UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LAURIVAN SARETA

IDENTIFICAÇÃO DE PERDA NÃO TÉCNICA DE ENERGIA ELÉTRICA

CHAPECÓ
2019

LAURIVAN SARETA

IDENTIFICAÇÃO DE PERDA NÃO TÉCNICA DE ENERGIA ELÉTRICA

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Guilherme Dal Bianco

CHAPECÓ
2019

Bibliotecas da Universidade Federal da Fronteira Sul - UFFS

Sareta, Laurivan
Identificação de perda não técnica de energia elétrica. / Laurivan Sareta. -- 2019.
38 f.:il.

Orientador: Guilherme Dal Bianco.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal da Fronteira Sul, Curso de Ciência da Computação, Chapecó, SC , 2019.

1. Perda não técnica de energia elétrica. 2. Aprendizado de Máquina. I. Bianco, Guilherme Dal, orient. II. Universidade Federal da Fronteira Sul. III. Título.

Elaborada pelo sistema de Geração Automática de Ficha de Identificação da Obra pela UFFS com os dados fornecidos pelo(a) autor(a).

LAURIVAN SARETA

IDENTIFICAÇÃO DE PERDA NÃO TÉCNICA DE ENERGIA ELÉTRICA

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Guilherme Dal Bianco

Este trabalho de conclusão de curso foi defendido e aprovado pela banca em: 06/12/19

BANCA EXAMINADORA:

Dr. Guilherme Dal Bianco - UFFS

Dr. Denio Duarte - UFFS

Me. Andressa Sebben - UFFS

RESUMO

Devido às mudanças nos sistemas elétricos de vários países e o aumento da competitividade, as empresas distribuidoras de energia elétrica precisaram buscar melhorar o desempenho financeiro e técnico para obter maior produtividade, eficiência e lucratividade. Uma das maneiras de melhorar e maximizar a energia disponível é reduzindo as fraudes e roubos envolvendo o sistema de energia. Este percentual de energia que é entregue mas não é faturada devido a má fé dos consumidores é caracterizado como perda não técnica de energia elétrica. Para resolver esse problema, as operadoras dos sistemas de distribuição fazem inspeções nos consumidores. No entanto, é inviável inspecionar todos os clientes. Uma alternativa de menor custo é aplicação de algoritmos de aprendizado de máquina, como por exemplo, Floresta de Caminhos Ótimos, Máquina de vetores de suporte, Árvore de decisão, Detecção de *outliers* e Mapa de auto-organização. A intuição é obter a probabilidade do consumidor estar cometendo fraude. Atualmente os estudos para identificação de perda não técnica que utilizam métricas de avaliação diferentes e não há comparativos com algoritmos supervisionados tradicionais, tampouco com não supervisionados. Este trabalho propõe uma análise experimental dos algoritmos Floresta de Caminhos Ótimos, SVM, Árvore de decisão supervisionado e Floresta de Caminhos Ótimos não supervisionado. Desta forma, o objetivo é comparar os algoritmos para identificar as probabilidades de um consumidor estar cometendo fraude no sistema de energia elétrica, utilizando as mesmas métricas e base de dados de maneira a entender o comportamento dos algoritmos. Os experimentos demonstraram que é possível criar uma base de dados com rótulos sintéticos de fraudes, sendo que para estes experimentos o SVM se saiu superior aos algoritmos Floresta de Caminhos Ótimos e Árvore de decisão.

Palavras-chave: NTL; Perda não técnica de energia elétrica; Floresta de Caminhos Ótimos; OPF; SVM.

ABSTRACT

Due to changes in the electrical systems of many countries and increasing competitiveness, electric utilities have had to seek to improve financial and technical performance for greater productivity, efficiency and profitability. One of the ways to improve and maximize available energy is by reducing fraud and theft involving the power system. This percentage of energy that is delivered but not billed due to bad consumer faith is characterized as non-technical loss of electricity. To solve this problem, distribution system operators carry out inspections of consumers. However, it is impossible to inspect all customers. A lower cost alternative is the application of machine learning algorithms such as Optimal Path Forest, Support Vector Machine, Decision Tree, Outliers Detection, and Self-Organization Map. The intuition is to get the probability that the consumer is committing fraud. Currently, studies for non-technical loss identification using different evaluation metrics and no comparisons with traditional supervised or unsupervised algorithms. This paper proposes an experimental analysis of the optimal path forest, SVM, supervised decision tree and unsupervised optimal path algorithms. Thus, the objective is to compare the algorithms to identify the probabilities of a consumer committing fraud in the electricity system, using the same metrics and database in order to understand the behavior of the algorithms. The experiments demonstrated that it is possible to create a database with synthetic fraud labels, and for these experiments SVM was superior to the Optimal Path Forest and Decision Tree algorithms.

