



GRUPO DE ESTUDO DE SISTEMAS DE INFORMAÇÃO E TELECOMUNICAÇÃO PARA SISTEMAS ELÉTRICOS - GTL

SISTEMA INTELIGENTE DE TRIAGEM AUTÔNOMA DE CHAMADOS EMERGENCIAIS

**RAUL SENA FERREIRA(1);BRUNO MONTEIRO ACCIOLI DA SILVA(1);LUCAS MELO DE CHIARA(2);JOAO PEDRO FERREIRA DOS SANTOS FREITAS(3);HUGO DINIZ REBELO(1);JULIA DELGADO(1);TASSIO SIMIONI(1);ADONIS MARCEL DE ARAÚJO CARVALHO(1)
RADIX ENGENHARIA E DESENVOLVIMENTO DE SOFTWARE S/A(1);CPFL SANTA CRUZ(2);CPFL PAULISTA(3)**

RESUMO

Gerenciar chamados de clientes relacionados a problemas técnicos é uma tarefa de grande importância em uma Distribuidora de Energia. Devido aos picos de demanda por tal serviço, se faz necessário que o atendimento seja rápido e assertivo em diferenciar quais chamados necessitam de deslocamento de uma equipe de manutenção, de forma a evitar o envio de uma equipe desnecessariamente, reduzindo desperdícios logísticos em relação a tempo e custo. Neste Informe técnico falaremos sobre um sistema que faz uso de inteligência artificial para identificar quais são os possíveis chamados que irão gerar deslocamentos improcedentes por parte da equipe de manutenção, evitando custos e otimizando recursos.

PALAVRAS-CHAVE

Machine learning; deep learning; power utility

1.0 INTRODUÇÃO

Em média, 26% dos incidentes de interrupção de energia emergencial reportados por clientes da distribuidora CPFL não são de responsabilidade da distribuidora. Em outras palavras, são incidentes já solucionados, ou incidentes de responsabilidade do cliente (ex: problema com disjuntores), ou mesmo interrupção de energia elétrica por inadimplência ou programada. Quando uma equipe de campo é enviada para resolver tais incidentes gera-se um deslocamento improcedente (DI) dessa equipe, consumindo recursos da distribuidora que poderiam ser utilizados em outras áreas, provocando consequente queda na qualidade do serviço. Nos próximos parágrafos, será introduzida uma breve explicação de como funciona esse fluxo de atendimento.

Figura 1 ilustra o fluxo de informação de um incidente de interrupção de energia emergencial. Inicialmente o cliente cria um chamado entrando em contato com a distribuidora. Posteriormente, esse chamado é analisado captando informações da rede elétrica, podendo ou não agrupar outros chamados de outros usuários, gerando assim um incidente. Tal fluxo é atualizado ao longo do tempo, enquanto novas informações do incidente são descobertas. Esse efeito de atualização pode gerar múltiplas versões para o mesmo incidente, onde cada versão gerada torna a versão anterior obsoleta para execução do fluxo. Entretanto, ela permanece guardada na base de dados como histórico do incidente.

Após a geração do incidente, ele passa por um classificador de regras de confirmação automática, no qual analisa quais consumidores foram afetados e a quantidade dos mesmos. Caso esse incidente não seja confirmado automaticamente, ele é considerado um incidente suspeito e passa por uma etapa na qual avalia o uso do canal de atendimento automático URA (Unidade de Resposta Audível) por um conjunto de regras automáticas. Após essa etapa, o fluxo continua com a etapa de operação humana, na qual avalia-se, incidente a incidente, gerando uma lista de incidentes prioritários e fazendo a seleção do canal de atendimento para cada incidente. Caso o canal de comunicação do incidente seja callback, a operação humana continua, pois o callback é uma central de atendimento no qual atendentes humanos fazem ligações telefônicas para os clientes, nessa etapa pode ser escolhido também o canal URA.

Após a execução por um dos canais de comunicação e que ocorra um contato com o cliente, pode ser que exista duas opções de finalização desse incidente, a opção 1, que representa que haveria um deslocamento improcedente e a opção 2 na qual não deverá ocorrer um deslocamento improcedente (incidente válido).

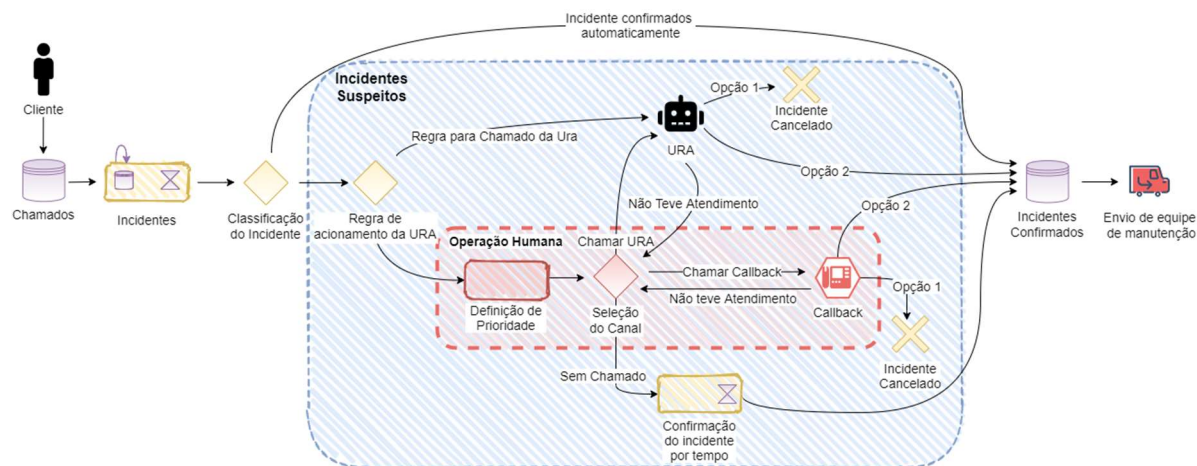


Figura 1- Fluxo de comunicação de um incidente.

