Modelo de Detecção de Fraudes Elétricas Baseado em Aprendizado de Máquina

Geam W. Pfeiff[1], **Felipe Araújo**[1], **Helder Oliveira**[1], **Denis Rosário**[1], **Eduardo Cerqueira**[1]

[1]Instituto de Tecnologia - Universidade Federal do Pará (UFPA)
Caixa Postal 1 – 66.075-100 – Belém – PA – Brazil

geam.pfeiff@itec.ufpa.br

{felipearaujo, heldermay, denis, cerqueira}@ufpa.br

Abstract. *Non-technical losses, in most cases caused by fraud, are the leading cause of the financial losses of electricity utilities. These losses dramatically decrease the quality of the electrical networks, increasing the chances of blackouts, short circuits, and equipment failures. Thus, it is strategic to develop models that can detect non-technical losses. This article presents an Electrical Fraud Detector Based on Machine Learning (EFDBML), which classifies users as fraudulent or honest based on electrical consumption patterns and stochastic features. EFDBML obtained a detection rate of 98.02% and a false positive rate of 2.47% after selecting the machine learning algorithm with the best performance.*

Resumo. *As perdas não técnicas, na maioria dos casos ocasionadas por fraudes, são as principais causadoras dos prejuízos financeiros das concessionárias de energia elétrica. Essas perdas diminuem drasticamente a qualidade das redes elétricas, aumentando as chances de ocorrer blecautes, curtos circuitos e avarias de equipamentos. Desta forma, torna-se estratégico o desenvolvimento de modelos que consigam detectar perdas não técnicas. Este artigo apresenta um Detector de Fraudes Elétricas Baseado em Aprendizado de Máquina (DFEBAM), o qual classifica usuários em fraudulentos ou honestos baseados nos padrões de consumo elétrico e em features estocásticas. O DFEBAM obteve uma taxa de detecção de 98,02% e uma taxa de falso positivo de 2,47% após selecionar o algoritmo de aprendizado de máquina com a melhor desempenho.*

1. Introdução

Com a revolução da indústria 4.0, as redes elétricas estão passando por constantes mudanças. Um relatório da Agência Internacional de Energia (AIE) afirma que a implantação de Redes Elétricas Inteligentes (REI) é crucial para alcançar um futuro energético mais seguro e sustentável. O surgimento dos medidores inteligentes trouxera grandes benefícios para a automatização das redes convencionais, porém introduziram novas possibilidades de fraudes elétricas. Pesquisas realizadas pela Agência Nacional de Energia Elétrica (ANEEL) afirmam que os prejuízos ocasionados em 2018 por perdas elétricas não mensuradas foram de 6,6 bilhões de reais só no Brasil [ANEEL 2019], isso equivale a 14% do mercado consumidor.

As perdas elétricas são definidas através da diferença entre a energia gerada e a energia faturada, essas perdas são classificadas em dois tipos: perdas técnicas (do inglês *technical loss (TLs)*) e perdas não técnicas (do inglês *non-technical loss (NTLs)*). O primeiro caso ocorre na transmissão e distribuição, pelos próprios meios utilizados para fornecer o serviço. Equipamentos como transformadores, medidores e a própria fiação causam perdas, porém essas são mensuradas pelas concessionárias de energia elétrica. NTLs ou também chamadas de perdas não mensuradas, consistem na relação entre a energia fornecida ao usuário e a energia consumida que não é faturada [Ramos et al. 2018]. Um dos grandes desafios vinculados a REI é a identificação de NTLs, onde na maioria dos casos são ocasionadas por fraudes elétricas [Zheng et al. 2017].

