conjunto de dados de treinamento cuja fonte de dados são informações de preços de brinquedos de dez empresas associadas a valores de receita. No painel superior direito, um modelo linear é ajustado aos dados. Esta linha de tendência modela a trajetória geral dos pontos e pode ser usada para previsão no futuro, conforme mostrado no painel inferior esquerdo e inferior direito



Fonte: Watt *et al.* (2016).

Técnicas de aprendizado de máquina para resolução de problemas de regressão

A listagem a seguir descreve brevemente algumas características de algoritmos e modelos de regressão mencionados ou utilizados neste trabalho:

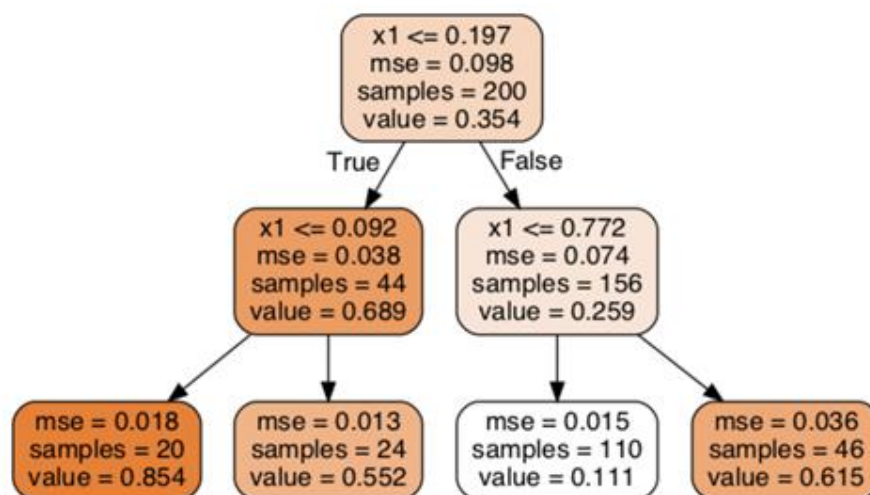
Regressão Linear (*Linear Regression*)

A regressão linear é um dos modelos mais simples de regressão utilizados em aprendizado de máquina (Figura 6). Ele modela a relação entre a variável independente (características) e a variável dependente (rótulo). No caso de regressão linear simples, uma variável independente e uma variável dependente são envolvidas nessa relação (ACHARYA *et al.*, 2019).

Regressão por árvore de decisão (*Decision Tree Regressor*)

A árvore de decisão é um exemplo de modelo que pode ser usado tanto para problemas de classificação quanto para problemas de regressão. A árvore de decisão segue uma abordagem por níveis de modo ganancioso ou recursivo. Ela divide um conjunto de dados em subconjuntos decrescentes e, ao mesmo tempo, uma nova subárvore de decisão é criada. O resultado é uma árvore de nós de decisão e nós-folha (último nível da árvore). A Figura 7 ilustra a estrutura da árvore de decisões. No último nível, temos os valores preditos, o erro médio e a quantidade de amostras por nó-folha. O algoritmo retorna para cada predição, o melhor valor predito encontrado (ACHARYA *et al.*, 2019).

Figura 7 – Esquema de uma árvore de decisão



Fonte: Sought (2021).

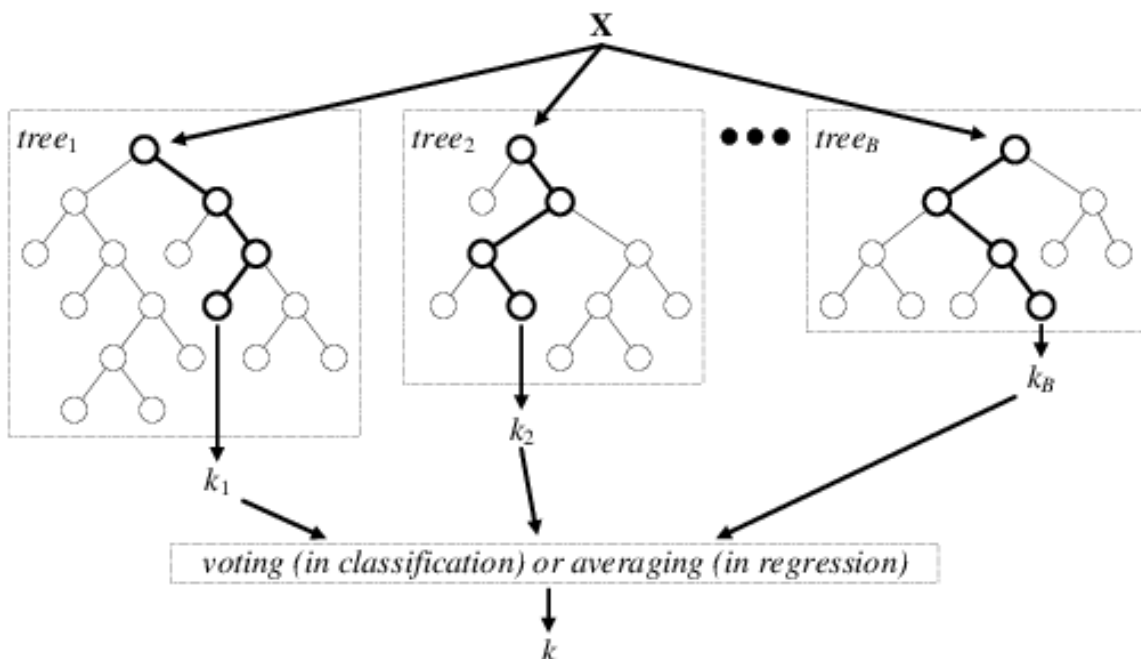
Regressão por floresta aleatória (*Random Forest Regressor*)

A regressão por floresta aleatória é um tipo de modelo que faz predições pela combinação

de uma sequência de modelos mais básicos. Cada modelo básico é uma árvore de decisão e o resultado do algoritmo de floresta aleatória é a saída cumulativa de todas as árvores de decisão. Essa técnica de usar vários modelos para obter uma melhor previsão é chamado de agrupamento de modelos. Todos os modelos básicos são construídos de forma independente usando uma subamostra diferente dos dados. O algoritmo floresta aleatória é muito eficiente quando se lida com dados tabulares com características numéricas ou características categóricas com menos do que centenas de categorias (sendo a configuração dos dados utilizados neste trabalho). Ao contrário dos modelos lineares, um algoritmo de floresta aleatória consegue capturar a interação não-linear entre as características e os rótulos.

A arquitetura do modelo floresta aleatória é ilustrado na Figura 8. Cada submodelo fornece uma previsão para um subconjunto de dados. Uma vez que todas as previsões parciais são feitas, uma única saída correspondente à média de todas as previsões é retornada.

Figura 8 – Esquema do algoritmo de floresta aleatória.



Fonte: Suparwito (2019).

Avaliação de modelos de regressão

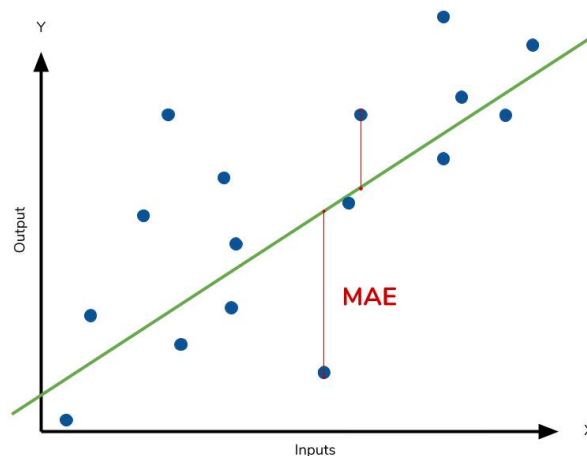
Quando o rótulo ou saída da previsão é um valor numérico, o método mais comum para caracterizar a capacidade de precisão do modelo é usar o erro médio absoluto MAE, uma

métrica que diz respeito aos resíduos do modelo. Essa métrica tem função de observação da diferença entre os valores reais e o preditos pelo modelo de aprendizado de máquina. O MAE, uma métrica intimamente relacionada à precisão da previsão é calculada 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. Formalmente, o MAE, de um estimador \hat{y} para N amostras é descrito na Equação 2.1:

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (2.1)$$

O valor resultante é geralmente interpretado como o quão (em média) os resíduos são distantes de zero ou o quão a distância média entre os valores observados (reais) estão das previsões do modelo. A Figura 9 mostra o que pode-se entender por resíduo (LARSON; FARBER, 2019).

Figura 9 – Descrição gráfica do MAE. A linha verde representa as previsões de um modelo e os pontos azuis representam os dados reais



Fonte: Kumar (2023)

Então, num cenário perfeito, a soma de todos os termos na Equação 2.1 tenderá a 0, ou seja, uma função não-negativa.

2.6 Aprendizado profundo

O aprendizado profundo (*Deep Learning* (DL)) é um subcampo específico do aprendizado de máquina: uma nova abordagem sobre representações de aprendizado a partir de dados

que enfatiza o aprendizado mais complexo por meio de camadas sucessivas de representações mais significativas dados. No aprendizado profundo, essas representações em camadas são aprendidas por meio de modelos chamados redes neurais, estruturados em camadas empilhadas umas sobre as outras.

As redes neurais profundas fazem um mapeamento entre entrada e destino por meio de uma sequência de transformações de dados simples e que essas transformações de dados são aprendidas pela exposição a exemplos (conjunto de características). A especificação do que uma camada faz com seus dados de entrada é armazenada nos pesos da camada e a transformação implementada por uma camada é parametrizada por seus pesos. A capacidade de aprender significa encontrar um conjunto de valores para os pesos de todas as camadas em uma rede, de modo que a rede mapeie corretamente as entradas de exemplo para seus alvos (*labels*) associados (Figura 10A). Para controlar a saída de uma rede neural, é necessário medir o quão longe essa saída está do que se espera. Este é o trabalho da função de perda (*loss function*) da rede, também chamada de função objetivo. A função de perda manipula as previsões da rede e o alvo (*ground truth*) e calcula uma pontuação de distância, capturando o desempenho da rede neste exemplo específico (Figura 10B).

Usa-se essa pontuação como um sinal de resposta (*feedback*) para ajustar um o valor dos pesos, de modo a diminuir a perda para o exemplo atual. Esse ajuste é o trabalho do otimizador, que implementa o que é chamado de algoritmo *backpropagation*: o algoritmo central no aprendizado profundo (Figura 10C) (CHOLLET, 2017).

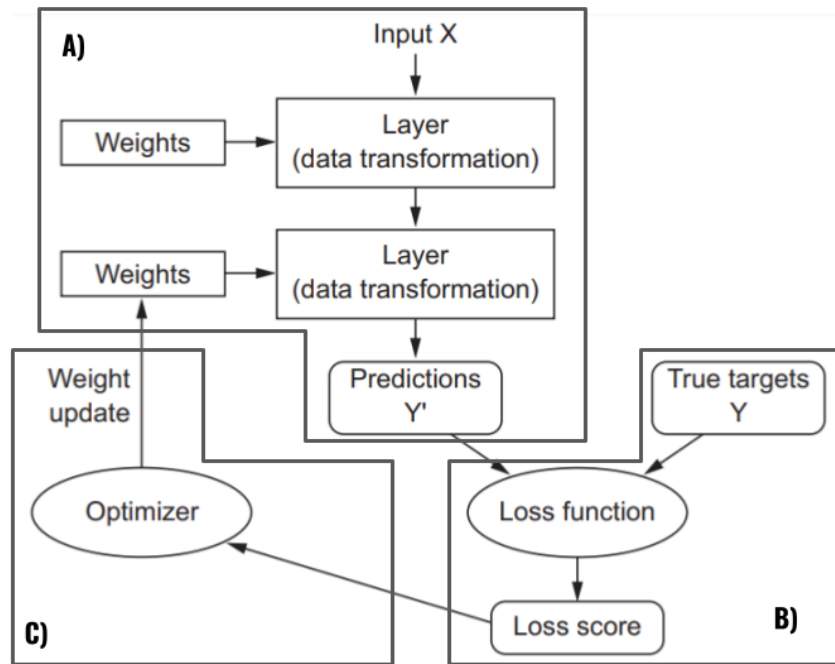
2.6.1 Redes Neurais Recorrentes

Uma rede neural recorrente (RNN) processa sequências iterando pelos elementos da sequência e mantendo um estado contendo informações relativas ao que se viu até o momento. Com efeito, uma RNN é um tipo de rede neural que possui um *loop* interno.

Dada uma sequência de entrada $x = (x_1, \dots, x_T)$, as RNNs calculam a sequência de vetor oculto $h = (h_1, \dots, h_T)$ e a sequência de vetor de saída $y = (y_1, \dots, y_T)$ iterando equações 2.2 e 2.3 com t variando de 1 a T .

$$h_t = \mathcal{H}(W_{xh} \times x_t + W_{hh} \times h_{t-1} + b_h) \quad (2.2)$$

Figura 10 – Esquema de funcionamento de um modelo de aprendizado profundo

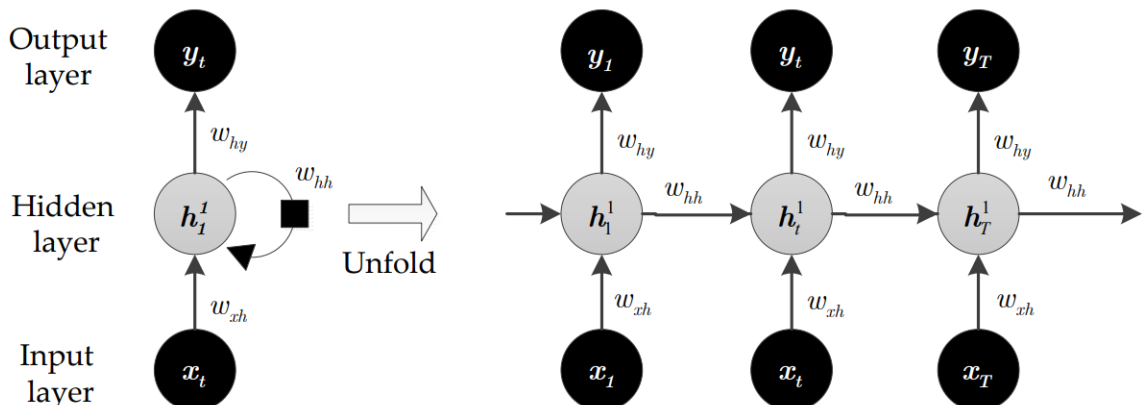


Fonte: adaptado de Chollet (2017).

$$y_t = W_{hy} \times h_t + b_y \quad (2.3)$$

Onde os termos W_{xh} , W_{hh} , W_{hy} denotam respectivamente as matrizes de peso das camadas de entrada, a camada oculta e a camada de saída. Os termos b_h e b_y denotam, respectivamente, vetores de viés (*bias*) oculto e de saída. O termo \mathcal{H} é a função de ativação da camada oculta.

Figura 11 – RNN e a representação sequencial do seu *loop* de sua computação. Os neurônios artificiais recebem informações de outros neurônios em intervalos de tempo anteriores.