Caso seja a opção 1, o incidente será cancelado, caso ocorra a opção 2, será pedido o envio de uma equipe de campo para o local do incidente. Caso não tenha sido possível contatar o cliente, independente do canal de comunicação, esse incidente volta para etapa de seleção de canal, sendo que esse loop poderá ocorrer até que se atinja um limiar de tempo definido pela distribuidora, caso isso ocorra, será enviada uma equipe de campo para solucionar o problema mesmo que ele não esteja confirmado. É a partir desse cenário onde ocorre iteração humana em grande quantidade para a detecção de incidentes improcedentes, com uma grande quantidade de incidentes suspeitos e com um limite temporal para solução que essa proposta se baseia.

Foram encontrados trabalhos semelhantes de sistemas para melhorar a gestão das equipes de campo (1,2), mas que não tentam resolver o problema de deslocamentos improcedentes. Foram encontrados também trabalhos para redução aos desperdícios de HH(homem-hora) (3,4,5), onde (3,4) propõe um canal de comunicação ao callback para reduzir a quantidade de deslocamentos improcedentes, porém sem utilização de sistemas de apoio à decisão. No trabalho (5) foi apresentado um sistema de apoio à decisão para priorização dos chamados para o callback. Porém, nesse trabalho foram utilizados apenas heurísticas para priorizar chamados que atendessem a certas condições pré-estabelecidas sem ser utilizados nesses sistemas os conceitos de aprendizado de máquina e inteligência artificial.

Em (16) e (17) os autores apresentam um sistema para deslocamentos improcedentes que utiliza IA para identificar tais chamados improcedentes e ordená-los em ordem de importância. Além disso, o sistema também conta com um processo de otimização usando MPC, para corrigir possíveis ordenações incorretas dadas pela IA do sistema. O presente informe técnico propõe um ecossistema que inclui um sistema similar ao proposto em (16) e (17), porém nossa proposta vai um passo além: um sistema que consegue cobrir não só a etapa de triagem de chamados improcedentes, mas todos os passos do processo de forma quase autônoma.

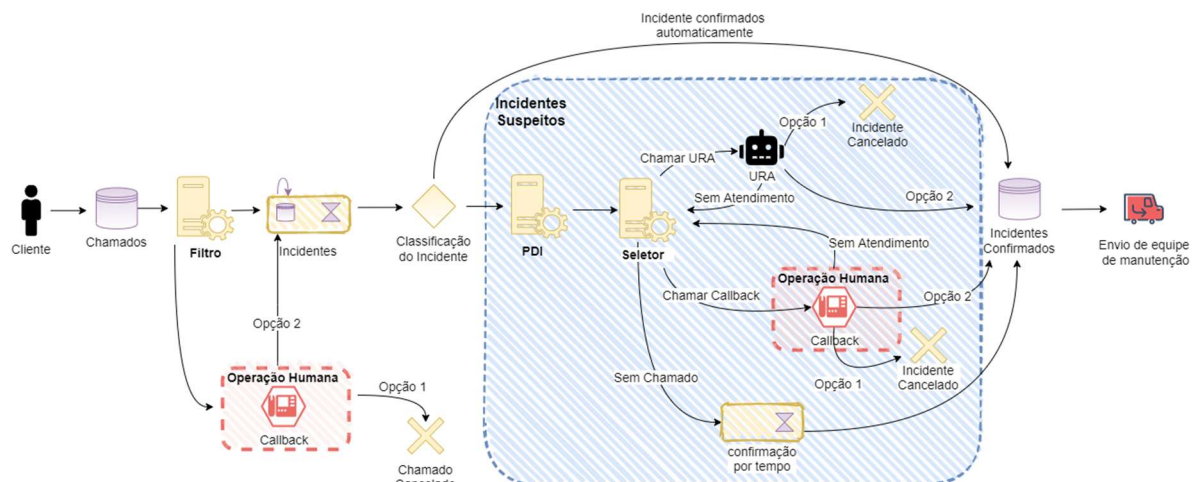
Esse informe técnico está organizado da seguinte forma: na seção 2 apresentada a Arquitetura do modelo, com suas características e nas seções seguintes serão detalhados cada etapa do ecossistema.

2.0 SOLUÇÃO PROPOSTA

Figura 2 ilustra a inclusão de 3 novas fases a que se referem esta proposta: filtro, PDI e seletor. O Filtro é a primeira etapa após a criação do chamado, no qual pretende-se fazer um filtro de chamados suspeitos antes que os mesmos passem pelo processo de geração de incidentes.

Esses chamados irão direto para o canal do callback de forma a agilizar o contato com cliente, acelerando o processo de comunicação. As etapas de identificação de prioridade e seleção de canal já dentro do conjunto de etapas feitas para os incidentes suspeitos, foram substituídas pelo PDI e Seletor respectivamente. Cada uma dessas etapas contém modelos de Aprendizado de Máquina com lógicas muito similares às antigas etapas, mas feita de forma autônoma, reduzindo para somente o canal de comunicação callback a operação humana no ecossistema de detecção de ineficiências.

