

IX SEMCITEC - SEMANA DO CONHECIMENTO.

**RELATO DE EXPERIÊNCIA: DETECÇÃO DE ANOMALIAS EM FILAS DE CALL
CENTER.**

Autores: Marcus Vinicius Galvão Rehm – Bacharel em Ciência Da
Computação. Universidade Federal Da Bahia (UFBA).

Samuel Leonardo Barboza Almeida – *Lato Sensu* em Arquitetura e
Gestão de Infraestrutura de TI. Faculdade de Informática e
Administração Paulista (FIAP).

GUARULHOS – SÃO PAULO.

1. INTRODUÇÃO

Soluções baseadas em aprendizado de máquina, especialmente as baseadas em *Deep Learning*, são utilizadas para resolver uma grande diversidade de problemas.

Em diversos segmentos de serviços brasileiros, a falta de infraestrutura é um grande problema. Segundo a Confederação Nacional da Indústria (CNI), foram investidos pouco mais de 2% do PIB em infraestrutura nos últimos 20 anos. (R. CRUZ FILHO Osvaldo, 2017).

A falta de instrumentos de monitoração dificulta a prestação de serviços em diversos aspectos. A indisponibilidade de um serviço pode causar grandes impactos econômicos e sociais nas mais variadas escalas. (SOUZA, 2006)

Para suprir esta deficiência, encontrou-se em dados de chamadas de *call center*, aliados a utilização de algoritmos de detecção de anomalias, uma solução útil para identificar eventos que podem indicar problemas atuais ou prever futuras falhas, acarretando a indisponibilidade de um serviço.

Este tipo de análise viabiliza ações pró-ativas, muitas vezes on-line ou perto do tempo real, para a identificação de falhas em processos produtivos, monitoração de sistemas sensíveis, identificação de problemas em serviços, etc.

Neste trabalho, foi relatada a experiência de desenvolvimento de um projeto de software para detecção de anomalias em filas de chamadas de *call center* utilizando uma Rede Neural Recorrente Autoregressiva do tipo *Long Short Term Memory* (LSTM), baseada na metodologia DeepAR, onde são utilizadas diversas séries temporais para treinar uma rede generalista capaz de fazer previsões de diversas séries do mesmo gênero (com ou sem histórico). Este conceito é mais eficiente do que algoritmos tradicionais para previsão de séries temporais como ARIMA e possibilita que modelos mais complexos sejam treinados sem *overfitting*. (SALINAS; FLUNKERT; GASTHAUS, 2017).

2. DESCRIÇÃO DETALHADA DA EXPERIÊNCIA

A internet das coisas é um assunto difundido e já existem ferramentas tecnológicas disponíveis para muitos serviços, tais como: distribuição de energia, água e gás. Porém, no contexto nacional, estes ainda sofrem com a falta de investimentos para a difusão da utilização destas tecnologias e falhas que poderiam ser identificadas de forma rápida, são muitas vezes detectadas tardiamente e com grande impacto na satisfação do cliente e na imagem da empresa. Portanto, a motivação por trás deste trabalho, surgiu da lacuna existente no monitoramento de pontos dos processos de operação.

Serviços como plataformas de e-commerce, provedores de TV por assinatura e outros que possuem interação com clientes por meio de *call center*, podem se beneficiar desta solução.

O presente trabalho visa tentar contornar a falta de monitoramento descrita acima de forma rápida e com baixo custo, podendo ser integrado ao processo de operação das empresas sem grande impacto nos fluxos. Para isso, basta o acesso a base de chamadas das filas relacionadas aos serviços que devem ser monitorados. Esta base de dados nada mais são do que as séries temporais da quantidade de chamadas das filas extraídas e contabilizadas da base de dados dos sistemas de operação das empresas.

Após análises das séries temporais de filas de chamadas de *call center*, foi identificado que existem comportamentos que podem apontar alguns problemas. Utilizando a metodologia DeepAR durante a fase de treinamento, um único modelo baseado em uma rede neural capaz de fazer previsões para estas séries, foi implementado.