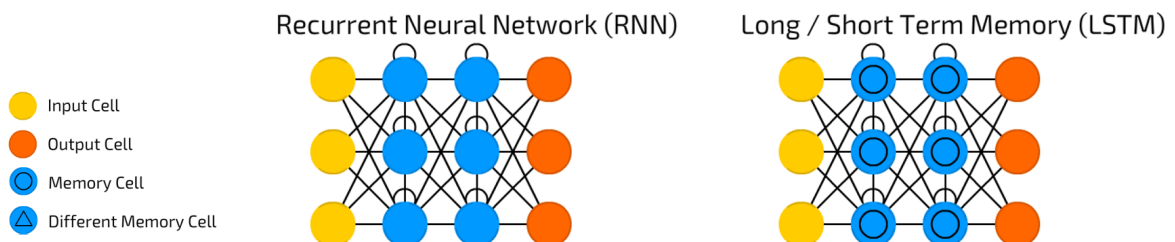
Fonte: adaptado de Husein e Chung (2019).

O estado do RNN é redefinido entre o processamento de duas sequências diferentes e independentes (como dois processos de negócio diferentes), portanto, se considera uma sequência como um único ponto de dados: uma única entrada para a rede. O que muda é que esse ponto de dados não é mais processado em uma única etapa; em vez disso, a rede faz um *loop* interno sobre os elementos de sequência (CHOLLET, 2017).

2.6.1.1 Long Short Term Memory

Uma variação de RNN é a LSTM, que consegue lidar com o problema das “dependências de longo prazo”, propôs a célula LSTM, que melhora a capacidade de lembrar da célula recorrente padrão. Ao invés de células recorrentes, a arquitetura de uma LSTM consiste em uma série de blocos conectados também recorrentemente, tal como a RNN, porém, chamados de blocos de memória. A seção do codificador pega uma entrada e a mapeia para um espaço disponível. O decodificador pega esse espaço disponível e o mapeia para uma saída (GALPHADE *et al.*, 2021).

Figura 12 – Diferença entre as arquiteturas de RNN e LSTM.



Fonte: adaptado de Jiao *et al.* (2020).

Além disso, a LSTM introduz os sinais de controle para orientar e mitigar potencialmente os desafios para os cálculos de gradiente, evitando que as informações mais antigas desapareçam gradualmente durante o processamento (YU *et al.*, 2019; SALEM, 2022; CHOLLET, 2021)

2.6.2 Mecanismo de auto-atenção

Existem muitas formas diferentes de atenção que se possa considerar, mas todas elas começam calculando pontuações de importância para um conjunto de características, com pontuações mais altas para características mais relevantes e pontuações mais baixas para características menos relevantes.

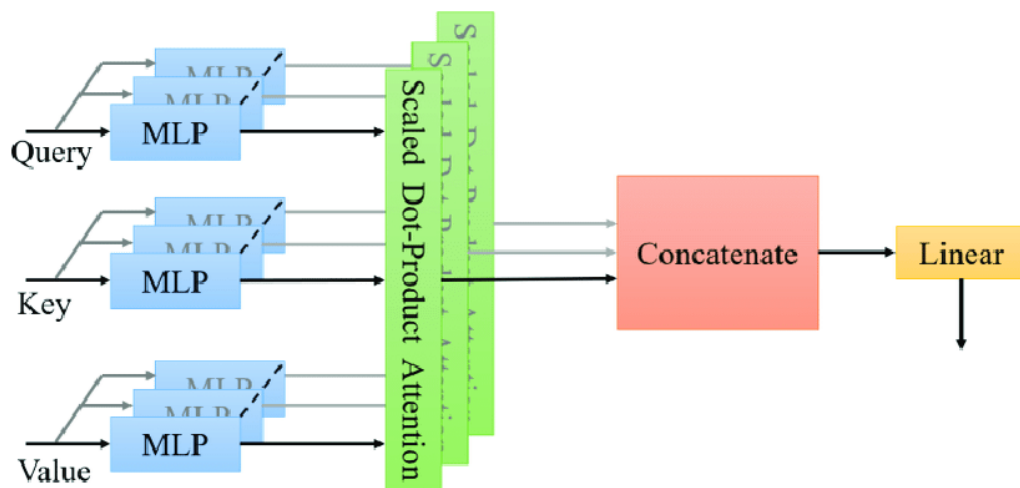
Duas formas de atenção a características comuns na literatura são listadas a seguir: **Max pooling** analisa um conjunto de características em uma região espacial e seleciona apenas um recurso para manter como importante. Essa é uma forma de atenção do tipo “tudo ou nada”: que mantém a características mais importante e descarta o restante.

Term Frequency – Inverse Document Frequency (TF-IDF) atribui pontuações de importância aos *tokens* com base na quantidade de informações que diferentes *tokens* provavelmente carregam. *Tokens* importantes são aumentados enquanto *tokens* irrelevantes desaparecem. Essa é uma forma contínua de atenção.

Como essas pontuações devem ser calculadas e como essas pontuações devem ser manipuladas, varia de abordagem para abordagem. (CHOLLET, 2021).

A Figura 13 ilustra a arquitetura de um bloco de atenção. A consulta (*Query*, Q) diz respeito ao dado de entrada, uma frase, um processo de negócio, uma sequência, toda informação que existiu previamente. A chave (*Key*, K é a unidade básica dessa sequência (palavra, instância de processo). Por fim, o valor *Value*, V que representa a similaridade entre um elemento K na consulta Q .

Figura 13 – Arquitetura de um bloco de auto-atenção.



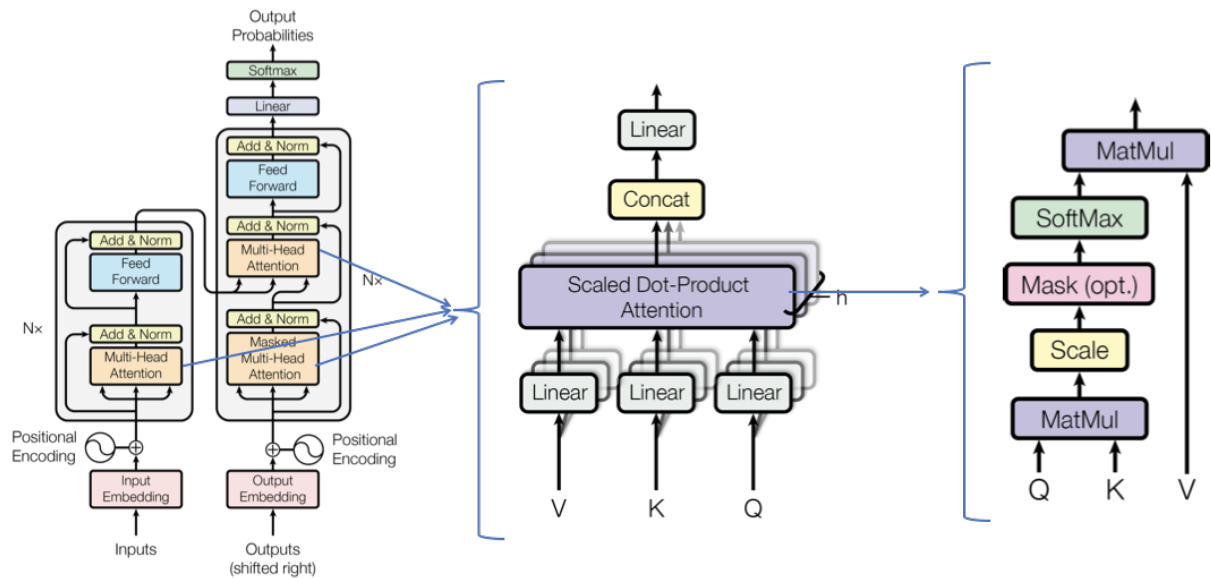
Fonte: adaptado de Jiao *et al.* (2022).

Dessas variáveis, tem-se o produto escalar, o qual é um mecanismo de atenção onde os produtos escalares são reduzidos por $\sqrt{d_k}$. Formalmente, com Q , K e V calculamos a atenção A como:

$$A(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.4)$$

Assumimos que tanto q quanto k são vetores d_k dimensionais cujos componentes são variáveis aleatórias independentes com média = 0 e variância = 1, então seu produto escalar, $q \cdot k = \sum_{i=1}^{d_k} u_i v_i$, tem média = 0 e variância = d_k . Como é interessante que esses valores tenham variância = 1, dividi-se por $\sqrt{d_k}$ (Ilustrado na Figura 14 (VASWANI *et al.*, 2017)).

Figura 14 – Bloco de auto-atenção (ao centro) compondo arquitetura de um modelo de *transformer*.



Fonte: Vaswani *et al.* (2017).

3 TRABALHOS RELACIONADOS

Este capítulo apresenta outras problemáticas encontrados na literatura e trabalhos que propõem a resolução de problemas semelhantes ao problema de predição de tempo restante para conclusão de um processo de negócio. São seccionadas as descrições dos trabalhos, as técnicas utilizadas, representação dos dados e objetivos. Por fim, uma comparação entre as abordagens utilizadas neste trabalho e nos demais textos referenciados nas seções seguintes.

3.1 Trabalho Relacionado - LSTM para Regressão e Classificação

O trabalho desenvolvido por Tax *et al.* (2017)¹ investiga uma abordagem baseada em LSTM para construir modelos preditivos para uma ampla gama de tarefas de 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, que também propõe resolver o problema deste trabalho e outra para predição da próxima atividade. Esta camada de saída voltada para predição da próxima atividade refere-se a um problema de classificação, que também está contido no estudo de monitoramento preditivo de processos. Cada um dos tipos de atividades contidas nas instâncias de processo foi representado como um caractere de 2 *bytes*, representando o *trace* do processo como uma *string* e reduzindo, assim, quantidade de memória alocada na solução.

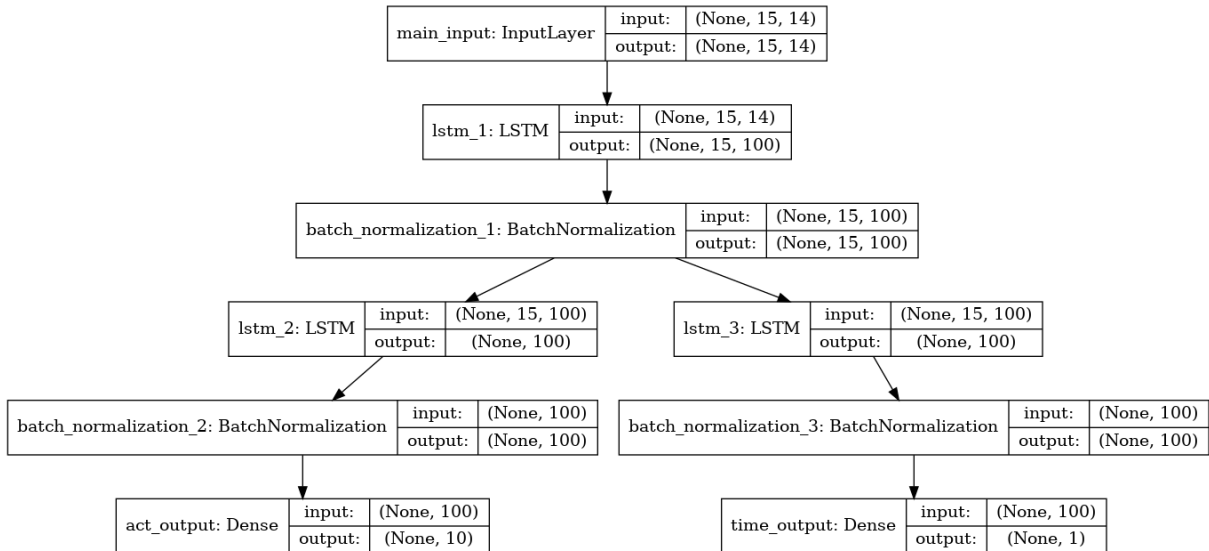
A representação dos dados temporais é apresentada em forma de ponto flutuante, remetendo a um nível de precisão maior na informação coletada. O conjunto de dados foi não necessariamente ordenado cronologicamente. O trabalho utilizou algumas instâncias (cerca de 60%) dos *logs* de eventos do conjunto de dados completos do *Helpdesk 17*. Para treinamento e avaliação do modelo, o conjunto de dados descrito em Verenich (2016) e o subconjunto “W” de Dongen (2012) foram utilizados. A partição segue a configuração de 2/3 dos primeiros processos ordenados cronologicamente como dados de treinamento e avalia as previsões de próxima atividade e de tempo no restante com 1/3 dos demais processos.

A arquitetura do modelo utilizado nesse trabalho é ilustrado na Figura 15. A dimensão da entrada do modelo é uma tupla do número máximo de etapas encontradas num processo e o número de atividades distintas (15, 14). A camada seguinte, de LSTM, recebe a entrada apresentada anteriormente e entrega para a camada seguinte, uma saída de tamanho 100. Dada a

¹ <<https://github.com/verenich/ProcessSequencePrediction>>

bifurcação, o caminho da esquerda resolve um problema de classificação dando qual o tipo da próxima atividade e o caminho da direita, retorna apenas um único valor que representa o tempo restante para conclusão do processo.

Figura 15 – Estrutura da arquitetura do modelo



Fonte: adaptado de Tax *et al.* (2017)

Dada a configuração da partição dos conjuntos de treino e testes citadas anteriormente, o modelo proposto nesse trabalho conseguiu obter dois valores de MAE importantes: 3,75 dias com os dados do conjunto *Helpdesk 17* utilizando uma camada de LSTM com 100 neurônios e 1,65 dia utilizando o conjunto de dados *BPI 12-W*, com a configuração de duas camadas LSTM com 100 neurônios cada.

3.2 Trabalho Relacionado - LSTM Para Regressão

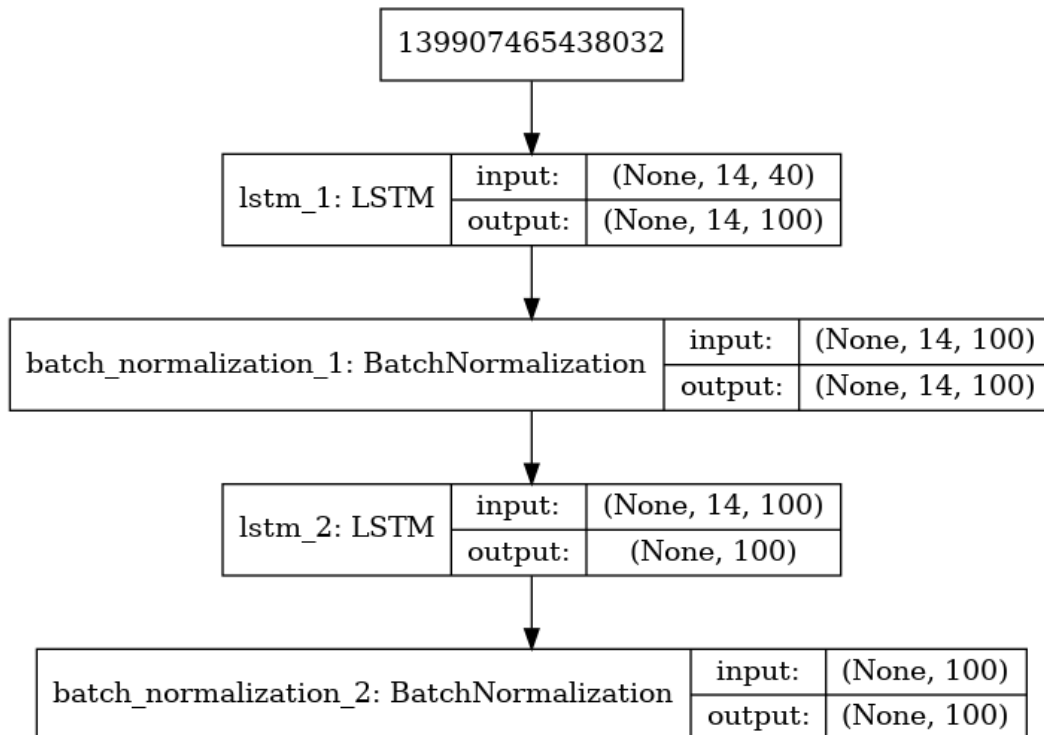
O modelo desenvolvido por Navarin *et al.* (2017)² explora informações como momento de início, momento de fim e dia da semana em que ocorreu o evento, além de representar cada atividade como uma codificação única (*One-hot Encoding*), a fim de produzir uma predição do tempo restante para conclusão de instâncias de processos em execução. Nesse trabalho, os conjuntos de dados de treino e teste considerados compreendem apenas aos gerados por processos completos. Essa escolha é motivada pelo fato de que o método proposto, se implementado em um ambiente de negócios real, deve prever um tempo para conclusão logo após a abertura do chamado. Se o processo não encerra, não há como prever a sua conclusão com precisão. O

² <https://github.com/nickgentoo/DALSTM_PM>

repositório³ desse trabalho não possui os dados de entrada para todos os experimentos, porém, no trabalho escrito, temos 15.682 instâncias de processos (cerca de 70% do conjunto de dados original).