A forma de desenvolvimento dessa solução permite um grande desacoplamento, tendo definidas suas formas de entradas e saídas e com total independência dos mesmos, providas pela definição prévia de entradas e responsabilidades de cada etapa.



Nas próximas seções desse informe técnico será mostrado como foi desenvolvida cada etapa inserida na solução, definindo o problema, solução e análises desenvolvidas.

Todas as análises desse informe técnico foram conduzidas num computador com processador *i7-8565U*, utilizando *Windows 10* como sistema operacional, o código da implementação usa *Python 3.8.8*, com as bibliotecas *tensorflow 2.4.1*, *pandas 1.1.5*, *numpy 1.19.5*, *pyproj 3.1.0*, *scipy 1.5.4*, *xgboost 1.3.3* e *scikit-learn 0.24.1*.

2.1 PDI

A etapa do PDI (Probabilidade de Deslocamento Improcedente) tem por objetivo, fazer a detecção de possíveis incidentes que gerarão DIs. Esse modelo foi pensado para ter uma saída contínua entre 0 a 100, que represente a probabilidade daquele incidente gerar ou não um DI. O motivo de modelar a saída do problema de forma linear, se dá pelo fato da seleção de canal, assim como propriamente os canais de comunicação utilizarem essa probabilidade para priorização dos incidentes.

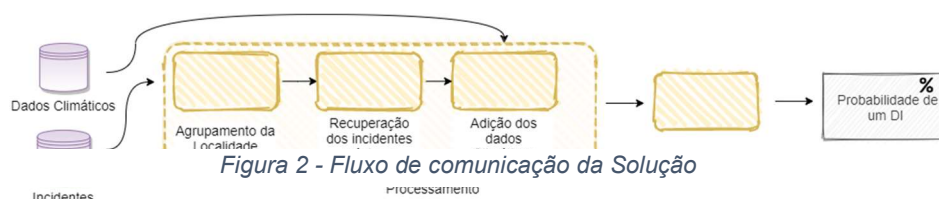


Figura 2 - Fluxo de comunicação da Solução

Figura 3 - Arquitetura do PDI

Representado na Figura 3, o PDI tem o objetivo de calcular uma probabilidade de cada incidente gerar um DI, sendo que essa probabilidade estará diretamente ligada com o grau de certeza que o modelo terá que o determinado incidente é ou não um DI. Essa operação inicia com o recebimento do incidente na etapa de processamento. Na etapa de processamento é buscado o grupo regional em que ocorreu o incidente. Esses grupos foram criados a partir de um modelo de agrupamento previamente treinado com as latitudes e longitudes de todas as unidades consumidoras conhecidas. Após a recuperação dos agrupamentos, são buscados os dados dos incidentes na vizinhança ocorridos num período de tempo próximo. Após a recuperação das informações de localidade, são buscados os dados climáticos da região onde ocorreu o incidente.

Após a etapa de processamento, os dados irão ao modelo que calcula uma probabilidade de aquele incidente gerar um DI, e com essa probabilidade, o PDI enviará essas informações a próxima etapa, que é o Seletor. Para a criação desse modelo com os dados dos incidentes, foram selecionadas 3 técnicas distintas de aprendizado supervisionado, o *Gradient Boosting*, a *Floresta Randômica*, e a *Rede Neural Feedforward* (6). Foi utilizado nestes experimentos a implementação da biblioteca *Keras* para a linguagem *Python*.

A segunda técnica utilizada para o PDI, o *Gradient Boosting* é um conjunto de técnicas com um considerável sucesso, diversas aplicações práticas, dada sua capacidade de ser customizável (8). A última técnica utilizada para o PDI, a *Floresta Randômica* é outra técnica que utiliza os conceitos de árvore de decisão. A vantagem dessa técnica é sua simplicidade e baixa quantidade parametrizações a serem feitas, com bons resultados de forma geral (9).

Para a execução desse experimento, foi escolhido um *dataset* de incidentes concluídos, e que ocorreram entre 1 de janeiro de 2021 até o dia 18 de maio de 2021 com 89.810 incidentes contendo os seguintes dados:

- Dados do Incidente

- Localidade Geográfica do Incidente e seu agrupamento.
 - Código do equipamento na rede afetado pelo incidente
 - Perigos relacionados ao incidente
 - Data da Criação do incidente
 - Data da última Revisão do Incidente
 - Status dos Incidentes na mesma região no momento de Criação
- Dados Climáticos
 - Temperatura
 - Intensidade do Vento
 - Direção do Vento
 - Intensidade do Vento em Rajada
 - Chuva em milímetros.
 - Probabilidade de Chuva do dia.

Para a etapa de agrupamento de localidade, é utilizado os dados de coordenada geográfica (latitude e longitude) que irão passar em um modelo de aprendizado não supervisionado *K-means* (10). Na etapa de recuperação dos incidentes próximos, são utilizadas as informações dos status dos incidentes próximos no período entre 1 hora anterior a criação do incidente até o momento do processamento desse incidente. Além disso, são utilizados dados do clima entre 4 horas anteriores a hora da criação até a hora da criação do incidente. Esses dados são periódicos, gerados a cada hora.