O objetivo final é identificar em cada série, em um determinado intervalo de tempo, quais são os valores aceitáveis para a quantidade de chamadas e com isso, caso o valor observado (real) esteja estatisticamente fora do intervalo considerado aceitável (previsto), o sistema identifica que a observação pode ser uma anomalia e dispara um alerta sobre um possível problema para os responsáveis ou toma algum tipo de ação.

Durante a fase de processamento inicial das *features*, foram coletadas diversas séries temporais de solicitações da mesma natureza de centrais de *call centers* em diferentes localidades de uma empresa.

Estes registros contém a quantidade acumulada de chamadas numa determinada fila em diversos pontos do tempo, agrupados por hora (0 a 23) e por dia da semana. Todos os dias, às 0h, estas filas são reiniciadas.

Os campos iniciais são:

- CIDADE - Cidade em que a solicitação de call center foi registrada.
- DT_HR_ABERTURA - Ponto no tempo com registro de ano, mês, dias e hora.
- OCORRENCIAS - Quantidade de ocorrências registradas desde a hora zero até aquele ponto do tempo.

A Tabela 1, a seguir, representa os dados iniciais que foram utilizados para alimentar o algoritmo. Estes não tem nenhum tipo de pré-processamento.

Tabela 1 - Dados iniciais

	CIDADE	DT_HR_ABERTURA	OCORRENCIAS
0	CIDADE A	2015-06-01 07:00:00	1
1	CIDADE A	2015-06-01 08:00:00	2
2	CIDADE A	2015-06-01 14:00:00	1
3	CIDADE A	2015-06-01 15:00:00	1
4	CIDADE A	2015-06-01 16:00:00	1
...	...	...	...
13302	CIDADE B	2018-04-03 15:00:00	2
13303	CIDADE B	2018-04-04 07:00:00	2
13304	CIDADE B	2018-04-04 08:00:00	2
13305	CIDADE B	2018-04-04 09:00:00	1
13306	CIDADE B	2018-04-04 10:00:00	2

2.1. Preparação dos Dados:

Podem existir pontos no tempo em que, por um motivo qualquer, não houve coleta da fila de chamadas. Para estes pontos, o campo OCORRÊNCIAS é preenchido com o valor 0 para que a série fique completa. Após este procedimento, algumas *features* complementares foram criadas com base nas métricas iniciais. São elas:

- DIA_SEMANA_ABERTURA - Dia da semana, baseado no campo DT_HR_ABERTURA.
- HORA_ABERTURA - Hora, baseado no campo DT_HR_ABERTURA.

- MES - Mês, baseado no campo DT_HR_ABERTURA.
- ACUMULADO_NO_DIA - Somatória todos os valores do campo OCORRENCIA do dia, acumulados até aquele ponto do tempo.
- ACUMULADO_NO_DIA_T-1 - Valor apresentado no campo ACUMULADO_NO_DIA na hora anterior (T-1).

A Tabela 2, a seguir, representa os dados já transformados e formatados, prontos para serem imputados no algoritmo.

Tabela 2 - Dados processados.

	ACUMULADO_NO_DIA	ACUMULADO_NO_DIA_T-1	MES	HORA_ABERTURA	DIA_SEMANA_ABERTURA	CIDADE
2018-01-01 00:00:00	0	14.0	1	0	0	CIDADE D
2018-01-01 01:00:00	0	0.0	1	1	0	CIDADE D
2018-01-01 02:00:00	0	0.0	1	2	0	CIDADE D
2018-01-01 03:00:00	0	0.0	1	3	0	CIDADE D
2018-01-01 04:00:00	0	0.0	1	4	0	CIDADE D

Com os dados processados, os *datasets* são divididos em três *datasets* menores: treino, validação e teste. O critério da divisão é baseado em períodos de tempo e foram utilizados durante as fases de treinamento e teste da rede neural.

2.2. Processamento:

Nesta fase foram aplicados procedimentos para preparar os dados a serem imputados no algoritmo da rede neural.