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 Verenich (2016). 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* seguida de uma camada de saída, onde l e n são parâmetros de construção da rede. A Figura 16 esquematiza a arquitetura do modelo utilizado nesse trabalho. Similar ao caminho que resolve o problema de regressão do trabalho de Tax *et al.* (2017), esse trabalho utiliza uma configuração de LSTM com as representações das 14 atividades distintas em vetores de 100 posições.

Figura 16 – Arquitetura apresentada em trabalho



Fonte: adaptado de Navarin *et al.* (2017)

Esse trabalho também utiliza os dados do conjunto *Helpdesk* e *BPI 12* e obteve os seguintes valores de MAE para predição de tempo restante: 4,01 dias com os dados do conjunto *Helpdesk 17* utilizando $l = 5$ (cinco camadas de LSTM) com $n = 250$ e 7,04 dias utilizando o conjunto de dados *BPI 12*, com a configuração com $l = 4$ (quatro camadas) LSTM com $n = 100$ cada.

³ <https://github.com/nickgentoo/DALSTM_PM>

3.3 Trabalho Relacionado - *Embeddings* e *Transformers*

Bukhsh *et al.* (2021)⁴ utilizam os mesmos conjuntos de dados (Verenich (2016)) e propõem projetar uma arquitetura de aprendizado profundo com camadas de atenção que funcione bem em rastreamentos de processo longos. Além disso, os autores abordam a utilização de *embeddings* e *transformers* para categorizar as instâncias de processos, diferente dos demais autores citados nesta seção, que utilizaram técnicas de codificação fora das camadas da rede neural. Também utiliza da representação de uma instância de processo como uma sequência de atividades que antecedeu a atividade atual, seguido de um preenchimento que adéqua, em tamanho único, todas as sequências apresentadas no conjunto de dados organizados nesse trabalho.

Ao contrário dos trabalhos de Tax *et al.* (2017) e Navarin *et al.* (2017), esse trabalho utiliza apenas o conjunto de dados *Helpdesk 17*, o particionamento não obedece uma cronologia, porém é similar em proporção ao apresentado em Tax *et al.* (2017). O conjunto de validação é uma amostra aleatória gerada durante o treinamento do modelo proposto. A arquitetura da rede utilizada nesse trabalho é ilustrada na Figura 17 e conta com um mecanismo de atenção que dá maior prioridade aos dados que representam as atividades dos processos e aos dados que trazem informações sobre os dados temporais. Cada um dos tipos de dados é um tipo de entrada que se concatena após a entrada referente as atividades serem transformadas.

Como método de avaliação, esse trabalho também utiliza do MAE como métrica, porém, as predições são realizadas conforme o número do estágio do processo. O modelo proposto prediz o tempo restante para conclusão de um processo com todos os estágios de número $1, 2, \dots, n$ e após isso, é calculada a média entre cada uma dessas n predições, resultando num MAE de 3,913 dias (segundo o resultado presente no experimento ⁵ disponibilizado).

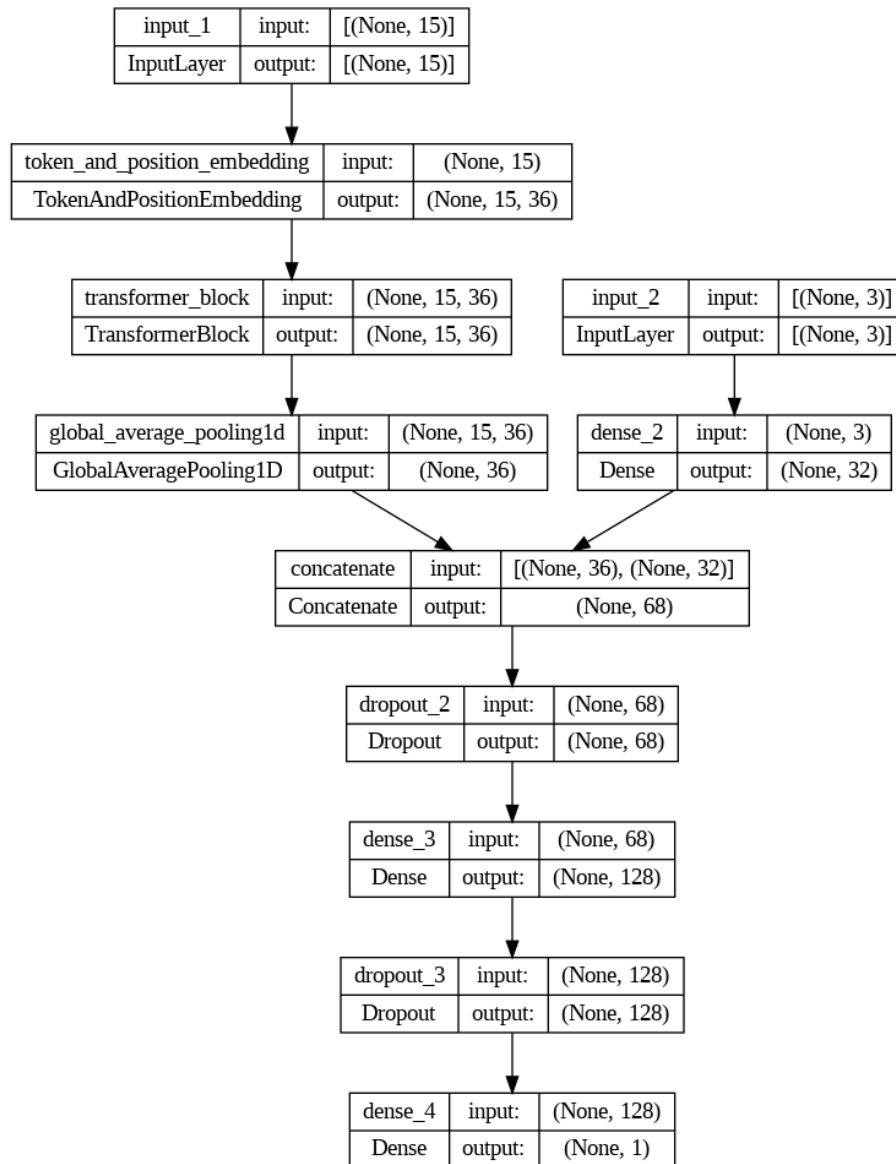
3.4 Trabalho Relacionado - LRCN

Park e 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

⁴ <<https://colab.research.google.com/drive/1tiOh2VS8yzOVON26CbmWn0oUn-dWAFhN>>

⁵ <<https://colab.research.google.com/drive/1tiOh2VS8yzOVON26CbmWn0oUn-dWAFhN>>

Figura 17 – Estrutura da arquitetura apresentada nesse trabalho



Fonte: adaptado de Bukhsh *et al.* (2021)

que contém informações sobre os desempenhos no processo de negócios. Os conjuntos de dados utilizados são principalmente das competições *Business Process Improvement (BPI) 12*, *BPI 13* e ao *Helpdesk 17*.

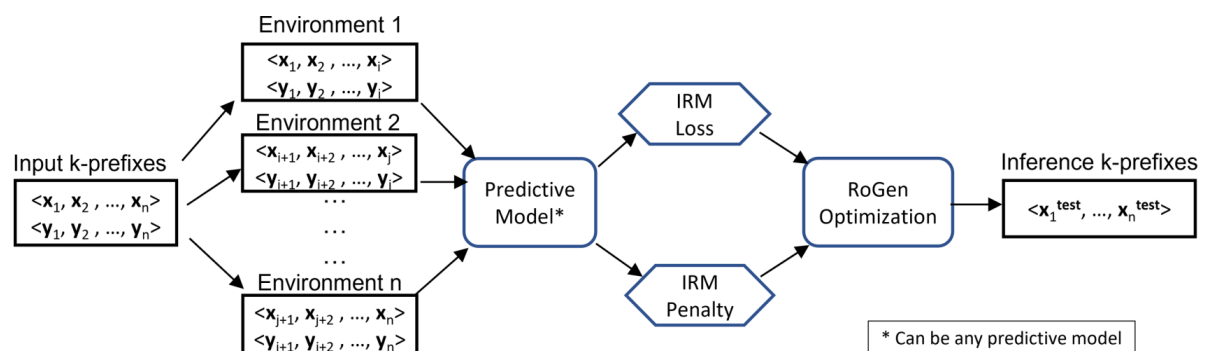
3.5 Trabalho Relacionado LSTM com IRM

Venkateswaran *et al.* (2021) também aborda a predição da próxima atividade e do próximo instante de tempo (*timestamp*). A proposta de predição do próximo instante de tempo para a atividade seguinte de um processo de negócio, toma como base um modelo de aprendizado profundo composto por duas camadas LSTM empilhadas e uma camada de saída

densa. Considerando que modelos de processos reais 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 ao longo do tempo, a solução proposta utiliza de 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 previsões mais precisas no decorrer do tempo. Os atributos são representados como um vetor de *embeddings* que possui um bom desempenho quando se considera a ordem dos dados. A solução foi avaliada com o conjunto de dados *Helpdesk 17* e dados de algumas versões do *BPI challenge* (*BPI 13*⁶ *BPI 15*⁷ *BPI 18*⁸ e *BPI 19*⁹).

A Figura 18 organiza graficamente o fluxo de trabalho de treinamento da abordagem projetada que usa Minimização de Risco Invariante, *Invariant Risk Minimization* (IRM) para treinar modelos robustos e generalizáveis. Os *k*-prefixos de entrada do *log* de eventos são primeiro divididos em diferentes conjuntos de dados (ambientes), os quais são posteriormente divididos em conjuntos de treinamento e teste. Os prefixos de entrada e os rótulos para as diferentes tarefas de previsão de cada ambiente de treinamento são então passados para o modelo preditivo RoGen que calcula a perda durante o treinamento (Perda de IRM) e penalidade (Penalidade de IRM) para cada ambiente *e*. A perda média e a penalidade em todos os ambientes são usadas para otimizar o modelo RoGen usando o algoritmo de otimização *Adam*. O modelo treinado pode então ser usado para inferência sobre novos *k* prefixos para previsões robustas de atividades futuras e seus instantes de tempo (*timestamp*).

Figura 18 – Fluxo de treinamento do RoGen



Fonte: adaptado de Venkateswaran *et al.* (2021)

⁶ <https://data.4tu.nl/articles/_/12714476/1>

⁷ <https://data.4tu.nl/collections/_/5065424/1>

⁸ <https://data.4tu.nl/articles/_/12688355/1>

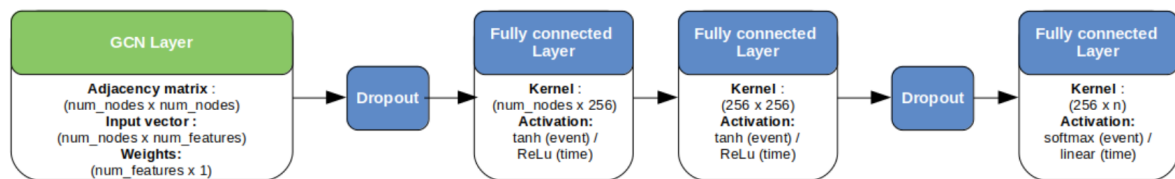
⁹ <https://data.4tu.nl/articles/_/12715853/1>

3.6 Trabalho Relacionado - GNN

Venugopal *et al.* (2021) apresenta uma solução para o problema de predição do próximo instante de tempo baseado em redes neurais de grafos (GNN). Similar à proposta de Tax *et al.* (2017), esse trabalho também contribui em experimentos de predição de qual tipo é a próxima atividade e o próximo instante de tempo (*timestamp*). A representação matricial do grafo é dada por vértices como estados (*e*) (Atividades) 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*. *BPI 12W*, um subconjunto do *BPI 12*, limitado apenas a eventos de itens de trabalho, que contém eventos executados manualmente.

A Figura 19 ilustra mais detalhes de como a arquitetura proposta por Venugopal *et al.* (2021) está organizada.

Figura 19 – Arquitetura de GNN para predição de tipo de atividade e predição *timestamp*. O valor de *n* na última camada denota o número de classes para o preditor de eventos e 1 para o preditor de tempo.



Fonte: adaptado de Venugopal *et al.* (2021)

Assim como Tax *et al.* (2017), a mesma arquitetura consegue resolver problemas distintos da área de PPM.

A Tabela 2 organiza as características de cada trabalho relacionado e o proposto neste.

Tabela 2 – Comparativo dos trabalhos relacionados.