Atualmente as pesquisas voltadas para a detecção de NTLs são divididas em três grandes áreas: métodos orientados a dados, orientados a rede e híbridos [Messinis and Hatziargyriou 2018a]. Métodos orientados a dados utilizam técnicas de aprendizado de máquina (do inglês *Machine Learning (ML)*) para a identificação de padrões de consumo dos usuários [Jokar et al. 2015]. Por outro lado, métodos orientados a rede necessitam de informações como topologia da rede, leituras de sensores instalados pela rede, leituras dos medidores inteligentes, perdas técnicas ocasionadas por equipamentos e meio de transporte entre outros dados, para modelar a rede de distribuição [Han and Xiao 2017]. Por fim, métodos híbridos consistem na combinação das outras duas técnicas [Messinis et al. 2019]. Nesse contexto, os métodos orientados aos dados são promissores devido a necessidade de uma menor variedade de informações e um menor gasto com infraestruturas, como a instalação de sensores e equipamentos extras na distribuição [Messinis and Hatziargyriou 2018a].

Métodos orientados a dados são subdivididos em supervisionados e não supervisionados dependendo do tipo de dados disponíveis e da natureza do algoritmo de ML escolhido. Algoritimos supervisionados como *Support Vector Machines (SVM)*, *Decision Tree (DT)*, *Random Forest (RF)*, *Gradient Boosted Machine (GBM)*, tem sido vastamente utilizados para classificar usuários que cometem NTLs [Buzau et al. 2018]. Algoritmos não supervisionados também tem sido utilizado devido não precisarem de dados rotulados, porém apresentam resultados de acurácia inferiores comparados aos supervisionados [Messinis and Hatziargyriou 2018b]. Estudos apontam que muitos algoritmos de MLs supervisionados tendem a melhorar suas métricas ao adicionar novas *features* estocásticas geradas a partir dos dados originais [Heaton 2016].

Este artigo propõem um Detector de Fraudes Elétricas Baseado em Aprendizado de Máquina (DFEBAM), o qual consiste em um método de detecção de NTL orientado aos dados. O DFEBAM tem objetivo de diferenciar padrões de consumo de usuários benignos e fraudulentos. Obteve-se uma análise comparativa entre o DFEBAM e métodos existentes na literatura, utilizou-se dados coletados de um cenário real na Irlanda. O DFEBAM mostrou significativas melhoras na taxa de detecção (*Detection Rate - DR)* e taxa de falsos positivos (*False Positive Rate - FPR)* comparada a trabalhos semelhantes, essas são as principais métricas usadas para avaliar os modelos que tratam desse problema.

O artigo está estruturado da seguinte forma, Seção 2 é abordado os trabalhos relacionados a detecção de NTL. Na Seção 3 é apresentado o DFEBAM, onde será analisado e discutido os resultados encontrados na Seção 4, e na seção 5 será introduzido a conclusão.

2. Trabalhos Relacionados

Nesta seção apresentam-se os trabalhos mais relevantes para essa pesquisa, a nível de embasamento teórico e comparação com a proposta. As pesquisas mapearam o estado da arte da computação ubíqua aplica para a identificação de NTL em REI.

Jokar *et al.* propuseram um método de detecção de roubo de energia baseado em padrões de consumo (*Consumption pattern-based energy theft detector - CPBETD*) [Jokar et al. 2015]. O método proposto é baseado em um SVM e consiste em analisar as amostras de cada usuário e classificá-las em fraudulentas ou não. Embora a proposta tenha demonstrado resultados satisfatórios, os autores não testaram sua metodologia em outros algoritmos de ML, os quais poderiam ter demonstrado resultados melhores.

Jindal *et al.* propuseram um esquema para identificação de NTL em todos os níveis da transmissão e distribuição da rede [Jindal et al. 2016]. A identificação a nível de distribuição consiste em um algoritmo baseado em DT combinado com um SVM. O SVM atua como classificador dos usuários, identificando padrões nos dados de consumo. Por outro lado, o DT estipula o consumo esperado de cada usuário baseado em dados como temperatura, números de moradores na casa, número de eletrodomésticos, horário do dia, estação do ano etc. Apesar de demonstrar uma melhora na precisão do classificador ao adicionar a saída da DT como uma entrada para o SVM, questões de privacidades dos usuários acabam sendo comprometidas.