As variáveis foram separadas de acordo com seu tipo. Como valores discretos temos os campos ACUMULADO_NO_DIA e ACUMULADO_NO_DIA_T-1, como valores categóricos temos os campos MES, HORA_ABERTURA e DIA_SEMANA_ABERTURA e CIDADE.

As variáveis categóricas são tratadas utilizando o conceito de *One Hot Encoding* e as variáveis discretas são colocadas em escalas menores para que otimização da rede neural ocorra de maneira mais rápida. Para a variável CIDADE foi utilizado o conceito de *embedding layer* pois o espaço vetorial consequente da alta cardinalidade desta variável tornaria impossível o treino do modelo. Após esse procedimento, cada tupla é transformada em um *array*.

A Figura 1, a seguir, representa os dados já transformados e formatados convertidos em *array*.

Figura 1 - Dados processados convertidos em *array*.

```
array([[[ 0.,  0.,  0., ...,  0.,  0.,  0.]],  
      [[ 2.,  0.,  0., ...,  0.,  0.,  0.]],  
      [[ 2.,  0.,  0., ...,  0.,  0.,  0.]],  
      ...,  
      [[174.,  0.,  0., ...,  0.,  1.,  0.]],  
      [[183.,  0.,  0., ...,  0.,  1.,  0.]],  
      [[194.,  0.,  0., ...,  0.,  1.,  0.]])
```

2.3. Algoritmo:

Tomando como base a metodologia de treinamento DeepAR, foi desenvolvida uma rede neural recorrente do tipo LSTM devido à natureza do problema a ser resolvido. Redes recorrentes são indicadas para a previsão de dados sequenciais e dependentes. (GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron, 2016).

Para implementar o algoritmo, dois *frameworks* foram usados para o desenvolvimento da rede neural: Keras e Tensorflow. Estes pacotes estão disponíveis para a linguagem Python e contém diversas funções que são muito úteis e facilitam o desenvolvimento de aplicações que utilizam códigos complexos, como LSTM.

As séries usadas para treinamento e previsão do modelo contém muitos zeros pois, nas primeiras horas do dia, durante a madrugada, é raro que algum chamado seja aberto. Por conta disso, foi utilizada a distribuição Binomial Negativa Inflada de Zeros (ZINB) como função de custo. Esta distribuição é interessante pois oferece uma maneira segura e consistente para modelar este tipo de situação. (KELVIN K. W. Yau¹, KUI Wang and ANDY H. Lee, 2003)

A ZINB assume que os dados podem ter duas naturezas distintas: apenas zero ou valores que estão em uma distribuição binomial negativa. O resultado é um experimento de Bernoulli que tem o intuito de determinar se este valor é zero ou um valor discreto.

O objetivo da rede neural é prever a quantidade de chamadas acumuladas (K) em um ponto no tempo e a probabilidade de sucesso deste valor (θ).

Com relação a estrutura da rede neural, a quantidade e organização das camadas, bem como a quantidade de neurônios e a função de ativação utilizada, foram determinadas de modo empírico. Nenhuma técnica em especial foi utilizada para otimizar ou ajudar na escolha das configurações.

Durante o treinamento, a alternância das séries temporais utilizadas a cada *batch* de dados dentro de cada *epoch* foi realizada como sugerido na metodologia DeepAR.

2.4. Predição e Avaliação dos Resultados:

Após o processo de treinamento, foi feita uma previsão para um determinado período (*backtesting*). Desta forma, foi possível saber o que seria o comportamento normal de cada série, métricas de dispersão e as probabilidades de problema. Os valores de dispersão são obtidos utilizando as fórmulas da Distribuição Binomial Negativa e a métrica de valor máximo (UPPER_BOUND) é o valor limite para o qual K é considerado fora do normal. (DEVORE, Jay L., 2009).

A Tabela 3, a seguir, representa os resultados da execução do algoritmo.

Tabela 3 - Resultados do Algoritmo.