Trabalho	Arquitetura	Codificação das atividades	Conjunto de dados	Tipo do problema
Tax <i>et al.</i> (2017)	LSTM	Caracteres Unicode	Helpdesk 17 BPI 12 W	Predição do tempo restante para conclusão Predição da próxima atividade
Navarin <i>et al.</i> (2017)	LSTM	OHE	Helpdesk 17 BPI 12	Predição do tempo restante para conclusão
Bukhsh <i>et al.</i> (2021)	Transformer	Token and Positional Embedding	Helpdesk 17	Predição do tempo restante para conclusão Predição da próxima atividade
Park e Song (2020)	LRCN	Representação Matricial	Helpdesk 17 BPI 12, BPI 13	Predição do tempo restante para conclusão Predição da próxima atividade
Venkateswaran <i>et al.</i> (2021)	LSTM	Catagórico	Helpdesk 17 BPI 13, BPI 15, BPI 18, BPI 19	Predição do próximo instante de tempo Predição da próxima atividade
Venugopal <i>et al.</i> (2021)	GNN	Representação Matricial	Helpdesk 17 BPI 12 W	Predição do próximo instante de tempo Predição do tipo da próxima atividade
Este trabalho	BiLSTM Auto-atenção Transformer	Tokens	Helpdesk 17 BPI 12 A25,A50,A75	Predição do tempo restante para conclusão

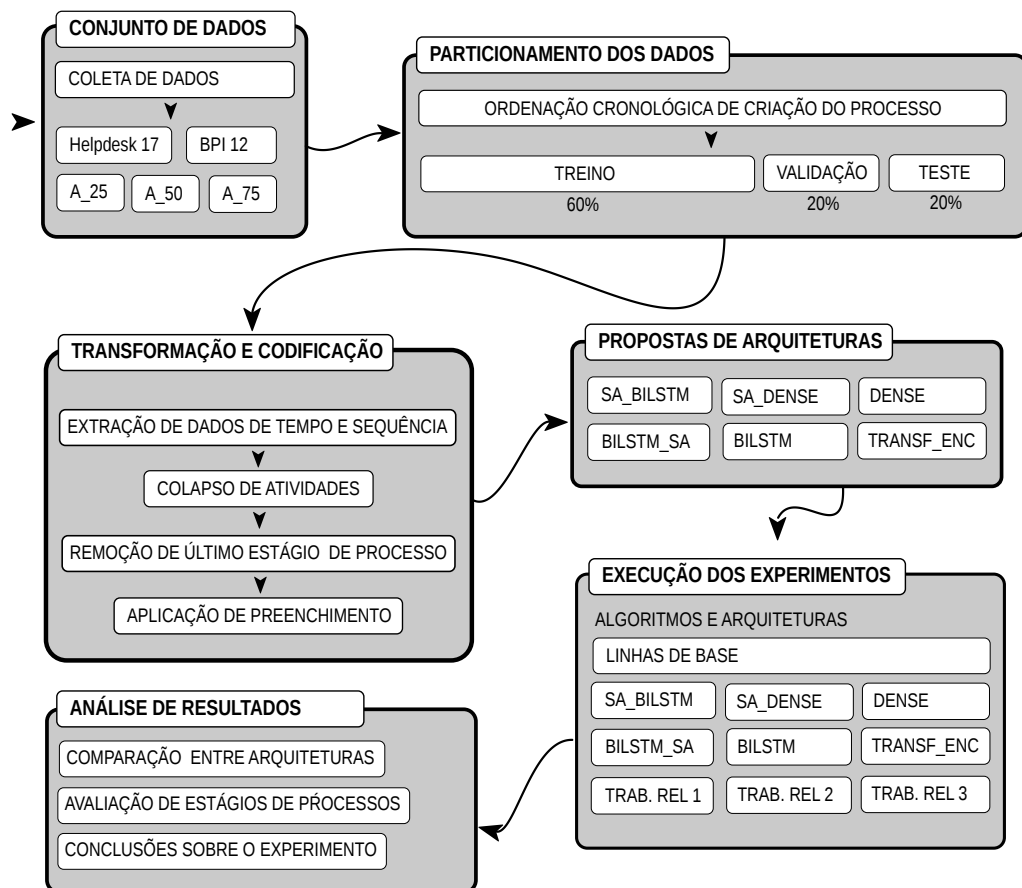
Fonte: Elaborado pelo autor.

4 METODOLOGIA

Este capítulo engloba todas as tomadas de decisão para a execução do experimento, trata também da manipulação de dados, engenharia de características, escolha da abordagem de treinamento do modelo de aprendizado de máquina, aprendizado profundo, teste, validação e avaliação do modelo de aprendizado profundo quando se busca a predição de tempo restante de conclusão de um processo de negócio.

A Figura 20 ilustra a organização da execução das etapas do procedimento metodológico deste trabalho.

Figura 20 – Esquema da sequência de etapas do processo metodológico.



Fonte: Elaborado pelo autor.

4.1 Conjuntos de Dados

Este trabalho utiliza dois conjuntos de dados amplamente conhecidos na literatura, o *BPI 12* e o *Helpdesk 17* e três conjuntos de dados não abertos ao público (cedidos pela SEFAZ-CE) descritos a seguir.

BPI 12

O conjunto de dados *Business Process Intelligence 2012* (BPI 12) 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, Polato (2017) 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.

SEFAZ-CE

Os dados utilizados nesta etapa do trabalho provém dos *logs* de eventos da SEFAZ-CE¹. Esse conjunto de dados é composto por processos de negócio executados entre 1º de abril de 2015 a 9 de dezembro de 2021. Ao todo, são 1.201.347 processos distintos e mais de 7 milhões de (7.719.966) de eventos de processo, distribuídos em 356 assuntos (departamentos) distintos. Destes variados conjuntos de dados, três assuntos (A) foram selecionados para servirem de material de experimentação conforme os seguintes requisitos:

- Os processos do assunto *A_25* que representa o que mais se aproxima do quartil de 25% do desvio padrão (≈ 34 dias) de todo o conjunto de dados da SEFAZ-CE (387.38 instâncias). A natureza desse conjunto de dados se relaciona com processos que envolvem solicitações cadastrais entre 1º de abril de 2020 e 9 de dezembro de 2021;
- Os processos do assunto *A_50*, referente a impostos, que datam entre 8 de outubro de 2020 a 9 de dezembro de 2021, representa o que mais se aproxima do quartil de 50% do desvio padrão (≈ 55 dias) de todo o conjunto de dados da SEFAZ-CE (18.416 instâncias);

¹ <<https://www.sefaz.ce.gov.br/>>

- Os processos do assunto *A_75* que representa o que mais se aproxima do quartil de 75% do desvio padrão (≈ 100 dias) de todo o conjunto de dados da SEFAZ-CE (25.685 instâncias). Esse processo trata de regime de tributações e seus registros foram capturados entre 8 de abril de 2015 a 28 de setembro de 2021.

Atributos dos conjuntos de dados

Como descrita as principais características de um registro de eventos na Tabela 1, o processo de escolha de características dos conjuntos de dados apresentados nesta seção também seguirão a mesma ideia. Uma descrição resumida dos atributos do conjunto de dados *Helpdesk 17* (também aplicável aos demais conjuntos de dados) escolhidos para este experimento é apresentada na Tabela 3.

Tabela 3 – Descrição das características do conjunto de dados do *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.

Fonte: Elaborado pelo autor.

4.2 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. Os demais conjuntos de dados: *A_25*, *A_50* e *A_75* possuem respectivamente 6.140, 2.812 e 1.400 instâncias de processos pertencentes a 38.738, 18.416 e 25.685 eventos. A composição dos conjuntos de treino, validação e testes segue a proporção

60%, 20% e 20% respectivamente, proporções de particionamento bastante utilizadas como padrões em aprendizado de máquina (LAKSHMANAN *et al.*, 2020).

Detalhes a respeito do particionamento e a distribuição das instâncias de processos e seus eventos em cada conjunto são organizados na Tabela 4.

Tabela 4 – Números de instâncias de processo e de eventos relacionados ao particionamento dos conjuntos de dados *BPI 12*, *Helpdesk 17*, *A_25*, *A_50* e *A_75*

Conjunto de Dados		Partição			Total
		Treino	Validação	Teste	
<i>BPI 12</i>	Instâncias de Processos	7.852	2.618	2.617	13.087
	Eventos	85.963	26.865	30.509	143.337
<i>Helpdesk 17</i>	Instâncias de Processos	2.745	896	916	4.557
	Eventos	10.254	3.417	3.004	16.675
<i>A 25</i>	Instâncias de Processos	3684	1228	1228	6.140
	Eventos	23.180	7.845	7.713	38.738
<i>A 50</i>	Instâncias de Processos	1687	562	563	2.812
	Eventos	10.784	3.744	3.888	18.416
<i>A 75</i>	Instâncias de Processos	840	280	280	1.400
	Eventos	14495	5688	5502	25.685

Fonte: Elaborado pelo autor.

A Tabela 5 organiza mais detalhes sobre a quantidade média de eventos por processo. O conjunto de dados *A_75* possui, em média, processos bem mais longos que os demais, seguido pelo *BPI 12*. Os conjuntos *A_25* e *A_50* possuem uma quantidade média similar, configurando um tamanho médio entre os apresentados até aqui. Por fim, o *Helpdesk 17* é composto em média por processos mais curtos.

Tabela 5 – Quantidade média de eventos por processo.

Conjunto de dados	Média de eventos por instância de processo
BPI 12	10,9526
Helpdesk 17	3,6592
A_25	6,3091
A_50	6,5490
A_75	18,3464

Fonte: Elaborado pelo autor.

Em resumo, o critério de particionamento engloba também as características: todos os processos do conjunto de treino ocorreram antes de todos os processos do conjunto de validação e esse último, antes de todos os processos de negócios que compõe o conjunto de teste. A Tabela 6 organiza as informações a respeito da quantidade de atividades distintas encontradas

em cada um dos subconjuntos.

Tabela 6 – Quantidade de atividades distintas por subconjunto utilizado no experimento.

Conjunto de dados	Quantidade de atividades distintas por subconjunto de dados		
	Treino	Validação	Teste
<i>BPI 12</i>	24	24	23
<i>Helpdesk 17</i>	9	12	11
<i>A_25</i>	588	314	319
<i>A_50</i>	297	242	244
<i>A_75</i>	315	166	160

Fonte: Elaborado pelo autor.

Dada essa organização de informações (Tabela 6), é clara a alta diversidade das atividades dos processos que compõem os conjuntos *A_25*, *A_50* e *A_75*. Essa grande variação está ligada diretamente à flexibilidade que a modelagem do processo pode permitir. Por outro lado, processos dessa natureza geralmente são bem mais desafiadores.

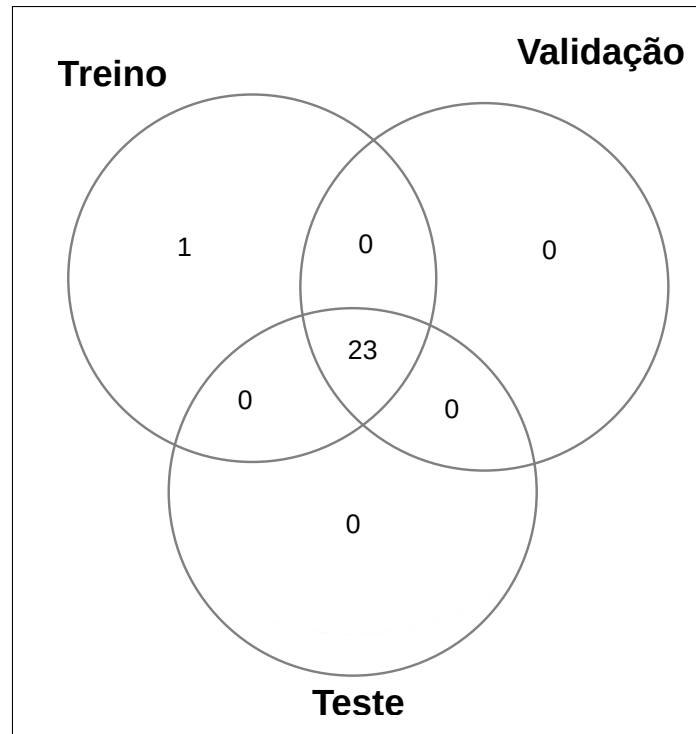
Conjuntos podem ser representados graficamente usando diagramas de *Venn*, onde o conjunto universal U , contém todos os objetos em consideração, é representado por um retângulo. Dentro desse retângulo, círculos ou outras figuras geométricas são usadas para representar conjuntos. Os diagramas de *Venn* são frequentemente usados para indicar as relações entre os conjuntos, nesse caso, a relação entre as atividades distintas em cada conjunto (treino, validação e teste) (ROSEN, 2007; YESHPANOV *et al.*, 2021).

Os diagramas que seguem, mostram como se apresentam as atividades distintas entre cada subconjunto (treino, validação e teste) de dados.

A Figura 21 mostra a clara boa distribuição das atividades pelos conjuntos de treino, validação e teste. A intersecção entre as atividades dos subconjuntos de treino, validação e teste compõe praticamente todo o conjunto U . Conforme a Tabela 6, há apenas um tipo de atividade que não existe no conjunto de teste (subconjunto de atividades recentes). Ao longo da existência da instância de processos que esse conjunto de dados registra, praticamente, o modelo do processo não se altera, compondo assim, *logs* de eventos que facilitam a aprendizagem do modelo proposto pelos autores.

O diagrama dos dados do *Helpdesk 17*, apresentados na Figura 22 também configura um bom conjunto de dados. Há apenas um tipo de atividade que existe exclusivamente na partição de teste. Apesar de não ser o ideal, do ponto de vista de um bom particionamento, não dificulta de modo considerável o processo de treinamento, teste ou validação dos modelos e

Figura 21 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados do *BPI 12*.



Fonte: Elaborado pelo autor.

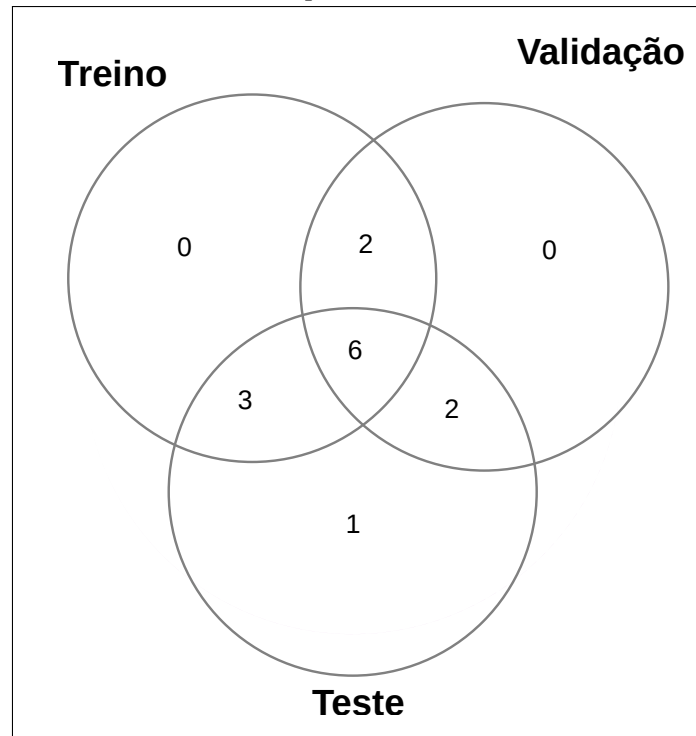
arquiteturas apresentadas neste trabalho.

O diagrama dos dados do *A_25*, apresentados na Figura 23 ilustra a existência de várias atividades distintas no conjunto de treino, evidenciando também que há poucas atividades exclusivas nos conjuntos de validação e teste. A problemática nessa análise é de que o modelo aprenderá mais informações que não vão ser tão cruciais nas etapas de validação e teste, dada a grande mudança do processo no decorrer de sua existência. Muitas atividades distintas num subconjunto de treino, que comporta os processos mais antigos dos dados do conjunto *A_25*.

Uma análise sob o Diagrama de *Venn* dos dados do conjunto *A_50*, (Figura 24) gira em torno dos fatos de que a quantidade de atividades exclusivas entre os conjuntos de treino e validação, validação e teste e teste e treino são praticamente as mesmas, e vão, teoricamente, dificultar a aprendizagem do modelo nas três etapas da execução do experimento (treinamento, validação e teste). Além disso, existe uma quantidade razoável de atividades exclusivas para o conjunto de treino (eventos anteriores aos contidos em validação e teste), isso significa que o processo mudou durante o passar do tempo.

O conjunto *A_75* configura uma partição para treino, validação e teste que não é

Figura 22 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados *Helpdesk 17*.