Heaton fez um estudo analisando o desempenho de alguns modelos de ML ao adicionar novas *features* estocásticos gerados a partir dos dados originais [Heaton 2016]. Todos os classificadores testados apresentaram uma melhora significativa na acurácia ao utilizarem essas novas features aos algoritmos. Com base nisso, o DFEBAM utilizou novas entradas estocásticas como mínimo, máximo, média, desvio padrão, mediana, somatória e variância para melhorar ainda mais os resultados do modelo.

Punmiya and Choe propuseram um detector de roubo de energia baseado em aumento de gradiente (*Gradient Boosting Theft Detector - GBTD*) [Punmiya and Choe 2019]. O detector demonstrou significativas melhoras em termo de DR e FPR comparado a métodos relacionados. Contudo, os autores não fizeram uso de nenhum método para a escolha dos hiper parâmetros para os classificadores e não utilizaram técnicas de validação cruzada para a divisão dos dados.

Com base na análise dos trabalhos relacionados pode-se concluir que técnicas para melhorar os resultados dos modelos são válidas desde que preservem questões de privacidade dos clientes. Outro ponto importante a se ressaltar é fazer a análise dos métodos com diferentes algoritmos de ML e com diferentes hiper parâmetros para os classificadores, isso garante resultados melhores e mais confiáveis.

3. DFEBAM

Nessa seção apresenta-se o DFEBAM. A Figura 1 apresenta uma visão geral do método proposto, o qual utiliza como entrada os dados de consumo elétrico dos usuários e disponibiliza uma classificação de cada amostra em honesta ou fraudulenta. Para isso, o método consiste de três etapas principais: (1) geração dos dados fraudulentos; (2) tratamento dos dados; e (3) seleção do algoritmo de ML. Essas etapas serão explicadas a seguir.

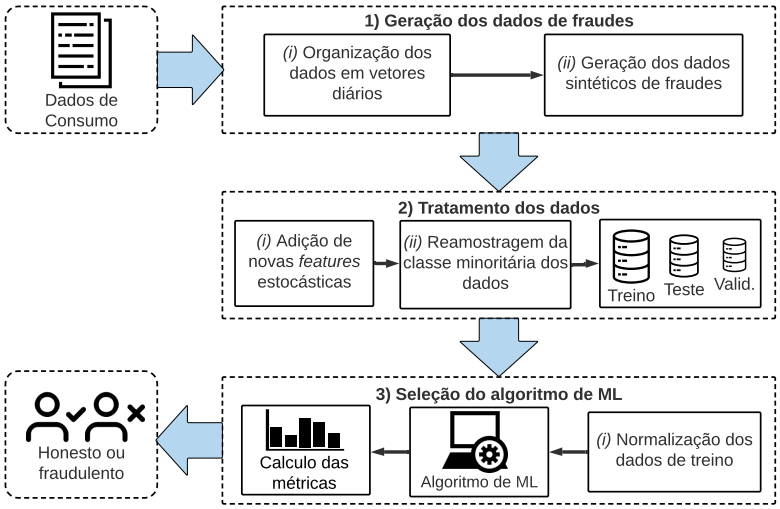

Figura 1. Visão geral do DFEBAM

3.1. Base de dados

Para este trabalho foi considerado uma base de dados amplamente utilizada para cenários de REI, a *Irish smart energy dataset* [CER 2012]. A base de dados é constituída por 4710 clientes, contendo usuários residenciais e comerciais, coletada na Irlanda no período de 2009 a 2010, somando um total de 520 dias. As leituras de consumo de energia foram feitas por medidores inteligentes a cada meia hora, onde a primeira leitura corresponde ao intervalo de 0h0min0s às 0h29min59s, ou seja, cada dia contém 48 leituras distribuídas sequencialmente. A base de dados foi obtida em um cenário constantemente monitorado no qual os participantes aceitaram os termos de compromisso e passaram por questionários antes e depois da coleta, portanto assumiu-se que a base de dados é constituída apenas de dados reais, não contendo dados de usuários fraudulentos.