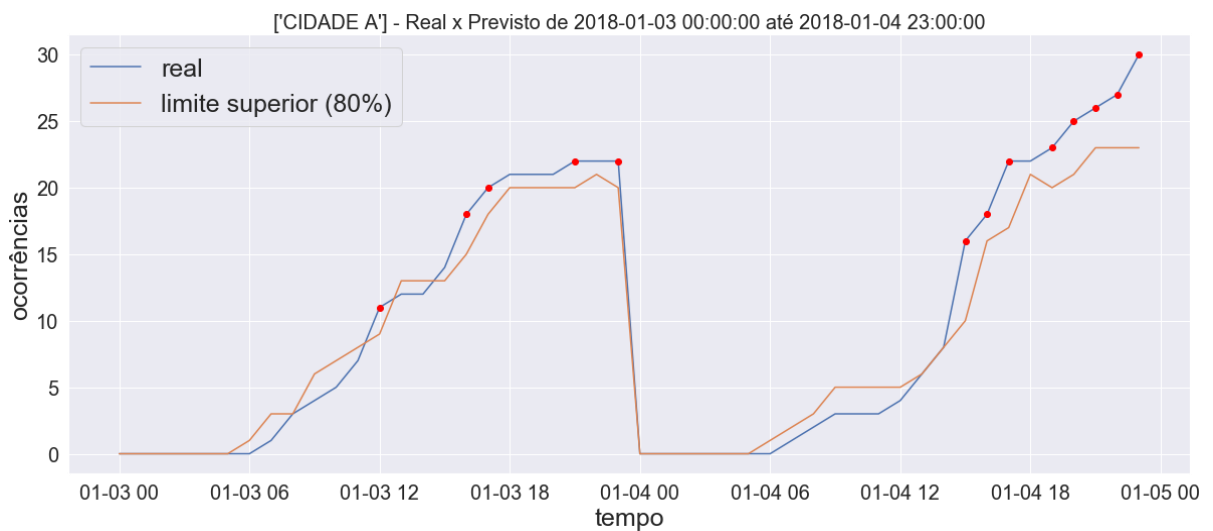
	CIDADE	MEDIA_PREVISTA	VARIANCE	STD	ACUMULADO_NO_DIA	UPPER_BOUND	PROB_PROBLEMA
2018-01-03 00:00:00	CIDADE A	0.135099	0.140713	0.38	0	0.0	0.12
2018-01-03 01:00:00	CIDADE A	0.196849	0.382421	0.62	0	0.0	0.13
2018-01-03 02:00:00	CIDADE A	0.214572	0.502181	0.71	0	0.0	0.13
2018-01-03 03:00:00	CIDADE A	0.214557	0.484010	0.70	0	0.0	0.13
2018-01-03 04:00:00	CIDADE A	0.208953	0.443241	0.67	0	0.0	0.13
2018-01-03 05:00:00	CIDADE A	0.321560	0.907104	0.95	0	0.0	0.17
2018-01-03 06:00:00	CIDADE A	0.747904	2.252989	1.50	0	1.0	0.34
2018-01-03 07:00:00	CIDADE A	1.502217	5.600717	2.37	1	3.0	0.80
2018-01-03 08:00:00	CIDADE A	1.987274	2.590542	1.61	3	3.0	0.84
2018-01-03 09:00:00	CIDADE A	4.095403	5.966654	2.44	4	6.0	0.84
2018-01-03 10:00:00	CIDADE A	4.692477	7.010145	2.65	5	7.0	0.86

Sempre que um valor realizado (ACUMULADO_NO_DIA) se encontra acima do valor máximo previsto (UPPER_BOUND) e a probabilidade de problema (PROB_PROBLEMA) para aquele ponto no tempo é superior a 94%, um ponto de alerta é gerado. Este ponto pode ser

interpretado como um possível problema e faz com que o mesmo seja identificado de maneira mais rápida, antes que este generalize e se torne maior e seja reportado sem que haja nenhuma análise humana.