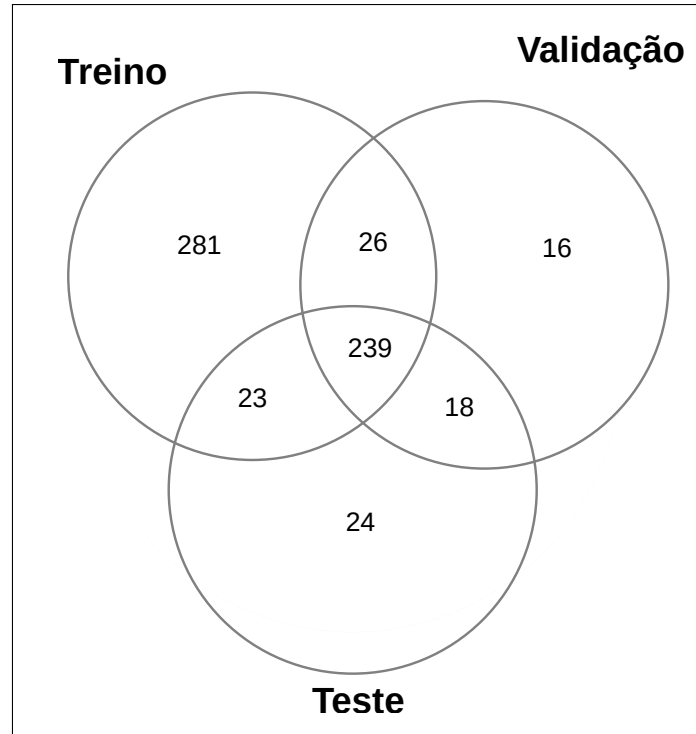


Fonte: Elaborado pelo autor.

promissora. Na Figura 25 mostra intersecções de pares de conjuntos com muitas atividades distintas e a intersecção entre os conjuntos de treino, validação e teste é muito pequena em relação à quantidade total de atividades distintas, configurando uma má distribuição e uma alta mudança de atividades do processo na maior partição (treino), o que era esperado, dada a natureza dos processos selecionados e descritos neste capítulo. Adicionalmente, os conjuntos de validação e teste (de modo respectivo, conjuntos dos processos pouco menos e mais recentes) também sofrem dessa flexibilidade conforme os registros das instâncias desse processo que compõe o conjunto *A_75*. É um conjunto de dados que aparenta ser desafiador para um modelo de aprendizado profundo, principalmente os baseados em sequências e reconhecimento de padrões.

A Tabela 7 organiza informações sobre as intersecções das atividades distintas entre os conjuntos de treino, validação e teste. Quanto menor a diferença entre as intersecções apresentadas, melhor a distribuição das atividades e mais eficiente será o processo de treinamento do modelo.

Figura 23 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados A_25.



Fonte: Elaborado pelo autor.

Tabela 7 – Intersecção das atividades distintas entre os conjuntos de treino, validação e teste.

Conjunto de dados	Total (U)	$TR \cap TS$	$TR \cap VL$	$TR \cap TS \cap VL$
BPI 12	24	24	24	23
Helpdesk 17	14	9	8	6
A 25	627	262	265	239
A 50	337	216	220	203
A 75	444	75	80	56

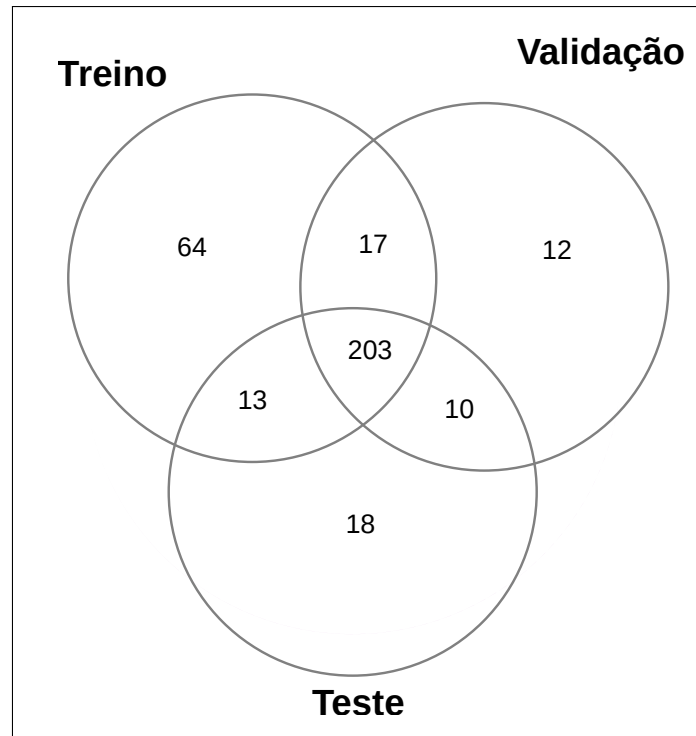
Fonte: o autor.

Nota: U: Conjunto universo; TR: Conjunto de treino; VL: Conjunto de validação; TS: Conjunto de teste

4.3 Transformação e codificação

Nesta etapa do processo metodológico 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 onde o evento se encontra. Uma etapa assume um valor entre $1..n$, onde n é o máximo de etapas que o processo possui (coluna *Etapa* da Tabela 9). A definição do tempo em dias se deu pelo uso de três atributos derivados do *Instante da criação* do evento: a duração (coluna *Duração* da Tabela 9) do evento, o tempo de execução do processo

Figura 24 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados A_50.



Fonte: Elaborado pelo autor.

até o evento atual (coluna *Tempo passado* da Tabela 9) e o tempo restante para a conclusão do processo (coluna *Tempo restante para conclusão* da Tabela 9).

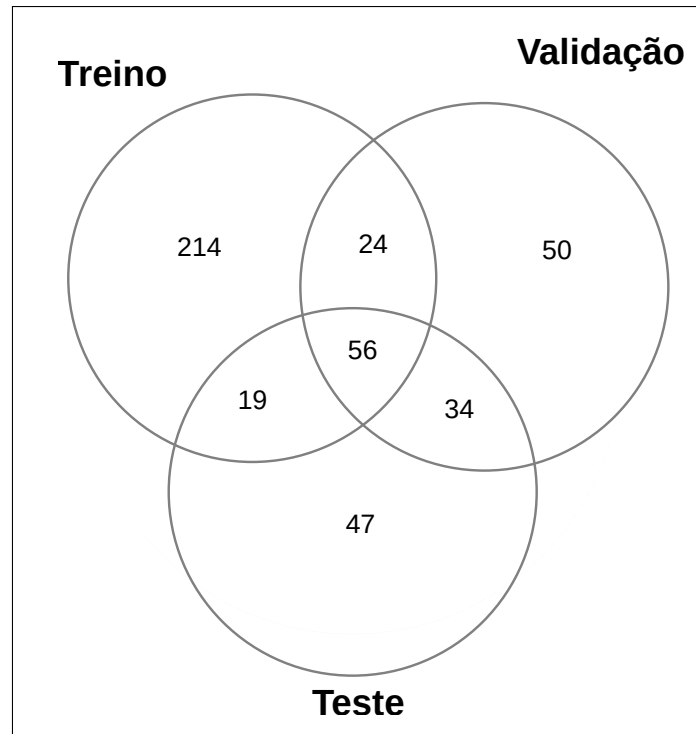
Cada processo é representado por várias amostras (eventos) nos conjuntos de dados. Não só a variedade, mas a quantidade de eventos em um processo é um fator importante que reflete a dimensão da entrada para cada uma das arquiteturas apresentadas neste trabalho. A Tabela 8 contém informações sobre o tamanho do processo com a maior quantidade de etapas encontradas em cada um dos conjuntos. Os conjuntos A_75 e BPI 12 possuem no mínimo um processo com muitas etapas. Nos conjuntos A_25 e A_50, seus maiores processos têm tamanho próximo. O Conjunto Helpdesk 17 é, em geral, formado por processos de poucas etapas.

Tabela 8 – Tamanho do maior processo encontrado nos conjuntos de dados.

	Tamanho do maior processo
BPI 12	55
Helpdesk 17	10
A_25	24
A_50	22
A_75	69

Fonte: o autor.

Figura 25 – Diagrama de informação sobre a quantidade de atividades distintas nos conjuntos de treino, validação e teste do conjunto de dados A_75.



Fonte: Elaborado pelo autor.

Nas diferentes partições, os processos são completos. Todas as suas amostras estão contidas em todos os três subconjuntos de dados (treino, validação e teste). Não existe parte de um mesmo processo em dois ou mais subconjuntos. As linhas que possuem etapas 5 e 4 (Tabela 9) onde os valores da coluna **Tempo restante para conclusão** são iguais a 0, foram removidas do conjunto de treino, validação e teste. Uma vez que o processo encerra, não se vê a necessidade de prever nenhuma outra etapa após essa.

Tabela 9 – Amostras do conjunto de dados *Helpdesk 17*.

ID	Atividade	Etapas	Duração	Tempo passado	Tempo restante para conclusão
1	1	1	0.000000	0.000000	51.146700
1	1	2	14.042100	22.001400	37.104600
1	12	3	0.000100	37.001600	37.104500
1	9	4	0.000100	37.001700	37.104400
1	12	5	0.102700	37.104400	37.001700
1	9	6	0.000100	37.104500	37.001600
1	2	7	15.000200	37.104600	22.001400
1	2	8	22.001400	51.146700	-0.000000
2	1	1	0.000000	0.000000	44.007300
2	12	2	0.000300	44.005600	44.007000
2	9	3	0.001400	44.007000	44.005600
2	2	4	44.005600	44.007300	0.000000

Fonte: Elaborado pelo autor.

Uma maneira de resumir a representação 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 acumularem 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 si mesma *Atividade* está descartada.

A Tabela 9 apresenta amostras do conjunto de dados *Helpdesk 17* (os conjuntos *BPI 12*, *A_25*, *A_50* e *A_75*, possuem a mesma estrutura, porém valores diferentes). A repetição em sequência de atividades nas etapas (*Etapa*) 7 e 8 da instância de processo de *ID 1* (Tabela 9), 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, representou-se essa sequência de atividades (linhas 1 a 3 da Tabela 9) como uma sequência de prefixos na forma matricial (Tabela 10) onde a quantidade de etapas do maior processo é 10. O conjunto *BPI 12* possui a mesma estrutura com *padding* igual à 55. Os demais conjuntos de dados *A_25*, *A_50* e *A_75* possuem, respectivamente, tamanhos 24, 22 e 69 (Tabela 8)

Tabela 10 – Prefixos na amostra do conjunto de dados *Helpdesk 17*.

Prefixo e Padding						Atributos temporais			Rótulo	
1	2	3	4	5	6	Duração	Tempo Passado	Step	Tempo Restante Para Conclusão	
1	0	0	0	0	0	14,0421	22,0014	1	88,2513	
1	12	0	0	0	0	0,0001	37,0016	2	37,1045	
1	12	9	0	0	0	0,0001	37,0017	3	37,1044	
1	12	9	12	0	0	0,1027	37,1044	4	37,0017	
1	12	9	12	9	0	0,0001	37,1045	5	37,0016	
1	12	9	12	9	2	37,0016	88,2513	6	22,0014	

Fonte: Elaborado pelo autor.

No exemplo anterior, os valores iguais a 0 são usados como preenchimentos (*padding*), que possibilitam uniformizar o número de colunas na representação das atividades das instâncias de processos. Essa matriz resultante substitui a coluna *Atividade* e representa o histórico das etapas passadas (Tabela 9).

4.4 Arquiteturas propostas

As arquiteturas propostas neste trabalho são estruturadas utilizando *Embeddings* para codificação vetorial das atividades.

As propostas das arquiteturas A, B, C, D, E, F, G, H e I organizadas na Figura 26 visam mesclar o suporte aos dados temporais e sequenciais que o processo de negócio possui, dado pela *Bidirectional Long Short Term Memory* (BiLSTM), atrelado à identificação da importância de certas etapas do processo, feita pela camada de atenção, semelhante ao trabalho de Wang *et al.* (2019) que também utiliza conjuntos de dados já conhecidos, porém com ênfase em atividades e tamanhos de prefixos específicos.

O modelo mais complexo (Figura 26 A)) é um modelo sequencial onde os dados de definição da atividade 9 passam por uma camada de *Embedding* e em seguida são processados por uma camada de autoatenção (*Self-Attention 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 Encoding*, 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 onde 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 9). 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, as quais 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 de avanço e de retrocesso, 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.

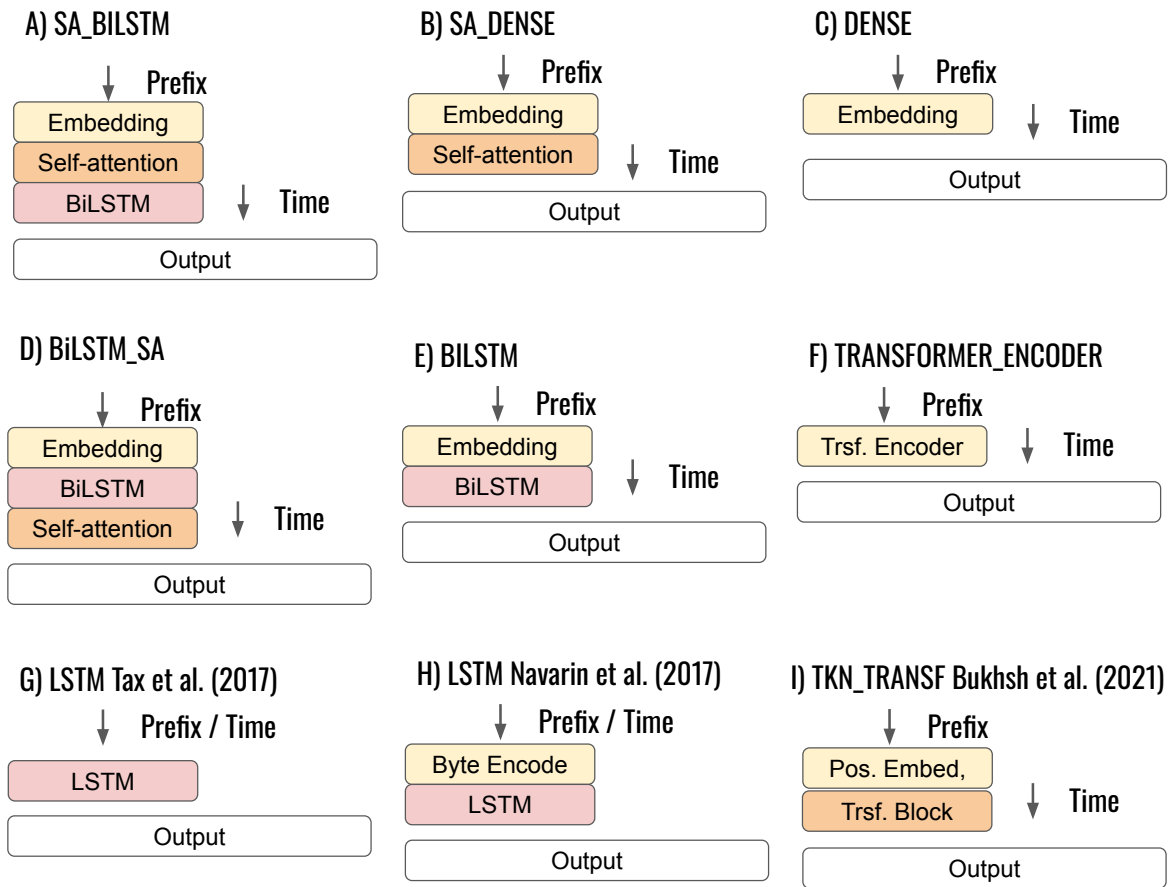
- C) DENSE** Seguindo a ideia de reduzir a complexidade do modelo, esta arquitetura é uma das mais simples apresentadas nessa seção. Além da ausência da BiLSTM, a camada de atenção também não é usada aqui. Este modelo se assemelha à modelos clássicos de aprendizado de máquina.
- D) BiLSTM_SA** Esta versão 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 um pouco mais simples de *RNN*, como apresentadas em (TAX *et al.*, 2017) e (NAVARIN *et al.*, 2017), porém, com uma arquitetura que funciona em ambas as direções e com a concatenação dos dados temporais com os relativos às atividades.
- F) TRANSFORMER_ENCODER** A segunda arquitetura mais simples deste experimento, em relação a camadas, se configura como a arquitetura *C* e utiliza uma camada de transformação e codificação se guida da saída ao invés de uma camada de *Embedding*.
- G) LSTM Tax *et al.* (2017)** Uma das arquiteturas utilizadas em trabalhos externos (Seção 3.1), a LSTM utilizada por (TAX *et al.*, 2017), é um modelo simples com poucas camadas, porém resolvida simultaneamente de problemas de regressão e classificação.
- H) LSTM Navarin *et al.* (2017)** Também utilizando uma LSTM com poucas camadas, (NAVARIN *et al.*, 2017) (Seção 3.2) soluciona problemas de regressão como o de prever o tempo restante para conclusão de uma instância de processo.
- I) TKN_TRANSFORMER Bukhsh *et al.* (2021)** Arquitetura descrita na Seção 3.3, desenvolvido por (BUKSHSH *et al.*, 2021), utiliza de tokenização, prefixos e *transformers* para solucionar o mesmo problema apresentado neste trabalho.