3.2. Geração dos dados de fraudes

Após o carregamento do arquivo de dados, é necessário fazer uma reorganização das leituras de consumo com o objetivo de padronizar diferentes granularidades ao método. Os dados elétricos são reagrupados em vetores diários, onde cada vetor contém o número de leituras diárias. Finalizado o processo de reorganização, é necessário gerar os dados sintéticos de fraude (Eqs. 1-6), o qual foi proposto por [Jokar et al. 2015] e modificado por [Punmiya and Choe 2019]. Os dados sintéticos são gerados para toda a série temporal, desde a primeira leitura até a ultima, onde x é um vetor com as leituras reais de consumo diário de cada usuário, tal que $x = \{x_1, ..., x_{48}\}$, *random()* é uma função que gera um número aleatório no intervalo estipulado, $randint(0, 1)$ uma função que retorna os valores zero ou um, $mean()$ é uma função que retorna a média dos valores passados e t é o número da leitura, tal que $t \in [1, 48]$.

$$f1(x_t) = x_t \times \alpha, \ (onde \ \alpha = random(0.1, 0.9)) \tag{1}$$

$$f2(x_t) = x_t \times \beta_t, \ (onde \ \beta_t = random(0.1, 1.0)) \tag{2}$$

$$f3(x_t) = x_t \times \gamma_t, \ (onde \ \gamma_t = randint(0, 1)) \tag{3}$$

$$f4(x_t) = mean(x) \times \theta_t,\ (onde\ \theta_t = random(0.1, 1.0)) \tag{4}$$

$$f5(x_t) = mean(x) \tag{5}$$

$$f6(x_t) = x_{48-t} \tag{6}$$

A Eq. 1 representa uma fraude na qual o usuário diminui seu consumo diário de forma contínua, tendo sempre a mesma proporção entre leituras do mesmo dia. A Eq. 2 demonstra o comportamento de um cliente que desvia diferentes quantidades de energia durante o dia, hora podendo ser maior, hora menor. A Eq. 3 representa um caso de fraude no qual há uma interrupção ou uma anulação nas leituras de consumo em determinadas horas do dia. A Eq.4 demonstra a diminuição com base no consumo médio diário. As Eqs. 5 e 6 não alteram a somatória de consumo diário, porém *f5* envia sempre a média dos valores diários e *f6* muda a ordem das leituras. Esses dois últimos casos de fraudes são voltados para cenários onde o valor da fatura varia de acordo com o horário do dia.

3.3. Tratamento dos dados

Como demonstrado em [Heaton 2016], a adição de novas *features* estocásticas pode melhorar o desempenho de alguns classificadores. Para os modelos avaliados, os novos recursos foram gerados a partir dos vetores diários, tanto dos dados benignos quanto dos dados fraudulentos, e foram adicionadas como novas entradas, ou seja, o modelo sem *features* estocásticas tem 48 entradas e o modelo com todas as novas *features* tem 55 entradas. As novas entradas utilizadas foram a Média, Mediana, o Valor máximo, o Valor mínimo e o Desvio padrão gerados a partir das 48 amostras. Porém, ao analisar os dados, observou-se que as leituras correspondentes ao intervalo das 19:00 horas às 23:00 horas, eram mais relevantes, então gerou-se o somatório e a variância desse intervalo também, totalizando 7 novas entradas.

Com a agregação dos dados sintéticos de fraudes, é gerado um desequilíbrio de classes onde os dados fraudulentos tem um volume seis vezes maior do que os dados benignos. Bases de dados desbalanceadas podem gerar modelos ruins quando aplicados na prática mesmo apontando métricas satisfatórias, para evitar esse problema utilizou-se uma Técnica de super-amostragem, a *Synthetic Minority Over-sampling Technique* (SMOTE), a qual cria novas amostras sintéticas baseada nas amostras reais da classe minoritária, isso possibilita equilibrar a quantidade de dados por classes.