A Figura 2, a seguir, representa a comparação entre a série temporal prevista pelo algoritmo e a série realizada, pontuando onde houveram desvios consideráveis e que foram apontados como anomalias.

Figura 2 - Séries temporais e anomalias.



3. RELACIONAR A EXPERIÊNCIA COM A TEORIA

Neste trabalho foram aplicado três conceitos: DeepAR, Redes Neurais Recorrentes e Distribuição Binomial Negativa.

DeepAR é uma metodologia que descreve como uma Rede Neural Recorrente deve ser treinada, para que esta possa ser generalizada de forma a aprender um modelo global, com base nos dados históricos de diversas séries temporais utilizadas no treinamento. Isso se aplica muito bem ao contexto descrito neste trabalho, pois um único modelo pode ser utilizado para realizar as previsões de mais de uma localidade, aumentando assim a acurácia dos resultados e reduzindo a complexidade de manutenção de vários modelos. (SALINAS; FLUNKERT; GASTHAUS, 2017).

Redes Neurais Recorrentes são utilizadas para prever dados sequenciais onde existe dependência entre o registro atual e o registro anterior. Dessa forma, o modelo se baseia no status $t-1$ da fila como *input* para a previsão. No experimento foi implementado lotes de 24 observações prévias para “esquentar” a memória da rede neural e com isso prever o tempo t logo após as 24 observações, ou seja, observaram-se as últimas 24 horas para prever o valor mais provável da próxima hora do dia. (GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron, 2016).

Distribuição Binomial Negativa é um conceito estatístico baseado no experimento de Bernoulli, em que trabalha com a quantidade mínima de testes necessárias para que um evento de sucesso ocorra. (DEVORE, Jay L., 2009)

4. CONSIDERAÇÕES FINAIS

Dado o contexto inicial referente a falta de investimentos e infraestrutura no Brasil, foi apresentado uma maneira de contornar problemas relacionados a falta de instrumentação utilizando dados de *call center*. Estes dados são séries temporais de filas de chamadas.

Após análises nestes dados, foi identificado que existem comportamentos que podem apontar alguns problemas. Utilizando inteligência artificial, foi implementada uma rede neural combinada com análises estatísticas para detecção de anomalias. Estas anomalias são possíveis problemas ou falhas no fornecimento de um serviço.

A combinação dos conceitos utilizados (Metodologia DeepAr, Redes Neurais Recorrentes e Distribuição Binomial Negativa) oferece uma solução confiável e robusta para previsão de séries temporais e detecção de anomalias, trazendo a luz informações que só estariam disponíveis caso ferramentas de monitoração fossem implementadas.

Esta solução não tem nenhum impacto na operação de uma empresa ou instituição e traz benefícios diretos a curto prazo, dado que a operação passa a ter mais informações disponíveis a partir do momento da implementação.

5. REFERÊNCIAS

1. DEVORE, Jay L.. **Probabilidade e estatística : para engenharia e ciências**. 6th ed. Pinheiro J, Silva N da, editors. Cengage Learning, 2009.
2. GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. 1. ed. Cambridge: The MIT Press, 2016.
3. KELVIN K. W. Yau¹, KUI Wang and ANDY H. Lee. **Zero-inflated negative binomial mixed regression modeling of over-dispersed count data with extra zeros**. Biometrical J., 2003.
4. R. CRUZ FILHO Osvaldo. **Gestão ou Gerenciamento, Análise das Falhas no Investimento de Infraestrutura no Brasil**. Revista Gestão e Gerenciamento - nº 6, 2017.
5. SALINAS, David; FLUNKERT, Valetin; GASTHAUS, Jan. **DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks**, 2017. Disponível em: <https://arxiv.org/abs/1704.04110>. Acesso: 8/7/2020 às 20h00.
6. SOUZA, Marcelo de. **Proposta de um sistema de gestão empregando instrumentação inteligente e redes de campo na automação do processo de tratamento de água**. Dissertação (Mestrado): Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Energia e Automação Elétricas, 2006.