4.5 Execução dos experimentos

Nesta etapa final do processo metodológico, é proposta a execução de cada um dos modelos apresentados neste trabalho com todos os conjuntos de dados conforme a Seção 4.1. Além disso, há a experimentação² com as arquiteturas propostas de alguns dos trabalhos relacionados (Capítulo 3).

² <<https://github.com/RonildoSilva/Reproducao-Experimentos-PPM/>>

Figura 26 – Arquiteturas de redes neurais usadas: (A) SA_BILSTM; (B) SA_DENSE; (C) DENSE; (D) BILSTM_SA; (E) BILSTM; (F) TRANSFORMER_ENCODER; (G) (TAX); (H) LSTM (NAVARIN); (I) TKN_TRANSF (BUKHSH)



Fonte: Elaborado pelo autor.

4.5.1 Metodologia de avaliação

Todos os modelos descritos na Figura 26 recebem como entrada, as sequências de prefixos e de definições das características temporais do conjunto de treino.

A Tabela 11 organiza os valores de desvio padrão de cada um dos subconjuntos definidos para o experimento e análise de significância de cada partição dos dados de eventos de processos.

A Tabela 12 organiza a média do tempo restante para conclusão (rótulo) para cada uma das partições dos dados.

Em alguns casos isolados nos conjuntos de dados *BPI 12* e *Helpdesk 17*, principalmente nos conjuntos *A_25*, *A_50* e *A_75*, o valor do desvio padrão é maior que a média de cada agrupamento dos rótulos manipulados e isso evidencia bastante o quão dispersos são os dados

Tabela 11 – Desvio padrão (em dias) das *labels* dos conjunto de treino, validação e teste do *BPI 12*, *Helpdesk 17*, *A_25*, *A_50* e *A_75*.

Desvio Padrão do Tempo Restante Para Conclusão					
Conjunto	BPI 12	Helpdesk 17	A_25	A_50	A_75
Treino	13,0282	11,1193	42,3115	70,2296	109,5282
Validação	11,9414	9,9398	23,2113	25,0358	89,6194
Teste	8,5743	12,5011	24,9786	25,3953	89,6764

Fonte: Elaborado pelo autor.

Tabela 12 – Média (em dias) das *labels* dos conjunto de treino, validação e teste do *BPI 12*, *Helpdesk 17*, *A_25*, *A_50* e *A_75*.

Média do Tempo Restante Para Conclusão					
Conjunto	BPI 12	Helpdesk 17	A_25	A_50	A_75
Treino	11,0929	36,1666	23,8269	48,6381	57,1359
Validação	11,2576	37,7909	16,2409	19,7326	71,3337
Teste	8,4947	29,1493	17,0729	20,3092	71,2313

Fonte: Elaborado pelo autor.

desses conjuntos.

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 presentes numa determinada etapa.

5 RESULTADOS

Este capítulo trata da discussão e dos resultados dos experimentos realizados com os algoritmos e arquiteturas de modelos de aprendizado preditivos propostos. Como soluções de linha de base foram utilizadas as seguintes estratégias:

Dummy Regressor¹ Algoritmo que retorna o valor médio dos valores esperados no conjunto de treinamento, sem observar os valores de entrada. A estratégia adotada é utilizar a média do rótulo.

Regressão Linear² Algoritmo que ajusta um modelo linear com coeficientes para minimizar a soma residual entre os alvos observados no conjunto de dados e os alvos previstos por aproximação linear. Os parâmetros padrões da biblioteca foram preservados, exceto o de adequação da interceptação (`fit_intercept`), que nesse experimento é igual a falso.

Regressão com Floresta Aleatória³ Uma floresta aleatória é um meta-estimador que ajusta uma série de árvores de decisão organizada em várias subamostras do conjunto de dados e usa a média para auxiliar na predição. Vinte estimadores foram usados e tamanho da profundidade máxima foi de 19.

Regressão com Aumento de Gradiente Leve e Extremo⁴ (*Light GBM* e *XGBoost*) São algoritmos de que treinam múltiplas árvores de decisão e depois combina os resultados. Eles permitem que o modelo aprenda mais rapidamente do que outros algoritmos, mas também oferece uma vantagem em situações com muitos recursos a serem considerados⁵. Cem estimadores foram usados com profundidade máxima igual a 7. Para o algoritmo que usa o aumento leve, os parâmetros padrões foram preservados.

As arquiteturas propostas neste trabalho também foram comparadas com as soluções de aprendizado profundo do estado da arte para predição de tempo restante para conclusão de processos de negócios, 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 13 apresenta os resultados obtidos com a métrica MAE para os conjuntos de dados *BPI 12*, *Helpdesk 17*, *A_25*, *A_50* e *A_75*. Nessa experimentação, são consideradas apenas as instâncias de processos que chegaram a sua conclusão e nenhuma instância de processo existe em mais de um conjunto (treino, validação ou teste). As métricas apresentadas para os trabalhos relacionados são obtidas a partir da execução de novos experimentos usando suas arquiteturas e particionamento e filtragem apresentadas neste trabalho (Capítulo 4). Conseqüentemente, os particionamentos propostos

⁵ <<https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article>>

neste trabalho, diferem dos usados nos trabalhos originais, porém, para poder existir uma análise geral, buscou-se, assim, a mesma filtragem e particionamento único para uma comparação mais justa entre todas as abordagens.

Tabela 13 – Desempenho dos modelos usando a métrica MAE.

Modelo	<i>BPI 12</i>	<i>Helpdesk 17</i>	<i>A_25</i>	<i>A_50</i>	<i>A_75</i>
	Valor MAE (95% I.C.)*	Valor MAE (95% I.C.)*	Valor MAE (95% I.C.)*	Valor MAE (95% I.C.)*	Valor MAE (95% I.C.)*
Dummy Regressor	7,64 (7,58 - 7,69)	12,06 (11,77 - 12,32)	19,06 (18,65 - 19,47)	29,49 (29,00 - 29,98)	61,81 (60,02 - 63,59)
Regressão Linear	6,03 (5,98 - 6,08)	12,02 (11,48 - 12,67)	11,89 (11,57 - 12,21)	20,49 (19,92 - 21,05)	50,88 (49,17 - 52,59)
Random Forest	4,89 (4,82 - 4,96)	5,52 (5,34 - 5,69)	10,37 (10,02 - 10,72)	17,30 (16,57 - 18,03)	44,89 (43,13 - 46,65)
XGBoost	5,02 (4,95 - 5,09)	5,73 (5,55 - 5,90)	10,59 (10,23 - 10,95)	18,12 (17,38 - 18,87)	44,85 (43,09 - 46,62)
Light GBM	4,84 (4,78 - 4,90)	5,41 (5,26 - 5,57)	11,32 (10,96 - 11,68)	17,97 (17,33 - 18,61)	48,88 (47,15 - 50,61)
LSTM Tax <i>et al.</i> (2017)	N.A.**	6,17 (5,50 - 6,33)	N.A.**	N.A.**	N.A.**
LSTM Navarin <i>et al.</i> (2017)	7,25 (7,17 - 7,34)	8,46 (8,30 - 8,62)	N.A.***	N.A.***	N.A.***
TKN_TRANSF Bukhsh <i>et al.</i> (2021)	4,91 (4,88 - 4,94)	5,69 (5,53 - 5,86)	9,45 (9,14 - 9,75)	10,92 (10,50 - 11,35)	47,30 (45,56 - 49,05)
SA_BiLSTM	<i>3,80(3,74 - 3,87)</i>	5,70 (5,50 - 5,89)	<i>8,89 (8,53 - 9,25)</i>	12,83 (12,23 - 13,42)	52,08 (50,11 - 54,04)
DENSE	4,25 (4,18 - 4,31)	7,58 (7,42 - 7,75)	10,06 (9,71 - 10,42)	14,53 (13,94 - 15,13)	49,88 (47,92 - 51,85)
BiLSTM_SA	3,72 (3,65 - 3,79)	5,89 (5,70 - 6,07)	<i>8,91 (8,55 - 9,26)</i>	12,67 (12,07 - 13,26)	51,58 (49,62 - 53,53)
BiLSTM	<i>3,72 (3,65 - 3,79)</i>	<i>5,58 (5,39 - 5,77)</i>	8,48 (8,12 - 8,84)	12,85 (12,23 - 13,47)	46,15 (44,32 - 47,97)
SA_DENSE	3,71 (3,65 - 3,78)	5,22 (5,04 - 5,40)	8,98 (8,60 - 9,35)	13,76 (13,11 - 14,40)	47,45 (45,51 - 49,38)
TRANSF_ENCO	4,24 (4,18 - 4,31)	6,64 (6,48 - 6,80)	9,91 (9,56 - 10,27)	14,49 (13,92 - 15,07)	51,17 (49,18 - 53,15)

Fonte: Elaborado pelo autor.

Nota: Células com valores em preto e negrito, representa o melhor resultado obtido. Células com valores em preto e itálico representam os empates dentro do intervalo de confiança entre as arquiteturas e algoritmos apresentados.

Nota: * Intervalo de confiança abaixo e sem interseção.

Nota: ** Não foi possível realizar o experimento completo. O modelo gera dimensões diferentes para os conjuntos de treino e teste para a partição proposta neste trabalho.

Nota: *** Não foi possível realizar o experimento completo. O modelo gera dimensões de acordo com a quantidade de atividades distintas e com o maior tamanho de um processo encontrado no conjunto. Sem a alteração da arquitetura proposta pelo autor relacionado, a experimentação ocasiona erro de memória.

5.1 Discussão de resultados

A listagem a seguir discorre a respeito das características, resultados de cada uma das arquiteturas apresentadas neste trabalho. As linhas de base serão discutidas pontualmente, conforme os impactos positivos dos resultados obtidos.

5.1.1 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 processo, o qual é o caso do *BPI 12*, produzindo um resultado mediano entre as arquiteturas apresentadas neste trabalho e difere em 1,44 dias do melhor *baseline* (*Light GBM*) apresentado.

O modelo em questão, também produz resultados semelhantes quando o conjunto de dados é o *Helpdesk 17* e configura também uma arquitetura produtora de um resultado mediano (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.

O *SA_BiLSTM*, ao usar um conjunto de dados fora do escopo da literatura (*A_25*), produz um bom resultado. Com 8,89 dias de MAE, a arquitetura proposta aqui, por ter camada de RNN, dá suporte a sequências maiores, como apresentado na Tabela 5.

O conjunto *A_50*, também possui características semelhantes ao *A_25*: possuem a média de eventos por processos bem semelhantes e configura também o segundo melhor resultado com essa arquitetura. A *BiLSTM* segue sendo uma boa arquitetura para processos com poucas etapas em média.

Por fim, a arquitetura *SA_BiLSTM* utilizando o conjunto *A_75* não produziu resultados muito promissores. O conjunto apresentado, além de ter uma quantidade média alta de eventos por processo, o desvio padrão que compõe os conjuntos de treino, validação e teste são também muito altos e (Tabela 11) dificultam bastante a predição do tempo restante para a conclusão de um processo de negócio.

5.1.2 *DENSE*

Apesar dessa arquitetura ser a mais simples dentre as propostas neste trabalho e a próxima dos modelos de *baseline* do que diz respeito à robustez, uma arquitetura baseada apenas em camadas densas, produzem os resultados com maior erro absoluto médio que os demais modelos de aprendizado profundo apresentados neste trabalho. De modo geral, a ausência de camadas de atenção e/ou de RNN, comprometem bastante o alcance de um resultado promissor, visto que o problema envolve dados em sequência e que etapas específicas de um processo são mais importantes que outras, definindo a natureza dele, o que não é priorizado nesse experimento.

5.1.3 *BiLSTM_SA*

Esta arquitetura é uma variação clara da arquitetura *SA_BiLSTM (A)*, que inicialmente produzirá uma matriz de relação entre as atividades (produto de vetores), reduz também a faixa de representação dos *embeddings* para valores entre 0 e 1, reduz o tamanho da representação das atividades, e nesse caso, também produz resultados promissores. Ao ter sequências maiores no *log* de eventos e maiores distribuições das atividades, como no *BPI 12*, o erro absoluto médio

é um pouco maior que o melhor resultado apresentado na Tabela 13 e melhor em relação aos *baselines* e arquiteturas dos trabalhos relacionados apresentados.

Em contrapartida, não é um bom modelo quando as sequências são menores ou com uma distribuição menor de atividades, como no conjunto de dados *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 autoatenção com *BiLSTM* e não configura o conjunto dos melhores resultados.

Para um conjunto de dados incomum na literatura como o *A_25*, a arquitetura *BiLSTM_SA* pode ser encontrada entre os três melhores resultados nesse experimento. Mais uma vez, processos com quantidades maiores de etapas associadas a uma boa definição do processo, auxiliam a LSTM a ser mais assertiva na predição do tempo restante para conclusão do processo.

A *BiLSTM_SA* conseguiu ser a melhor arquitetura para o conjunto *A_50*. Apesar dos conjuntos *A_50* e *A_25* terem características similares, a ordem da camada de RNN na arquitetura, nesse caso, influenciou na obtenção de um resultado tecnicamente tão bom quanto.

O conjunto de dados *A_75* configura semelhante ao *Helpdesk 17* em relação às arquiteturas que produzem os melhores resultados e é uma arquitetura que produz um MAE mediano em relação às arquiteturas propostas nestes trabalho.

5.1.4 *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 consegue produzir resultados próximos aos melhores resultados obtidos em todas as estratégias analisadas neste trabalho e é a única arquitetura sempre presente nos melhores resultados, sempre presente nos intervalos de confiança obtidos em cada experimento, independente do conjunto de dados utilizado.

No conjunto *Helpdesk 17*, a arquitetura *BiLSTM* apresenta resultados tão promissores quanto o *SA_BiLSTM*. Isso evidencia a necessidade e importância da RNN para resolver esse tipo de problemática.