Para a avaliação do modelo usaremos 3 métricas distintas, a acurácia, o valor AUROC e o f1-score. A acurácia, a métrica mais simples e conhecida, vai comparar o valor predito e o real enquanto mais alto o valor, mais eficiente foi o modelo na classificação. Essa métrica vai de 0 a 1. A segunda métrica utilizada é o valor da AUROC (área sobre a curva ROC) e se trata de um valor que simplifica a curva ROC, que resumidamente demonstra o quão eficiente está o modelo independente da escolha do limiar de probabilidade, o que torna a sua apresentação muito interessante. O valor do AUROC varia de 0 até 1. Quanto maior o AUC, melhor (14).

Existem duas métricas complementares entre si que são a precisão e a sensibilidade. Enquanto a precisão avalia quantos dos itens selecionados são relevantes, o recall quer avaliar quantos itens relevantes foram selecionados. É partir dessas duas métricas que nasce o f1-score, que resumidamente é uma média harmônica entre essas duas métricas. A f1-score varia entre 0 e 1, e quanto maior melhor (15). Para o f1-Score e a acurácia nessa avaliação, valores acima de 50% de serem DI, serão classificados como DI, caso seja inferior a 50%, será considerado que o incidente foi avaliado como precedente.

<i>Técnica</i>	<i>Acurácia</i>	<i>AUROC</i>	<i>F1-Score</i>
<i>Floresta Randômica</i>	0.641	0.691	0.536
<i>Rede Neural</i>	0.608	0.678	0.551
<i>Gradient Boosting</i>	0.561	0.711	0.515

Tabela 1 - Resultado do Experimento do PDI

Olhando os resultados das 3 abordagens do modelo, vemos uma leve vantagem para o *Gradient Boosting* na métrica AUC sobre as outras, sendo que as outras duas outras métricas ficaram um pouco abaixo. A Floresta Randômica apresenta o melhor resultado na acurácia e com bons resultados tanto na AUROC quanto na f1-Score. Já a Rede Neural apresenta na f1-score, o melhor resultado entre as 3 técnicas. De forma geral, as técnicas obtiveram resultados bem interessantes e similares entre si. A acurácia e F1-Score são sensíveis ao limiar do experimento, que é de 0.5, e esse valor acabou penalizando mais o *Gradient Boosting*.

2.2 SELETOR

A etapa do Seletor tem por objetivo substituir a etapa de operação humana na escolha de qual incidente irá para cada canal de atendimento. A substituição dessa etapa visa tornar todo processo mais autônomo e adaptável. A execução dessa etapa era feita por operadores humanos de forma empírica, utilizando critérios subjetivos de disponibilidade dos canais de atendimento e chance do incidente gerar um DI.

Nessa etapa é feito a separação e envio dos incidentes para os canais de atendimento, de forma que maximize a detecção de incidentes que gerem deslocamentos improcedentes. Para a definição da arquitetura do seletor e seu funcionamento, foi feito um estudo de requisitos e análise de possíveis regras que envolvessem a seleção de um determinado canal para um determinado incidente.

A partir desse estudo foi percebida a necessidade de termos uma arquitetura que permitisse, com pouca alteração, a inserção de um possível novo canal, já que existem operações em paralelo ao projeto deste informe técnico que contemplam a inserção de novos canais de comunicação. Outro requisito avaliado, foram as regras que chamam automaticamente a URA, que são as seguintes:

- Incidente é emergencial.
- Incidente é relacionado a Falta de Energia.
- Incidente não é confirmado.
- Incidente não possui consumidores classificados como especiais.
- Incidente não possui alguma relação de prioridade.
- Incidente é considerado isolado.
- O incidente não deve ser classificado como energia restaurada.

Dados esses requisitos, foi pensado uma arquitetura de 2 etapas que pode ser visualizada na Figura 4, nas quais podemos definir como a avaliação da capacidade do canal de comunicação e a da Heurística de Seleção do canal. A etapa nomeada de Modelo de Capacidade, foi desenvolvida para ser uma etapa que avalia a capacidade momentânea do canal de comunicação. Já a segunda etapa, nomeada de Heurística da Seleção será a responsável pela separação dos incidentes para cada canal de comunicação.

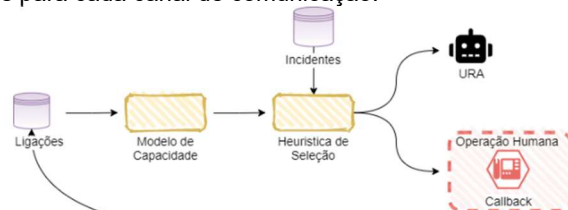


Figura 4 - Fluxo de Informação do Seletor

Ainda no estudo de requisitos, foi percebido que apesar da possibilidade do uso da URA, não era comum a opção do mesmo pelo atendente responsável, a opção era majoritariamente pelo uso do Callback, fazendo que o maior uso do canal URA fosse definido por regra já automatizadas previamente a definição da probabilidade de aquele incidente gerar um DI. Esse fato pode produzir uma considerável subutilização da URA, e gerar uma demanda além da capacidade do callback. A partir desse ponto, foi feito um estudo de efetividade e uso tanto do callback, quanto da URA para o entendimento do comportamento desses canais em relação aos seus usos.

A primeira análise foi de efetividade, que avalia a quantidade percentual de chamados que foram feitos e obtiveram ou não um contato. Para essa análise foram extraídos os dados da URA entre agosto de 2020 até dezembro de 2020, sendo que foram retirados os chamados que não obtiveram contato por alguma falha na ligação, seja por falha na própria URA e/ou falha no cadastro do cliente. Essas falhas não ocorrem no Callback por ser um atendimento humano.