Keywords:

. OPF; Optimal Path Forest; non-technical loss identification; SVM;

LISTA DE FIGURAS

Figura 2.1 – Passo (A) do algoritmo Floresta de Caminhos Ótimos 12
Figura 2.2 – Passo (B) do algoritmo Floresta de Caminhos Ótimos 13
Figura 2.3 – Passo (C) do algoritmo Floresta de Caminhos Ótimos 13
Figura 2.4 – Passo (D) do algoritmo Floresta de Caminhos Ótimos 14
Figura 3.1 – BaseA Sub1 30
Figura 3.2 – BaseB Sub1 31
Figura 3.3 – BaseA Sub2 32
Figura 3.4 – BaseB Sub2 33
Figura 3.5 – BaseA Sub3 34
Figura 3.6 – BaseB Sub3 35

SUMÁRIO

1 INTRODUÇÃO 8
1.1 Objetivos 9
1.1.1 Objetivo Geral 9
1.1.2 Objetivos Específicos 9
1.2 Justificativa 9
2 TRABALHOS RELACIONADOS 11
2.1 Supervisionado 11
2.1.1 Floresta de Caminhos Ótimos (*Optimum-Path Forest* -OPF) 11
2.1.2 Máquina de vetores de suporte (*Support Vector Machines* - SVM) 16
2.1.3 Árvore de Decisão 18
2.2 Não supervisionado 19
2.2.1 Floresta de Caminhos Ótimos 19
2.2.1.1 Mapa de auto-organização (*Self Organizing Map* - SOM) 21
2.2.1.2 Detecção de *outliers* (*outlier detection*) 22
3 ANÁLISE EXPLORATÓRIA 24
3.1 Base de Dados 24
3.2 Pré processamento 24
3.3 Métricas e Configurações 26
3.4 Execução dos Experimentos 28
3.4.1 **Algoritmos Supervisionados** 28
3.4.2 **Algoritmo não-supervisionado** 31
4 CONCLUSÃO 36
4.1 Trabalhos Futuros 36
REFERÊNCIAS 37

1 INTRODUÇÃO

A energia elétrica tornou-se com, o passar dos anos, um elemento básico na vidas das pessoas e de grande importância para o desenvolvimento dos países [Lakshmi and Kumar, 2013]. Dessa forma, foi necessário que os sistemas elétricos de vários países sofressem mudanças. Uma dessas mudanças é a privatização das empresas de energia elétrica, fato que favoreceu o aumento da competitividade e, consequentemente demandou maiores investimentos, principalmente para melhorar o desempenho financeiro e técnico, visando maior produtividade, eficiência e lucratividade [Ramos et al., 2011]. Uma das maneiras de melhorar e maximizar a energia disponível é reduzindo as fraudes e roubos envolvendo o sistema de energia [Ramos et al., 2011]. Porém, mesmo que as operadoras dos sistemas de distribuição (OSD) aloquem esforços para detectar o roubo de eletricidade e a imposição de impedimentos legais, o fenômeno ainda ocorre [Messinis and Hatziargyriou, 2018]. Ainda é muito complexo calcular ou medir a extensão das perdas, pois é difícil determinar onde elas ocorrem [Ramos et al., 2011].

Estima-se que as empresas de serviços públicos no mundo todo perdem mais de US$ 25 bilhões por ano em virtude do roubo de eletricidade. Tal problema acarreta em vários efeitos negativos, como a sobrecarga da unidade de geração, resultando em sobretensão e podendo danificar aparelhos de clientes, como também pode desarmar a unidade de geração, interrompendo a alimentação de energia para todos os clientes. Tais fatores levam as concessionárias a repassar essas perdas aos clientes genuínos na forma de tarifas [Depuru et al., 2011].

Devido à inviabilidade de inspecionar todos os clientes, uma das formas é utilizados algoritmos de aprendizado de máquina (Floresta de Caminhos Ótimos [Ramos et al., 2011] [Júnior et al., 2016], Máquina de vetores de suporte [Nagi et al., 2010], Detecção de *outliers* [Messinis and Hatziargyriou, 2018], Mapa de auto-organização [Messinis and Hatziargyriou, 2018], Árvore de decisão [Cody et al., 2015]) para identificação de perda não técnica. Estes algoritmos podem ser orientados a dados ou à rede, dependendo de quais dados da rede elétrica **são** utilizados (por exemplo, topologia de rede ou medições de rede).

Os métodos orientados a dados utilizam apenas dados relacionados ao consumidor, como o consumo de energia, tipo de consumidor, etc. Da mesma forma, os métodos orientados a dados são divididos em supervisionados e não supervisionados.

Para [Monard and Baranauskas, 2003], no aprendizado supervisionado é provido ao algoritmo de aprendizado uma série de exemplos de treinamento para os quais o rótulo da classe

associada é conhecido. Já no aprendizado não supervisionado, verifica-se os exemplos fornecidos a fim de determinar se alguns deles podem ser de alguma forma agrupados, formando agrupamentos ou *clusters*. Após os agrupamentos serem definidos, é necessário realizar a análise e determinar o que cada agrupamento representa no contexto do problema que está sendo analisado.