O comportamento ao utilizar o conjunto de dados *BPI 12* se preserva. Produz o segundo melhor resultado e se adéqua bem ao problema proposto.

Num caso mais específico, o conjunto *A_25* e a arquitetura *BiLSTM* produzem o

melhor resultado em relação as demais combinações de experimentos propostas neste trabalho.

No caso do conjunto *A_25*, a *BiLSTM* não é a mais promissora, porém se encontra entre as três melhores soluções. Esse conjunto se mostrou mais adequado quando existe uma camada de atenção atrelada a de RNN.

Essa arquitetura também está na terceira posição entre os melhores modelos. No *A_75*, os resultados são mais distintos em relação aos demais conjuntos, apresentando resultados aceitáveis quando se utiliza algoritmos dados como linhas de base, pelo fato da natureza do processo escolhido.

5.1.5 *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 autoatenção se conecta em toda a sequência processada, um ponto de melhor desempenho em relação as *RNN*.

A ordem onde a camada de atenção acaba por definir bem o resultado, porém, quando o conjunto é o *A_25*, essa arquitetura não se mantém como o principal, mas fazendo parte das três melhores abordagens.

O conjunto de dados *A_50* combinado com a arquitetura proposta nesse experimento não produz um resultado promissor e fica fora das melhores opções de arquitetura para esse ambiente e problemática.

Entretanto, essa arquitetura se adéqua bem ao conjunto de dados *A_75*, sendo o conjunto de dados mais flexível em seus processos: possui muitas etapas que demoram bem mais para serem concluídas e que mudam bastante a natureza de suas atividades.

5.1.6 *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 26 C). Ambos os resultados não são tão promissores e são apenas melhores que os resultados obtidos com a arquitetura *DENSE*, já que não possuem camadas que dão suporte a sequências. Os resultados obtidos utilizando o tanto os *datasets BPI 12* quanto *Helpdesk 17* produzem o pior resultado dentre as arquiteturas apresentadas neste trabalho.

Assim como a arquitetura *DENSE*, a *TRANSFORMER_ENCODER* também fica fora de todos os resultados promissores, porém produz um erro médio absoluto menor que a

arquitetura de apenas camadas densa.

O diferencial nesse caso é que quando o conjunto de dados é o *A_75*, a arquitetura densa (*DENSE*) é ligeiramente mais eficiente que essa em questão.

5.1.7 *LSTM Tax et al. (2017)*

É uma arquitetura que funciona bem com partições que possuam sempre exemplares de processos de tamanho máximo nos conjuntos de treino, validação e teste, pois as dimensões dos conjuntos de treino, validação e teste são dadas pela quantidade distinta de atividades em cada uma delas. Uma vez que exista algum tipo de exclusividade de atividades em alguma partição, a execução do experimento não poderá ser concluída. A modelagem dessa arquitetura não permite a execução completa do experimento quando existem tamanhos máximos diferentes entre os subconjuntos citados, portanto, não se adéqua bem ao experimento proposto neste trabalho.

5.1.8 *LSTM Navarin et al. (2017)*

A arquitetura apresentada tem uma boa execução quando o dados de entrada são processos de tamanhos pequenos ou quando a quantidade de atividades distintas é considerável pequena, como os conjuntos de dados *BPI 12* e *Helpdesk 17*. Muita variação nesses aspectos, geralmente produz uma dimensão bastante grande, que é difícil de computar.

5.1.9 *TKN_TRANSF Bukhsh et al. (2021)*

Essa arquitetura dispensa RNNs e utiliza camadas de *transformers* além da tokenização das atividades organizadas por prefixos e tratadas de diferente formas em relação aos dados temporais (Seção 3.3). Produz um bom resultado quando o alvo é o conjunto *A_50* e não consegue obter êxito nos demais conjuntos de dados. Apesar da arquitetura de *transformer* ser bastante robusta, não se adapta bem a sequências dessa natureza.

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.

5.2.1 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. Para os demais conjuntos de dados, cada melhor resultado individual, veio da combinação de uma arquitetura ou algoritmo distinto. O conjunto *A_25*, teve como melhor combinação, a arquitetura *BiLSTM*. Para o conjunto *A_50*, a arquitetura *TKN_TRANSF* proposta por Bukhsh *et al.* (2021) foi mais promissora, mesmo não exigindo camadas de apoio a sequências. Por fim, o conjunto de dados *A_75*, teve o melhor resultado, utilizando o algoritmo *XGBoost*, que dispensa camadas densas, suporte a sequências e a mecanismos de atenção.

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

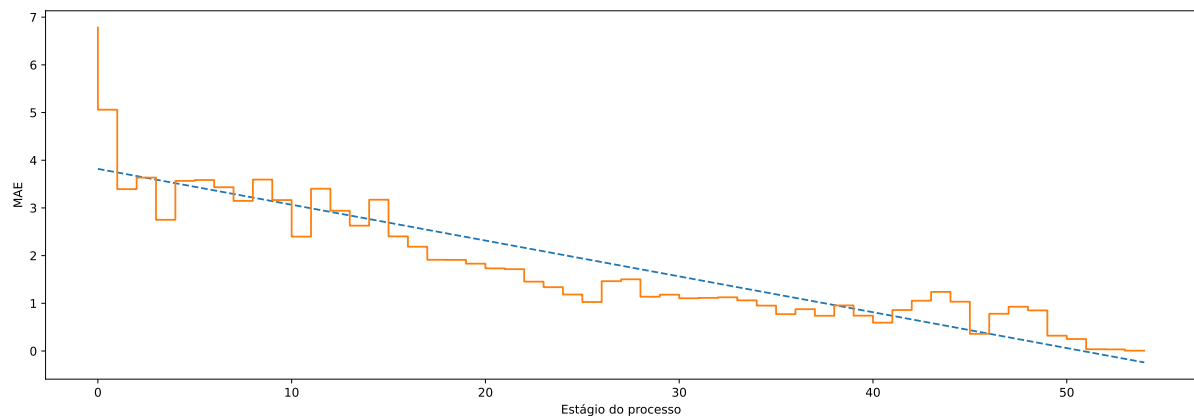
Para responder a esta pergunta, variou-se o estágio dos processos como entrada para os modelos e obteve-se o MAE para cada um dos estágios. Nesse experimento, foi utilizado o melhor modelo avaliado para cada conjunto de dados. As Figuras 27, 28, 29, 30 e 32 apresentam o MAE obtido para cada estágio do processo dos conjuntos de dados *BPI 12* e *Helpdesk 17*, *A_25*, *A_50* e *A_75*, 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 predizer o tempo restante. Este comportamento também pode indicar que os modelos conseguem distinguir melhor as atividades realizadas mais ao final da execução dos processos.

5.2.2.1 Análise sobre estágios do BPI 12

A Figura 27 ilustra um comportamento esperado, onde a tendência do tempo restante para conclusão do processo representa, de maneira macro, um decrescimento a cada etapa conhecida de um processo de negócio. A linha de tendência apresentada torna essa análise mais nítida e confirma a eficiência do modelo e a maturidade do processo de negócio utilizado, que expressa o desempenho esperado do processo de negócios com base nas capacidades reais, a fim de prever o desempenho real dos processos de negócios (LOOY, 2014).

Figura 27 – BPI 12 - MAE por estágio do processo (*SA_DENSE*).

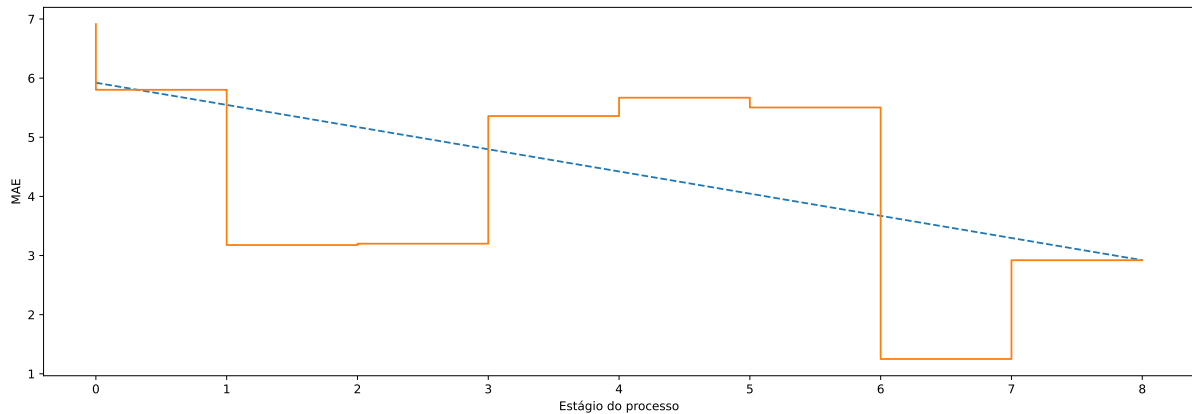


Fonte: Elaborado pelo autor.

5.2.2.2 Análise sobre estágios do Helpdesk 17

A Figura 28, que se refere ao conjunto de dados do *Helpdesk 17* tem uma linha de tendência menos íngreme que a apresentada na Figura 27. Além de apresentar um valor de MAE relativamente grande entre os estágios próximos à metade do tamanho total, ainda tem valores relativamente bons em estágios seguintes. Como existem poucos estágios nesse conjunto de dados, maiores erros, gera uma linha de tendência visualmente menos íngreme.

Figura 28 – Conjunto de dados *Helpdesk 17* - MAE por estágio do processo (*SA_DENSE*).

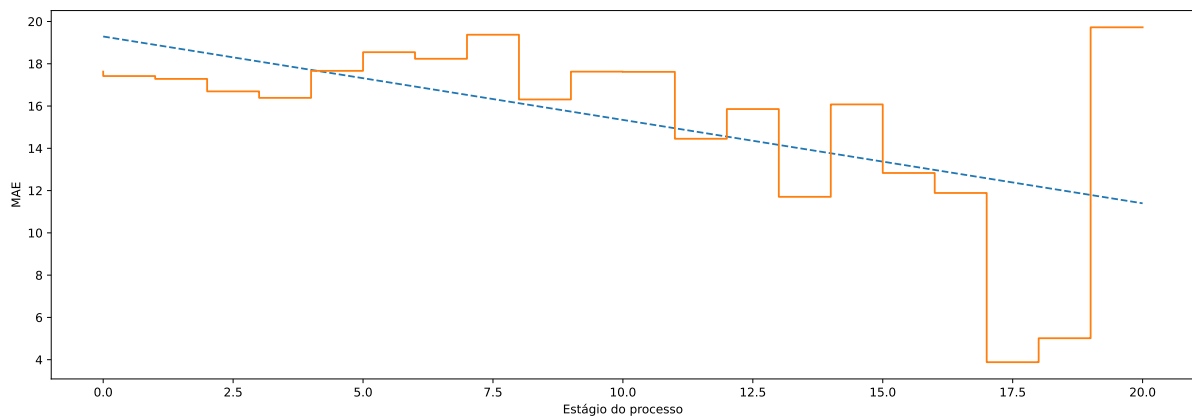


Fonte: Elaborado pelo autor.

5.2.2.3 Análise sobre estágios do A_25

Outro comportamento que se configura bem distinto dos resultados apresentados anteriormente (Figuras 27 e 28) está apresentado na Figura 29, referente ao conjunto de dados *A_25*. Tem-se uma quantidade menor de etapas, a linha de tendência que segue os resultados por estágio decresce tal qual a apresentada na Figura 28, onde os valores de MAE decrescem nos estágios iniciais, têm um aumento envolvendo estágios de tamanho médio, e por fim, configura um pico na etapa final, porém, com o um valor de MAE muito elevado.

Figura 29 – Conjunto de dados *A_25* - MAE por estágio do processo (*BiLSTM*).



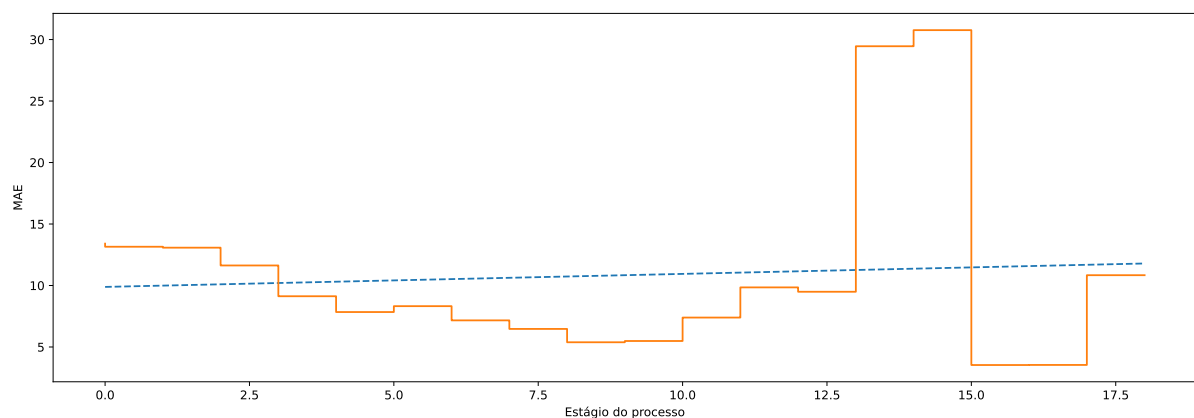
Fonte: Elaborado pelo autor.

5.2.2.4 Análise sobre estágios do A_50

O conjunto de dados *A_50*, por sua vez, conclui a última etapa com um valor de MAE pouco satisfatório. Apesar da arquitetura *TKN_TRANSF Bukhsh et al. (2021)* possuir o melhor resultado da Tabela 13, a linha de tendência apresentada na Figura 30 é não decrescente.

Existe também um grande pico do valor do MAE em etapas finais que vem logo após uma mudança no sentido dos resultados. Há um decréscimo até metade do tamanho máximo da instância de processo, seguido de um aumento abrupto de MAE, antes da conclusão com um resultado ótimo na etapa seguinte e finalizando com valor de MAE próximo do apresentado (em negrito) na Tabela 13. Apesar dos últimos estágios decrescerem muito o valor de MAE, a natureza do desse processo é muito flexível e tem, portanto, padrões de fluxo de atividades pouco previsíveis.

Figura 30 – Conjunto de dados *A_50* - MAE por estágio do processo (*TKN_TRANSF Bukhsh et al. (2021)*).



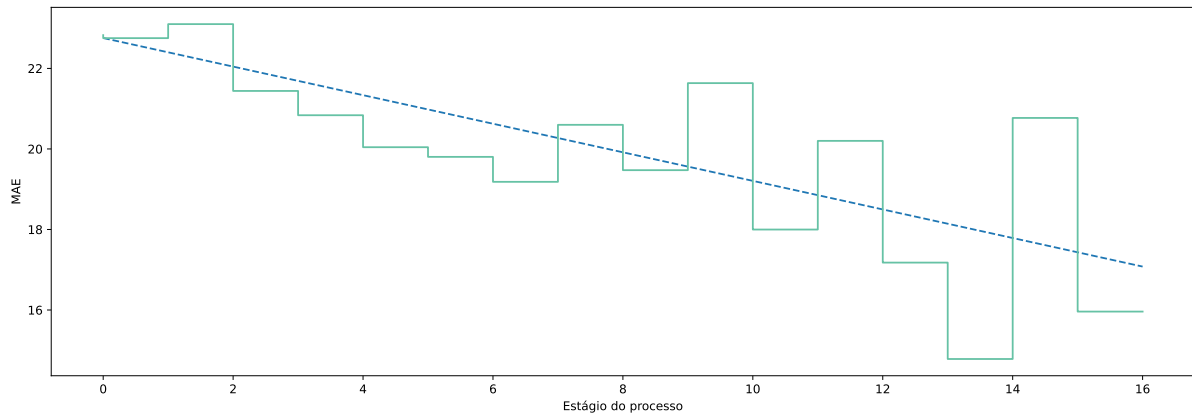
Fonte: Elaborado pelo autor.