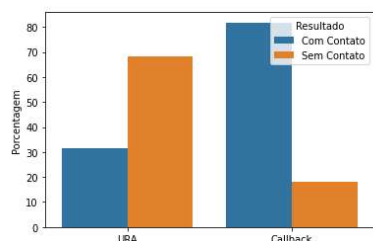


Figura 6 - Quantidade de Chamados com atendimento

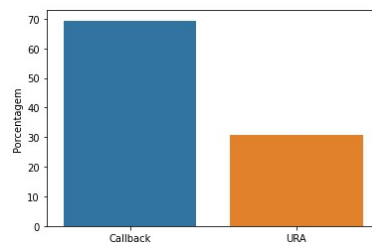


Figura 5 - Total de incidentes evitados comparativamente

Na Figura 6, vemos uma grande diferença na porcentagem dos atendimentos com um contato, demonstrando que o callback leva uma grande vantagem nesse quesito. Essa diferença pode ser explicada já pela forma de escolha do canal que é feita. Outro ponto que pode explicar essa diferença são as limitações técnicas na operação da URA, que dificulta a interação com usuário.

Uma outra forma de avaliar a capacidade do canal de atendimento de evitar DIs pode ser feita analisando os dados dos incidentes. Essa é uma métrica importante pois a diferença entre a efetividade no ato de conseguir uma fazer o atendimento pode ser compensado pela capacidade de gerar muitos atendimentos. Sendo assim, mesmo o canal tendo baixa efetividade, dada sua capacidade de realizar um alto volume de ligações, tem o poder evitar muitos incidentes. Essa análise foi feita utilizando a extração dos dados dos incidentes entre 08/2020 até 12/2020 demonstrada na Figura 5.

Vemos também nessa análise que existe uma clara vantagem do callback em relação a URA, porém é necessário ressaltar que esse resultado deve ser enviesado pelas regras de seleção do canal já existentes para determinados incidentes. A existência dessas regras pode levar a uma mudança na quantidade percentual de incidentes evitados, e isso aumenta pelo fato de existir uma seleção automática para URA, e um processo decisório empírico da seleção dos canais pelos operadores. Posterior a essa análise, foi buscado algum padrão específico no conjunto de dados

dos dois canais que poderia aumentar a efetividade para um sub grupo de clientes, mas não obtivemos nenhum padrão específico para algum sub grupo de clientes.

Dadas essas análises e observações feitas, conclui-se que o callback tem uma capacidade atual maior de evitar deslocamentos improcedentes, devido a capacidade de um atendimento obter contato com o cliente e as conhecidas limitações técnicas da URA. Sabendo disso, podemos montar a etapa da Heurística de seleção de canais, colocando os incidentes que tem maior probabilidade de ser um deslocamento improcedente, no canal de maior efetividade, que no caso é o callback.

2.2.1 MODELO DE CAPACIDADE DE ATENDIMENTO DO CALLBACK

O callback foi definido como canal prioritário, tornando-se o canal que receberá os incidentes com a maior probabilidade de ser DI. Uma etapa importante é saber qual é a capacidade do callback de atendimento em um determinado horário do dia. Para identificação desse atendimento, os dados utilizados foram a quantidade de atendimentos, a hora e os atendentes naquele determinado momento. Para os experimentos foram utilizadas 3 técnicas distintas: Modelo baseado em regras, *ARIMA* e o *RNN*.

O Modelo baseado em regras é o *baseline* desse experimento. Esse modelo pode ser definido como um conjunto de fórmulas capaz de calcular a capacidade de atendimento do callback, sendo que ele calcula inicialmente a capacidade de atendimento de um atendente qualquer em um determinado dia da semana, posteriormente ele calcula a produtividade a partir da capacidade e total de atendente por hora. Com essa produtividade por dia da semana multiplicada pela quantidade de atendimento e a quantidade de horas do turno é calculada a quantidade *N* de incidentes a serem atendidos pelo Callback.

$$\begin{aligned} capacidade_{atd}(d_{sem}) &= \frac{\sum_1^n eventos}{n} \\ Total_{AtendenteHora} &= \sum N_{atendente} * (horario_{ini} - horario_{fim}) \\ Produtividade(d_{sem}) &= \frac{capacidade_{atd}(d_{sem})}{Total_{AtendenteHora}} \\ N &= Produtividade(d_{sem}) * N_{atendente} * (horario_{ini} - horario_{fim}) \end{aligned}$$

O *ARIMA* (*autoregressive integrated moving average*) é um modelo autorregressivo de médias móveis que tem por objetivo prever um ponto futuro em uma série temporal. É um modelo estatístico amplamente utilizado previsão de séries temporais. Como o problema de previsão de capacidade pode ser generalizado como previsão de uma série temporal, foi utilizado nos experimentos esse modelo. (11)

A Rede neural recursiva (*RNN*), é um conjunto de técnicas de aprendizado supervisionado da classe de Redes neurais. Alguns autores já consideram esse conjunto de técnicas de aprendizado profundo (13). Atualmente a *RNN* possui algumas variações, nas quais além da *RNN* utilizaremos duas: *Long Short-Term Memory* (*LSTM*) e a *GRU*.

A *LSTM* (12) foi projetada com o intuito de evitar o problema de dependência de longo prazo, que pode ser definido pela dificuldade da rede de aprender informações de um passado mais distante, problema explorado no (13). Outra Variação da *RNN* e com muitas similaridades com a *LSTM*, a *GRU* foi proposta no artigo (18), como uma abordagem que também tenta evitar o problema da dependência de longo prazo, mas com algumas simplificações na sua arquitetura se comparada a *LSTM*, gerando um menor custo computacional, sem uma perda considerável em termos efetividade na solução do problema.