Após a super-amostragem dos dados, utilizou-se uma técnica de validação cruzada bloqueada para séries temporais para separar os dados em treino teste e validação, o *Blocking Time Series Split*. Foi estipulado 10 interações para a validação cruzada, ou seja, 10 combinações de treino, teste e validação, somando 52 dias a cada interação. Em cada conjunto os 39 (75%) primeiros dias foram separados para treino e os últimos 13 (25%) para teste. Dentre os conjuntos de treino, foram separados os últimos 9 dias (aproximadamente 17%) para a validação, será útil na escolha do melhor algoritmo de classificação e seleção das *features* estocásticas.

O *Blocking Time Series Split* adiciona margens em duas direções. Uma margem impede que o modelo memorize tendências futuras. A outra impede que o modelo rememorize padrões entre as interações. Apesar do aumento da complexidade ao usar uma técnica de validação cruzada, essa é indispensável devido tornar os modelos mais robusto a erros e gerar resultados mais confiáveis. Levar em consideração séries temporais no *Split*

dos dados é essencial para essa aplicação, isso garante que o modelo aprenda (treine) com dados passados e faça predições (teste) dos dados futuros.

3.4. Seleção do algoritmo de ML

Após o *Split* dos dados, os conjuntos de treinamento passam por uma normalização. A Normalização tem como objetivo transformar todas as variáveis em uma mesma ordem, o método proposto utilizou um intervalo de 0 à 1. Entradas normalizadas impedem que os algoritmos de ML fiquem inclinados às variáveis com maior ordem de grandeza, possibilitando a melhor generalização do modelo. Após a normalização, os algoritmos de ML passam pelo processo de treinamento no conjunto de dados designados para isso. Todos os algoritmos de ML implementados (SVM, XGBoost, CatBoost, LightBoost, RF) são classificadores supervisionados, ou seja, necessitam de dados rotulados. Com o intuito de selecionar o classificador para o método proposto, as métricas de avaliação são calculadas utilizando o conjunto de validação. Para a avaliação final do Modelo, utiliza-se o conjunto de teste para o cálculo das métricas.

Para um melhor desempenho dos algoritmos implementados, utilizou-se de uma técnica de pesquisa de grade (do inglês *Grid Search*). A pesquisa de grade tem como objetivo treinar os algoritmos com diferentes hiper parâmetros, posteriormente comparálos e selecionar os que tiveram os melhores resultados para o modelo final. Os hiper parâmetros usados para cada algoritmos são apresentados na Tabela 1.

Tabela 1. Hiper parâmetros dos algoritmos

Algoritmos	Hiperparâmetros
SVM	Kernel = 'rbf' C = [1, 10, 100, 1000] gamma = [0.1, 0.01, 0.001, 0.0001]
XGBoost	n_estimators = [50, 100, 150] max_depth = None (sem limites) gamma = [0.1, 0.01, 0.001] learning_rate = [0.3, 0.03, 0.003]
CatBoost	n_estimators = [50, 100, 150] max_depth = None (sem limites) learning_rate = [0.3, 0.03, 0.003]
LightBoost	boosting_type = 'gbdt' n_estimators = [50, 100, 150] max_depth = None (sem limites) learning_rate = [0.3, 0.03, 0.003]
RF	criterion = ['gini', 'entropy'] n_estimators = [50, 100, 150] max_depth = None (sem limites)

As métricas de avaliação são calculadas com base em uma matriz de confusão. A matriz de confusão é obtida a partir da comparação entre os valores preditos pelo modelo e os valores reais. Consiste em uma tabela contendo verdadeiro positivo (VP), falso positivo (FP), verdadeiro negativo (VN) e falso negativo (FN).