Por outro lado, a arquitetura *BiLSTM_SA*, segundo melhor resultado apresentado na Tabela 13 para o conjunto de dados *A_50*, possui um comportamento bastante distinto do apresentado quando se utiliza a arquitetura de Bukhsh *et al.* (2021). A Figura 31 ilustra um melhor comportamento da linha de tendência, ao haver um conjunto de validação bem definido que ajuda o modelo de aprendizado profundo a falhar menos, pois o modelo conhece mais das etapas passadas de um processo e consequentemente, o modelo segue menos enviesado quando se geram as previsões por etapas.

5.2.2.5 Análise sobre estágios do *A_75*

Por fim, na Figura 32, temos a análise sobre os resultados por etapa do conjunto de processos de negócios *A_75*, sendo o mais distinto entre todos os apresentados neste trabalho em relação aos modelos apresentados neste trabalho, pois o melhor resultado vem de um algoritmo de aprendizado de máquina considerado clássico em relação aos demais modelos apresentados. Mesmo o *XGBoost* apresentando um valor de MAE baixo em relação aos demais, tanto o

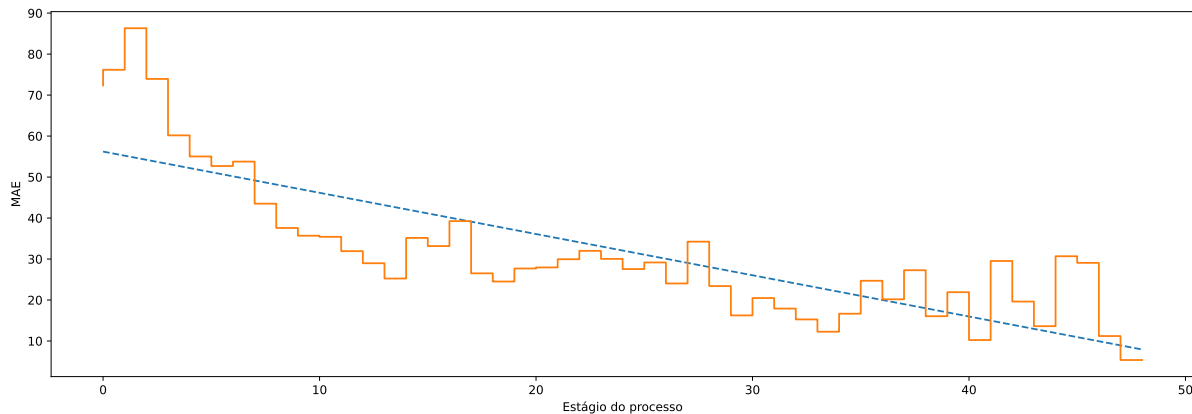
Figura 31 – Conjunto de dados A_50 - MAE por estágio do processo (*BiLSTM_SA*).



Fonte: Elaborado pelo autor.

algoritmo que usa *Random Forest* (também baseado em árvore), quanto o *BiLSTM* possuem tecnicamente o mesmo valor no intervalo de confiança apresentado principalmente na Tabela 13.

Figura 32 – Conjunto de dados A_75 - MAE por estágio do processo.



Fonte: Elaborado pelo autor.

5.2.3 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, a pesar de dentro do intervalo de confiança apresentado na Tabela 13, as arquiteturas propostas neste trabalho que utilizam LSTM e Camadas de Atenção, tem tecnicamente a mesma capacidade de prever o tempo restante de conclusão de um processo de negócio.

Os modelos apresentados nos trabalhos relacionados diferem na complexidade da LSTM (Tax *et al.* (2017) e Navarin *et al.* (2017)) e na existência de uma LSTM para suportar

as sequências (Bukhsh *et al.* (2021)), apesar de ambos os trabalhos utilizarem *embeddings*. O melhor modelo apresentado neste trabalho é mais robusto e consegue produzir resultados com desempenho superior a trabalhos relacionados, bem como outras estratégias mais tradicionais de aprendizado de máquina.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho investigou diferentes abordagens para predição do tempo restante para conclusão de um processo. Dentre as abordagens avaliadas neste trabalho, o uso de aprendizado profundo com camadas de rede recorrente e de autoatenção lidam bem com *logs* de eventos reais, independente de ter instâncias automatizadas e manuais no mesmo conjunto de dados ou processos de negócios de grande flexibilidade, ou até mesmo de naturezas totalmente distintas.

Como trabalhos futuros, é importante considerar outros conjuntos de dados, explorar *logs* de diferentes competições voltadas à monitoramento preditivo de processos, realizar uma verificação de como arquiteturas mais recentes podem contribuir na melhoria da solução desse problema, comparar outros algoritmos e modelos clássicos com os atuais, realizar mesclas dessas arquiteturas tais como as propostas de Ma *et al.* (2023) e seguir com a disponibilização de base de dados e repositório¹ com os experimentos desenvolvidos para dar suporte a novas práticas, pesquisas e reproduções.

¹ <https://github.com/RonildoSilva/PTRC_PN_DL>

REFERÊNCIAS

- AALST, W. V. D. Process mining. **Communications of the ACM**, ACM New York, NY, USA, v. 55, n. 8, p. 76–83, 2012.
- AALST, W. V. D. Data science in action. In: **Process mining**. [S.l.]: Springer, 2016. p. 3–23.
- ACHARYA, M. S.; ARMAAN, A.; ANTONY, A. S. A comparison of regression models for prediction of graduate admissions. In: IEEE. **2019 International Conference on Computational Intelligence in Data Science (ICCIDS)**. [S.l.], 2019. p. 1–5.
- BARATA, R. d. C. B. Dez coisas que você deveria saber sobre o qualis. **Boletim Técnico do PPEC**, v. 2, n. 1, p. 17p–17p, 2017.
- BOX, G. E.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. **Time series analysis:: forecasting and control**. [S.l.]: John Wiley & Sons, 2015.
- BUKSHSH, Z. A.; SAEED, A.; DIJKMAN, R. M. Processtransformer: Predictive business process monitoring with transformer network. **arXiv preprint arXiv:2104.00721**, 2021.
- CHOLLET, F. **Deep learning with Python**. [S.l.]: Simon and Schuster, 2017.
- CHOLLET, F. **Deep learning with Python**. [S.l.]: Simon and Schuster, 2021.
- DONGEN, B. van. **BPI Challenge 2012**. [S.l.]: Eindhoven University of Technology, 2012. Disponível em: https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204/1. Acesso em: 04 ago. 2023.
- GALPHADE, M.; MORE, N.; NIKAM, V.; BANERJEE, B.; KIWELEKAR, A. W. Understanding deep learning: Case study based approach. **Deep Learning and Edge Computing Solutions for High Performance Computing**, Springer, p. 135–148, 2021.
- HUSEIN, M.; CHUNG, I.-Y. Day-ahead solar irradiance forecasting for microgrids using a long short-term memory recurrent neural network: A deep learning approach. **Energies**, MDPI, v. 12, n. 10, p. 1856, 2019.
- JIAO, Q.; LIU, M.; NING, B.; ZHAO, F.; DONG, L.; KONG, L.; HUI, M.; ZHAO, Y. Image dehazing based on local and non-local features. **Fractal and Fractional**, MDPI, v. 6, n. 5, p. 262, 2022.
- JIAO, Z.; HU, P.; XU, H.; WANG, Q. Machine learning and deep learning in chemical health and safety: a systematic review of techniques and applications. **ACS Chemical Health & Safety**, ACS Publications, v. 27, n. 6, p. 316–334, 2020.
- KALENKOVA, A.; AGEEV, A.; LOMAZOVA, I. A.; AALST, W. M. van der. E-government services: Comparing real and expected user behavior. In: SPRINGER. **International Conference on Business Process Management**. [S.l.], 2017. p. 484–496.
- KUMAR, A. **Understanding Linear Regression and Regression Error Metrics Dataquest**. 2023. Disponível em: <https://www.dataquest.io/blog/understanding-regression-error-metrics/>. Acesso em: 02 mar. 2023.
- LAKSHMANAN, V.; ROBINSON, S.; MUNN, M. **Machine learning design patterns**. [S.l.]: O'Reilly Media, 2020.

- LARSON, R.; FARBER, B. **Elementary statistics**. [S.l.]: Pearson Education Canada, 2019.
- LOOY, A. V. **Business process maturity: A comparative study on a sample of business process maturity models**. [S.l.]: Springer Science & Business Media, 2014.
- MA, H.; YANG, P.; WANG, F.; WANG, X.; YANG, D.; FENG, B. Short-term heavy overload forecasting of public transformers based on combined lstm-xgboost model. **Energies**, MDPI, v. 16, n. 3, p. 1507, 2023.
- MAINDONALD, J.; BRAUN, J. **Data analysis and graphics using R: an example-based approach**. [S.l.]: Cambridge University Press, 2006. v. 10.
- MAURO, A. D.; GRECO, M.; GRIMALDI, M. A formal definition of big data based on its essential features. **Library Review**, Emerald Group Publishing Limited, 2016.
- MITCHELL, T. M. Machine learning. 1st edition. **IBM Journal of Research and Development**, McGraw-Hill, Inc., 1997.
- NAVARIN, N.; VINCENZI, B.; POLATO, M.; SPERDUTI, A. Lstm networks for data-aware remaining time prediction of business process instances. In: IEEE. **2017 IEEE Symposium Series on Computational Intelligence (SSCI)**. [S.l.], 2017. p. 1–7.
- PARK, G.; SONG, M. Predicting performances in business processes using deep neural networks. **Decision Support Systems**, Elsevier, v. 129, p. 113191, 2020.
- PASCHEK, D.; LUMINOSU, C. T.; DRAGHICI, A. Automated business process management–in times of digital transformation using machine learning or artificial intelligence. In: EDP SCIENCES. **MATEC Web of Conferences**. [S.l.], 2017. v. 121, p. 04007.
- POLATO, M. **Dataset belonging to the help desk log of an Italian Company**. [S.l.]: University of Padova, 2017. Disponível em: https://data.4tu.nl/articles/dataset/Dataset_belonging_to_the_help_desk_log_of_an_Italian_Company/12675977/1. Acesso em: 02 mar. 2023.
- PONSARD, C.; DARIMONT, R. Towards goal-oriented analysis and redesign of bpmn models. In: **MODELSWARD**. [S.l.: s.n.], 2019. p. 527–533.
- RASCHKA, S.; MIRJALILI, V. Python machine learning: Machine learning and deep learning with python. **Scikit-Learn, and TensorFlow. Second edition ed**, 2017.
- REIJERS, H. A. Business process management: The evolution of a discipline. **Computers in Industry**, Elsevier, v. 126, p. 103404, 2021.
- ROSEN, K. H. **Discrete mathematics and its applications**. [S.l.]: The McGraw Hill Companies,, 2007.
- SALEM, F. M. **Recurrent Neural Networks**. [S.l.]: Springer International Publishing, 2022. Disponível em: <https://doi.org/10.1007/978-3-030-89929-5>. Acesso em: 17 fev. 2023.
- SHUMWAY, R. H.; STOFFER, D. S.; STOFFER, D. S. **Time series analysis and its applications**. [S.l.]: Springer, 2000. v. 3.
- SILVA, R. O. da; MAGALHÃES, R. P.; CRUZ, L. A.; SOUZA, C. P. de; VASCONCELOS, D. R. de; MACÊDO, J. A. F. de. Predição de tempo restante para conclusão de processos de negócio utilizando aprendizado profundo. In: SBC. **Anais do XXXVIII Simpósio Brasileiro de Bancos de Dados**. [S.l.], 2023. p. 141–153.

- SOUGHT, P. **Programmer Sought Decision Trees (b) decision tree regression**. 2021. Disponível em: <https://www.programmersought.com/article/26023353718/>. Acesso em: 05 jun 2023.
- STJEPIĆ, A.-M.; IVANČIĆ, L.; VUGEC, D. S. Mastering digital transformation through business process management: Investigating alignments, goals, orchestration, and roles. **Journal of entrepreneurship, management and innovation**, v. 16, n. 1, p. 41–74, 2020.
- SUPARWITO, H. Factors influencing the difficulty level of the subject: Machine learning technique approaches. **International Journal of Applied Sciences and Smart Technologies**, v. 1, n. 1, p. 65–82, 2019.
- TAN, C. W.; BERGMEIR, C.; PETITJEAN, F.; WEBB, G. I. **Time Series Extrinsic Regression**. 2021.
- TAX, N.; VERENICH, I.; ROSA, M. L.; DUMAS, M. Predictive business process monitoring with lstm neural networks. In: SPRINGER. **International Conference on Advanced Information Systems Engineering**. [S.l.], 2017. p. 477–492.
- VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.
- VENKATESWARAN, P.; MUTHUSAMY, V.; ISAHAGIAN, V.; VENKATASUBRAMANIAN, N. Robust and generalizable predictive models for business processes. In: SPRINGER. **Business Process Management: 19th International Conference, BPM 2021, Rome, Italy, September 06–10, 2021, Proceedings**. [S.l.], 2021. p. 105–122.
- VENUGOPAL, I.; TÖLLICH, J.; FAIRBANK, M.; SCHERP, A. A comparison of deep-learning methods for analysing and predicting business processes. In: IEEE. **2021 International Joint Conference on Neural Networks (IJCNN)**. [S.l.], 2021. p. 1–8.
- VERENICH, I. **Helpdesk**. [S.l.]: Mendeley, 2016. Disponível em: <https://data.mendeley.com/datasets/39bp3vv62t/1>. Acesso em: 02 mar. 2023.
- VERENICH, I.; DUMAS, M.; ROSA, M. L.; MAGGI, F. M.; TEINEMAA, I. Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. **ACM Transactions on Intelligent Systems and Technology (TIST)**, ACM New York, NY, USA, v. 10, n. 4, p. 1–34, 2019.
- WANG, J.; YU, D.; LIU, C.; SUN, X. Outcome-oriented predictive process monitoring with attention-based bidirectional lstm neural networks. In: IEEE. **2019 IEEE International Conference on Web Services (ICWS)**. [S.l.], 2019. p. 360–367.
- WATT, J.; BORHANI, R.; KATSAGGELOS, A. K. **Machine Learning Refined: Foundations, Algorithms, and Applications**. [S.l.]: Cambridge University Press, 2016.
- YESHPANOV, R.; KHASSANOV, Y.; VAROL, H. A. Kaznerd: Kazakh named entity recognition dataset. **arXiv preprint arXiv:2111.13419**, 2021.
- YU, Y.; SI, X.; HU, C.; ZHANG, J. A review of recurrent neural networks: Lstm cells and network architectures. **Neural computation**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info ..., v. 31, n. 7, p. 1235–1270, 2019.

ZHENG, A.; CASARI, A. **Feature engineering for machine learning**: principles and techniques for data scientists. [S.l.]: "O'Reilly Media, Inc.", 2018.