Para a avaliação das 3 técnicas, utilizaremos somente os dados de ligações do callback de 1 de outubro de 2020 até 31 de janeiro de 2021. Utilizaremos como métrica o *MSE* (*mean-square error*), que é uma métrica que avalia o erro médio do modelo penalizando os erros de maior magnitude. Serão utilizados 80% dos dados, para treino e 10% dos dados para teste, e o restante será utilizado para validação do modelo. O Modelo baseado em regras utilizou os últimos 60 dias dos dados de treinamento, e a seguinte escala de atendentes:

- 6:00 as 8:00
 - 2 atendentes
- 8:00 as 14:00
 - 7 atendentes
- 14:00 as 17:00
 - 9 atendentes
- 17:00 as 19:00
 - 4 atendentes

A técnica *ARIMA* foi utilizada com as variáveis $q=60$, $d=1$ e $p=12$. E por último a *RNN-Simples* utiliza 3 camadas intermediárias de 150, 75 e 50 neurônios respectivamente. A *RNN-LSTM* tem 5 camadas intermediárias de 300, 150, 75, 50 neurônios respectivamente. A *RNN-GRU* tem a mesma configuração de 5 camadas intermediárias de

300, 150, 75 e 50 neurônios respectivamente. Todas as *RNN* tem nas camadas intermediárias um regularizador $L2 = 10^{-5}$.

<i>Técnica</i>	<i>MSE</i>
<i>Modelo Baseado em Regras</i>	14.50
<i>ARIMA</i>	13.57
<i>RNN-Simples</i>	11.75
<i>RNN-LSTM</i>	10.68
<i>RNN-GRU</i>	11.45

Tabela 2- Resultado do Experimento 1 do Seletor

Os resultados mostram um expressivo aumento de qualidade quando se utiliza modelos de *RNN*, com uma redução significativa no *MSE* em relação ao *ARIMA* e no modelo baseado em regras. A partir desse resultado, foi desenvolvido um segundo experimento com os modelos baseados em *RNN* e adicionando, duas novas variáveis. Os dados da quantidade de atendentes no determinado dia e os identificadores únicos de cada atendente (*Id Atendente*), a fim de gerar um modelo com maior precisão.

<i>Técnica</i>	<i>Quantidade Atendentes</i>	<i>Id Atendente</i>	<i>MSE</i>
<i>RNN-Simples</i>	Sim	Não	9.95
<i>RNN-LSTM</i>	Sim	Não	8.51
<i>RNN-GRU</i>	Sim	Não	8.70
<i>RNN-Simples</i>	Não	Sim	9.17
<i>RNN-LSTM</i>	Não	Sim	9.83
<i>RNN-GRU</i>	Não	Sim	9.74
<i>RNN-Simples</i>	Sim	Sim	9.20
<i>RNN-LSTM</i>	Sim	Sim	8.80
<i>RNN-GRU</i>	Sim	Sim	8.89

Tabela 3 - Resultado do Experimento 2 do Seletor

Os resultados do experimento 2, mostram uma melhora expressiva com essas duas variáveis, independente *RNN* utilizada, sendo que o melhor modelo que ficou com *RMSE* de 8.51 com a técnica *RNN-LSTM*, com somente a variável quantidade de atendentes.

2.3 FILTRO

É a única etapa que não existia antes no fluxo de comunicação exibido na Figura 1. O filtro foi pensado para identificar chamados com alta chance de gerar deslocamentos improcedentes. Essa etapa é basicamente formada por um modelo de aprendizado supervisionado no qual de forma similar, ao *PDI*, a partir de um chamado gera uma probabilidade de ser um *DI*. Caso a probabilidade gerada seja alta o suficiente, o chamado irá diretamente para o callback, que é o único canal definido para essa tratativa.

Após a chegada desse chamado no callback, os atendentes irão proceder da mesma forma que procederiam caso fosse um incidente. Caso após o contato fique determinado que seria gerado um *DI*, o chamado é cancelado e nem chega a se tornar um incidente. Caso o cliente confirme o ocorrido, o chamado irá prosseguir de forma idêntica ao fluxo anterior, virando um incidente ainda assim podendo na etapa de incidente virar um incidente suspeito ou não.

A definição da utilização de somente um canal de atendimento faz com que a seleção de canal após o filtro não seja necessária. Mas conforme mostrado nas arquiteturas do *PDI* e seletor já apresentadas, uma inserção futura de um seletor em separado para o Filtro ou mesmo a inserção no mesmo seletor já desenvolvido seria possível com poucas alterações, já que os valores de entrada e saída do *PDI* e Filtro são similares e o seletor não possui uma utilização de algum dado do incidente que não seja o identificador único e sua probabilidade de *DI*.

Para avaliar o uso do filtro no ecossistema, foram utilizadas as técnicas de *Gradient Boosting* e Floresta Randômica, já explicadas na seção do *PDI*. Para essa avaliação, foram utilizados os chamados já concluídos com as variáveis de Data do chamado, Localidade, Capacidade do transformador, categoria do consumidor, número de chamados abertos no transformador e fase da unidade consumidora. A saída do modelo será a variável que indica a probabilidade do chamado gerar um *DI*.