As métricas mais utilizadas para avaliar Detectores de fraudes elétricas são o *Recall* ou *DR* (Eq. 7) e a *FPR* (Eq. 8). A *DR* consiste na proporção de positivos reais que foram identificados corretamente, ou seja, a identificação correta de indivíduos que cometeram fraudes. A *FPR* avalia a taxa de falsos positivos, ou seja, indivíduos classificados como fraudulentos que não cometeram fraudes. Para essa aplicação a *FPR* é proporcional a um custo adicional a concessionária, haja vista a necessidade da designação de uma equipe técnica de inspeção aos usuários classificados erradamente. Outro ponto a se ressaltar é o constrangimento causado aos clientes que passam pela vistoria sem terem cometido fraudes.

$$Recall\ (DR) = \frac{VP}{VP + FN} \tag{7}$$

$$FPR = \frac{FP}{FP + VN} \tag{8}$$

Outras métricas usadas para a avaliação foram a *Acurácia* que avalia a quantidade de previsões corretas dentre todas as previsões feitas (Eq. 9). A *Precisão* que avalia a exatidão, ou seja, a quantidade de indivíduos classificados como fraudulentos que realmente cometeram fraudes (Eq. 10). O *F1-Score* que consiste na média ponderada da *Precisão* e do *Recall*, tem grande uso para avaliar classificadores binários, como é o caso em questão (Eq. 11). E por último a *AUC* que consiste na área sobre a curva *ROC*, é uma excelente métrica para analisar bases de dados desbalanceadas (Eq. 12).

$$Acurácia = \frac{VP + VN}{VP + FP + FN + VN} \tag{9}$$

$$Precisão = \frac{VP}{VP + FP} \tag{10}$$

$$F1 - Score = 2 * \frac{Recall * Precisão}{Recall + Precisão} \tag{11}$$

$$AUC = \int_{x=0}^{1} Recall(FPR^{-1}(x))dx \tag{12}$$

4. Resultados

Nessa seção apresentam-se os resultados obtidos. Devido ser criado um modelo para cada usuário, os resultados apresentados serão a média de todos os usuários. Três pontos serão abordados: A comparação qualitativa dos diferentes algoritmos de ML implementados no DFEBAM; os resultados experimentais do DFEBAM ao adicionar novas *features* estocásticas e a comparação do DFEBAM a outros trabalhos do estado da arte.

4.1. Avaliação entre diferentes algoritmos de classificação para o DFEBAM

Para a seleção do melhor classificador para o DFEBAM, utilizou-se da metodologia explanada na seção 3. Para uma avaliação justa entre os algoritmos, utilizou-se os conjuntos de treino e validação para treinar e calcular as métricas respectivamente. A validação é utilizada para a escolha do melhor algoritmo de ML para o modelo, o classificador com os melhores resultado será o algoritmo base para o modelo final.

A Tabela 2 expõem os resultados dos algoritmos de classificação, comparando todas as métricas apresentadas na Sessão 3.4. Pode-se observar que o RF teve os melhores resultados em todas as métricas comparadas, apresentando uma melhora significativa em DR que chegou a 97,53% e em FPR que diminuiu para 2,96%, sendo essas as métricas mais relevantes para o DFEBAM. Com base nos experimentos, comprovou-se a superioridade do RF em relação aos outros algoritmos, logo esse foi escolhido como classificador para o DFEBAM.

Tabela 2. Comparação entre algoritmos de classificação

Algoritmos	Acurácia(%)	DR(%)	F1-Score(%)	FPR(%)	Precisão(%)	AUC(%)
SVM	88,85	95,24	93,55	5,25	92,29	72,87
XGBoost	93,43	95,96	96,10	4,53	96,42	87,12
CatBoost	94,32	96,64	96,46	3,85	96,41	87,28
LightBoost	94,04	96,85	96,48	3,64	96,26	87,01
RF	94,83	97,53	96,95	2,96	96,52	88,07

4.2. Avaliação das novas *features* estocásticas adicionadas ao DFEBAM

Para a avaliação das novas *features* estocásticas, obteve-se o resultado do DFEBAM utilizando o RF como classificador devido sua superioridade como explanado na Seção 4.1. Utilizou-se dos conjuntos de treino e validação para treinar e calcular as métricas simultaneamente. A Tabela 3 expõem os resultados ao adicionar cada *feature* separadamente e todas as setes juntas. Observou-se que os valores de DR chegaram a 98.21% e de FPR a 2.28% ao adicionar todas as *features*. Isso se explica devido as novas entradas ficarem entre as mais relevantes para o RF, onde entradas com alta relevância ajudam diretamente na classificação.