Dentro desse conjunto de dados, uma informação importante é o número de chamados do transformador que será pré-processada, gerando uma outra variável chamada de suspeito de incidente coletivo. Essa variável é muito interessante, pois conhecidamente quando ocorre mais de um chamado no mesmo transformador, existe uma probabilidade alta de que esses chamados venham a se tornar um incidente coletivo, fator que fará esse incidente não ser considerado suspeito. Por isso, foram avaliados 2 modelos de formas distintas, uma utilizando somente, os

dados onde temos chamados que são únicos no transformador, chamados onde não são únicos no transformador, e por fim o conjunto de dados completos. Para essas avaliações utilizaremos métrica AUROC, que também já foi utilizada no modelo do PDI.

Técnicas	AUROC Global	AUROC Coletivos	AUROC Isolados
<i>Floresta Randômica</i>	0.699	0.706	0.613
<i>Gradient Boosting</i>	0.707	0.705	0.632

Tabela 4 - Resultado do Experimento do Filtro

Analizando os dados dessa análise, percebemos valores distintos nas técnicas nos casos que só tem um chamado no transformador (AUROC Isolados), nos casos que têm vários chamados no transformador (AUC Coletivos), e no conjunto de dados Completo (AUROC Global).

Percebemos que quando separamos o *Dataset*, a Floresta Randômica apresentou um resultado um pouco superior no AUROC coletivos, ao contrário do *Gradient Boosting* que foi superior no AUROC isolados e no AUROC global. Apesar de resultados muito semelhantes, de forma geral o *Gradient Boosting* apresentou um resultado superior, tendo melhor resultado global e no conjunto de chamados isolados, demonstrando que deve ser uma melhor solução utilizar somente o *Gradient Boosting*.

3.0 CONCLUSÃO

De forma geral foi visto nesse informe técnico um ecossistema para a triagem de incidentes de chamados emergenciais, onde foram obtidos resultados promissores em cada uma das etapas necessárias para o funcionamento completo do sistema, trazendo maior autonomia, menos custo operacional e melhor alocação de recursos.

Uma limitação deste trabalho é falta de avaliação do funcionamento em conjunto de todas as etapas, pois deve se fazer somente após um certo período de uso do ecossistema. A razão é que todas as etapas em algum grau são dependentes das outras. Eventuais mudanças podem ocorrer dentro dessas etapas, como pequenas variações no modelo ao longo do tempo para sua otimização e alterações no seletor de canais com eventuais adições de canais, tendo como um trabalho futuro de avaliação do impacto de um novo canal nos modelos e no seletor.

REFERENCIAS BIBLIOGRAFICAS

- (1) Rosa, A. R., Flor, D. d., Cruz, A. F., Teodoro, V. F., & Pereira, R. D. (2018). Novas funcionalidades para otimização do despacho e gestão de equipes coordenadas pelos Centros de Operação da Energia, incorporadas ao SIGOD. XXIII, Seminário Nacional de Distribuição de Energia Elétrica.
- (2) Freitas, V., Barbosa, R., & Bagui, R. (2017). Sistema Inteligente de Gestão de Serviço de Campo – GSC. IX Congresso de Inovação Tecnologia em Energia Elétrica.
- (3) Paula, F.A.S.; Morais, T.C.M.; Cruz, A.F.; Muniz, B.E.P.; Junior, A.A.S.; Pedroza, R.M. Combate aos desperdícios de HH - Ações na Operação voltadas aos Deslocamentos Improcedentes Técnicos, Impedimentos Comerciais, Reprovações de Vistoria, Reincidências e o Índice Ponderado de Eficiência Operacional. In: Seminário Nacional de Distribuição de Energia Elétrica, XXIII, 2018, Fortaleza.
- (4) Marques, M.E.M; Boas, C.V.V.; Castro, N.S.; Lima, D.R.C. Gestão sobre deslocamentos improcedentes. In: Seminário Nacional de Distribuição de Energia Elétrica, XXII, 2016, Curitiba.
- (5) Costa, V.A. Gestão e Priorização de Callback - Melhoria de performance. In: Seminário Nacional de Distribuição de Energia Elétrica, XXIII, 2018, Fortaleza.
- (6) Philip D. Wasserman. 1993. Advanced Methods in Neural Computing. John Wiley & Sons, EUA.
- (7) Rosasco, Lorenzo, et al. "Are loss functions all the same?" Neural computation 16.5 (2004)
- (8) Natekin, Alexey, and Alois Knoll. "Gradient boosting machines, a tutorial." Frontiers in neurorobotics 7 (2013): 21.
- (9) Ho, Tin Kam. "Random decision forests." Proceedings of 3rd international conference on document analysis and recognition. Vol. 1. IEEE, 1995.
- (10) Hamerly, Greg, and Charles Elkan. "Learning the k in k-means." Advances in neural information processing systems 16 (2004): 281-288.
- (11) Gilbert, Kenneth. "An ARIMA supply chain model." Management Science 51.2 (2005): 305-310.
- (12) Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
- (13) Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5.2 (1994): 157-166.
- (14) Fawcett, Tom. "An introduction to ROC analysis." Pattern recognition letters 27.8 (2006): 861-874.

- (15) Chicco, Davide, and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation." *BMC genomics* 21.1 (2020): 1-13.
- (16) Dal Pont, M. P., Ferreira, R. S., Teixeira, W. W., Lima, D. M., & Normey-Rico, J. E. (2019). "MPC with Machine Learning Applied to Resource Allocation Problem using Lambda Architecture." *IFAC-PapersOnLine*, 52(1), 550-555.
- (17) MARCOS VINÍCIUS PIMENTEL TEIXEIRA, JOÃO PEDRO FERREIRA DOS SANTOS FREITAS, RAUL SENA FERREIRA, ANDRÉ LUIZ DE MESQUITA MELO, TÁSSIO SIMIONI. "Aplicação de Inteligência Artificial no processo de triagem de chamados emergenciais da distribuidora visando redução de deslocamentos improcedentes no serviço de campo." XXV SNPTEE SEMINÁRIO NACIONAL DE PRODUÇÃO E TRANSMISSÃO DE ENERGIA ELÉTRICA, 2019.
- (18) Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

DADOS BIOGRÁFICOS



I am a data scientist at Radix Software and Engineering, and a Ph.D. researcher at LAAS-CNRS, France, since 2019. I work for the European project ETN-SAS in the domain of autonomous vehicles (AV). Before it, I graduated in Computer Science at Universidade Federal Rural do Rio de Janeiro (2014), and obtained a master's degree in Systems Engineering and Computing at Universidade Federal do Rio de Janeiro (2018), working with machine learning (ML) applied to dynamic environments. Currently, I apply several techniques (statistics, ML, and software engineering) to ensure safe decisions of ML-based perception tasks of AV at runtime.

(2) **BRUNO MONTEIRO ACCIOLI DA SILVA**
Bruno Monteiro Accioli da Silva, nasceu no Rio de Janeiro em 1995, graduou-se em Engenharia Eletrônica no CEFET-RJ e está cursando mestrado em Engenharia de Sistemas e da Computação, na UFRJ. Atua em projetos de pesquisa e desenvolvimento para o setor elétrico desde 2018 e atualmente trabalha um grupo voltado à formatação e execução de novos P&Ds no contexto da ANEEL. Seus principais interesses consistem no estudo da aplicação de técnicas de Inteligência Computacional em problemas no setor elétrico como um todo.

(3) **LUCAS MELO DE CHIARA**
Lucas Melo De Chiara, nasceu em São Paulo/SP em 1989, Graduou-se em Engenharia Elétrica na Universidade Estadual de Campinas (UNICAMP). Atua em projetos de pesquisa e desenvolvimento para o setor elétrico desde 2020 e atualmente gerencia um grupo voltado à formatação e execução de projetos de P&D no âmbito ANEEL. Seus principais interesses consistem no estudo da aplicação novas tecnologias e metodologias de gestão e execução de pesquisas para o desenvolvimento de melhorias e solução de problemas no setor elétrico como um todo.

(4) **JOAO PEDRO FERREIRA DOS SANTOS FREITAS**
JOÃO PEDRO FERREIRA DOS SANTOS FREITAS, nasceu em RESPLENDOR em 1995, graduou-se em ENGENHARIA ELÉTRICA na UNIVERSIDADE FEDERAL DE ITAUBÁ. Atua em projetos de pesquisa e desenvolve projetos de pesquisa e desenvolvimento para o setor elétrico desde 2018 e atualmente gerencia grupos para prospecção, formatação e execução de novos P&Ds regulados pela Aneel.

(5) **HUGO DINIZ REBELO**
Hugo Diniz Rebelo, nasceu no Rio de Janeiro em 1991, Graduou-se em Ciência da Computação na Universidade Federal Rural do Rio de Janeiro (UFRRJ) e finalizou mestrado em Engenharia de Sistemas e Computação, na COPPE-UFRJ. Atua em projetos de pesquisa e desenvolvimento para o setor elétrico desde 2019 e atualmente é analista de Dados em projetos de P&D ANEEL. Seus principais interesses consistem no estudo de técnica de de aprendizado de Máquina e suas aplicações em grande conjuntos de dados (big data).

(6) **JULIA DELGADO**
Julia Delgado, nasceu em Limeira-SP em 1994, Graduou-se em Análise e Desenvolvimento de sistemas na Metrocamp e está cursando um MBA em Analytics e Big Data, na FGV. Atua em projetos de pesquisa e desenvolvimento para o setor elétrico desde 2019 e atualmente atua como Product Owner em projetos de P&D no contexto da ANEEL. Seus principais interesses consistem no estudo da aplicação novas tecnologias e metodologias de gestão e execução de pesquisas para o desenvolvimento de melhorias e solução de problemas no setor elétrico como um todo.

(7) TASSIO SIMIONI
Engenheiro Químico pela UFSC, com especialização em processamento de gás natural pela UFRJ e Mestrado em Planejamento Energético pela COPPE-UFRJ com ênfase em tecnologia solar fotovoltaica. Atualmente é Gerente Geral Energia e Gás na RADIX, área que atende todo o setor elétrico e midstream de gás natural globalmente. Possui 10 anos de experiência em projetos de engenharia e tecnologia digital para múltiplas indústrias.

(8) ADONIS MARCEL DE ARAÚJO CARVALHO
ADONIS MARCEL DE ARAÚJO CARVALHO, nasceu em ALAGOINHAS-BA em 1986, graduou-se em ENGENHARIA QUÍMICA na UNIVERSIDADE FEDERAL DA BAHIA. Atua em projetos de pesquisa e desenvolvimento para o setor elétrico desde 2018, gerenciando a execução de projetos P&D no contexto da ANEEL. Seus principais interesses consistem no estudo da aplicação novas tecnologias e metodologias de gestão e execução de pesquisas para o desenvolvimento de melhorias e solução de problemas no setor elétrico como um todo.