Tabela 3. Resultado com novas features

Modelos	Acurácia(%)	DR(%)	F1-Score(%)	FPR(%)	Precisão(%)	AUC(%)
RF	94,83	97,53	96,95	2,96	96,52	88,07
$RF_{Média}$	94,86	97,57	96,96	3,02	96,61	88,34
$RF_{Mediana}$	94,83	97,55	96,99	3,01	96,56	88,19
RF_{Max}	94,97	97,56	97,03	2,93	96,35	88,50
RF_{STD}	94,98	97,57	97,04	2,92	96,66	88,53
RF_{Min}	95,20	97,68	97,17	2,81	96,79	89,00
RF_{SomaP}	95,71	97,98	97,46	2,51	97,06	90,02
RF_{VarP}	95,86	98,12	97,55	2,37	97,10	90,21
RF_{Todas}	96,27	98,21	97,78	2,28	97,47	91,42

Classificadores RF possuem um modulo de *Feature Importance*, esse disponibiliza um ranque das *features* mais relevantes para o modelo. Notou-se que as 7 novas entradas ficaram entre as 13 mais relevantes dentre as 55. Com base no ranque, utilizou-se um processo de *Feature Selection*, o qual extrai as entradas de menor relevância para as de maior relevância. A Figura 2 apresenta os resultados de DR e FPR em relação ao número

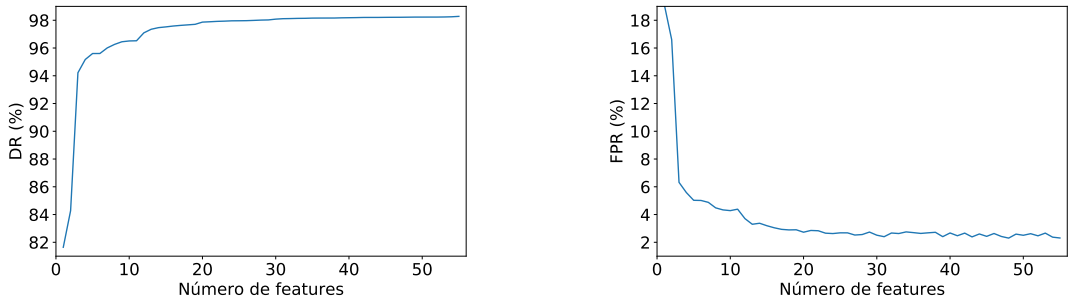

Figura 2. Performance da DR e da FPR em relação ao número de *features*

de entradas usadas para treinar o modelo. Observou-se que os valores entre as *features* 15 a 55 mantém uma certa constância, não oferecendo grandes melhorias nos resultados.

A complexidade do tempo de treinamento de um modelo RF é dada pela Eq. 13, onde n é o número de amostras, d é o número de entradas e k os números de árvores. Ou seja, reduzir o número de *features* reduz diretamente a complexidade de tempo de treinamento. Utilizar o modulo de *Feature Selection* possibilita estipular um número de entradas para diminuir a complexidade de tempo e recursos gastos, porém mantendo resultados satisfatórios. Logo, usuários que buscam uma menor complexidade de tempo, a sugestão seria utilizar apenas as 15 *features* mais relevantes.

$$Complexidade\ do\ tempo\ de\ treinamento = O(n * log(n) * d * k) \tag{13}$$

4.3. Avaliação do DFEBAM comparado a outros modelos

Com a escolha do RF como classificador base e a adição das novas *features*, o DFEBAM foi treinado e as métricas foram calculadas usando o conjunto de teste. Buscando uma comparação do desempenho entre métodos, utilizou-se todas as entradas para calcular a DR e a FPR que resultaram em 98,02% e 2,47% respectivamente. Melhorias nessas métricas implicam em uma maior taxa de acerto e um menor custo para as concessionárias como explicado na Subseção 3.4. Na Tabela 4 pode-se observar que o DFEBAM apresenta os melhores valores de DR e FPR, além de disponibilizar o modulo de *Feature Selection* e ser o único a considerar séries temporais no *Split* dos dados, tornando-o mais robusto a erros e mais próximo a cenários reais como explicado na Subseção 3.3.

Tabela 4. Comparação entre modelos

Modelos	DR (%)	FPR (%)	*Feature Selection*	Considera séries temporais
CPBETD [Jokar et al. 2015]	94,00	11,00	Não	Não
GBTD [Punmiya and Choe 2019]	97,00	3,00	Sim	Não
DFEBAM	98,02	2,47	Sim	Sim

5. Conclusão

Este trabalho apresentou um modelo para a detecção de fraudes elétricas em REI. O DFEBAM aprende padrões de consumo dos usuários e posteriormente classifica-os em honestos ou fraudulentos. Diferentes algoritmos de ML foram testados e comparados trazendo

contribuições experimentais. A proposta também foi comparada com outros detectores de fraude e demonstrou resultados superiores. Dentre os algoritmos de ML testados, o RF foi o que obteve os melhores resultados. Com o objetivo de melhorar ainda mais os resultados do DFEBAM, adicionou-se novas *features* estocásticas. As novas entradas resultaram em uma DR de 98,02% e uma FPR de 2,47% quando testadas no modelo final, o que são resultados excelentes para detectores de fraudes elétricas. Outro ponto importante levado em consideração foi o tratamento dos dados levando em consideração séries temporais, o que condiz com aplicações reais.

Referências

ANEEL (2019). Perdas de energia elétrica na distribuição. http://www.encurtador.com.br/imrIS, Acesso: 20/1/2020.

Buzau, M. M., Tejedor-Aguilera, J., Cruz-Romero, P., and Gómez-Expósito, A. (2018). Detection of non-technical losses using smart meter data and supervised learning. *IEEE Transactions on Smart Grid*, 10(3):2661–2670.

CER (2012). *Irish Social Science Data Archive*. http://www.ucd.ie/issda/data/commissionforenergyregulationcer/, Acesso: 7/8/2019.

Han, W. and Xiao, Y. (2017). Nfd: Non-technical loss fraud detection in smart grid. *Computers & Security*, 65:187–201.

Heaton, J. (2016). An empirical analysis of feature engineering for predictive modeling. In *SoutheastCon 2016*, pages 1–6. IEEE.

Jindal, A., Dua, A., Kaur, K., Singh, M., Kumar, N., and Mishra, S. (2016). Decision tree and svm-based data analytics for theft detection in smart grid. *IEEE Transactions on Industrial Informatics*, 12(3):1005–1016.

Jokar, P., Arianpoo, N., and Leung, V. C. (2015). Electricity theft detection in ami using customers' consumption patterns. *IEEE Transactions on Smart Grid*, 7(1):216–226.

Messinis, G. M. and Hatziargyriou, N. D. (2018a). Review of non-technical loss detection methods. *Electric Power Systems Research*, 158:250–266.

Messinis, G. M. and Hatziargyriou, N. D. (2018b). Unsupervised classification for non-technical loss detection. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE.

Messinis, G. M., Rigas, A. E., and Hatziargyriou, N. D. (2019). A hybrid method for non-technical loss detection in smart distribution grids. *IEEE Transactions on Smart Grid*, 10(6):6080–6091.

Punmiya, R. and Choe, S. (2019). Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Transactions on Smart Grid*, 10(2):2326–2329.

Ramos, C. C., Rodrigues, D., de Souza, A. N., and Papa, J. P. (2018). On the study of commercial losses in brazil: a binary black hole algorithm for theft characterization. *IEEE Transactions on Smart Grid*, 9(2):676–683.

Zheng, Z., Yang, Y., Niu, X., Dai, H.-N., and Zhou, Y. (2017). Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. *IEEE Transactions on Industrial Informatics*, 14(4):1606–1615.