Diante deste contexto, o presente trabalho tem como foco analisar experimentalmente técnicas e algoritmos de aprendizado de máquina para identificar as probabilidades de um consumidor estar cometendo fraude no sistema de energia elétrica.

1.1 Objetivos

1.1.1 Objetivo Geral

Analisar experimentalmente os algoritmos de aprendizado de máquina, com a finalidade de identificar consumidores que estejam cometendo fraude no sistema de energia elétrica.

1.1.2 Objetivos Específicos

- Analisar técnicas de identificação de perda não técnica;
- Analisar algoritmos OPF, SVM, Árvore de decisão já existentes que possam identificar a perda não técnica;
- Construir uma base de dados rotulada de consumidores de energia elétrica;
- Realizar experimentos com base nos algoritmos analisados com a finalidade de identificar técnicas relevantes.

1.2 Justificativa

Este trabalho se justifica pela sua oportunidade para o meio acadêmico, servindo como base para novos estudos em perda não técnica pois aprofunda a análise de métodos existentes na bibliografia, que foram demonstrados e validados a fim de identificar a sua eficiência em determinados casos.

Para as empresas distribuidoras de energia elétrica a oportunidade é apresentada na identificação das fraudes e consequentemente na redução das perdas não técnicas, minimizando as

atuais consequências financeiras e técnicas conforme expõe [Messinis and Hatziargyriou, 2018].

Este trabalho se justifica quanto à sua importância para a sociedade, pois a identificação das fraudes envolvendo a energia elétrica permite que a energia disponível seja maximizada. Com isso a sociedade é beneficiada, evitando sobrecargas na unidade de geração, sobretensão nas redes e evitando danificar aparelhos dos clientes. Se identificadas 10% das fraudes, cerca de 83.000 GWh de energia elétrica seriam conservados reduzindo a emissões de dióxido de carbono em 9,2 milhões de toneladas por ano [Depuru et al., 2011].

Quanto à viabilidade, esta se confirma pois atualmente as empresas distribuidoras de energia elétrica apresentam um gasto muito grande com verificação de fraudes, já que a forma mais eficiente é o deslocamento de técnicos para os locais onde a energia é utilizada, o que é inviável de se executar para todas as unidades consumidoras devido ao grande volume [Ramos et al., 2011].

2 TRABALHOS RELACIONADOS

O aprendizado de máquina é uma área da inteligência artificial que objetiva o desenvolvimento de técnicas computacionais de aprendizado e a elaboração de sistemas com a capacidade de adquirir conhecimento de forma automática, ou seja, um sistema de aprendizado é um programa de computador que consegue decidir baseando-se em sua experiências anteriores utilizadas para a solução bem sucedida de problemas ([Monard and Baranauskas, 2003]).

De acordo com [Batista et al., 2003], existem várias abordagens que podem ser utilizadas por um sistema computacional para adquirir aprendizado, como por exemplo, o aprendizado por hábito, por instrução, por dedução, por analogia e por indução. O aprendizado indutivo é um dos mais úteis pois permite a obtenção de novos conhecimentos através de exemplos ou casos previamente observados. Mas, o aprendizado indutivo é também um dos mais desafiadores, porque o conhecimento gerado ultrapassa os limites do que já se sabe, mesmo que não há garantia de que a verdade é preservada, [Monard and Baranauskas, 2003] complementa que mesmo assim, a inferência indutiva é um dos mais importantes métodos usados para derivar conhecimento novo e prever eventos futuros. O aprendizado indutivo pode ser dividido em supervisionado e não supervisionado.

Para [Monard and Baranauskas, 2003], o aprendizado supervisionado é provido ao algoritmo de aprendizado, ou indutor, uma série de exemplos de treinamento para os quais o rótulo da classe associada é conhecido. Já no aprendizado não supervisionado, o indutor verifica os exemplos fornecidos a fim de determinar se alguns deles podem ser de alguma forma agrupados, formando agrupamentos ou *clusters*. Após os agrupamentos serem definidos é necessário realizar a análise e determinar o que cada agrupamento representa no contexto do problema que está sendo analisado.

Nas sessões 2.1, 2.2 é apresentado as definições e exemplificados os dois métodos.

2.1 Supervisionado

2.1.1 Floresta de Caminhos Ótimos (*Optimum-Path Forest* -OPF)

No trabalho de [Ramos et al., 2011] é usado uma abordagem baseada no uso de floresta de caminhos ótimos (*Optimum-Path Forest*-OPF) para determinar se um usuário está se tornando um consumidor ilegal através de um classificador OPF, produzindo resultados rápidos

e precisos na identificação de perdas não técnicas.

Apesar do aumento no uso de técnicas de aprendizado de máquina, se torna crucial na escolha do algoritmo observar os pontos negativos e os problemas de cada um. Tendo isso em vista, o autor utiliza o cálculo de floresta de caminhos ótimos (*Optimum-Path Forest* - OPF) de uma forma diferente, reduzindo o problema do reconhecimento de padrões no espaço de característica induzido por aquele grafo.

Segundo [Ramos et al., 2011], as vantagens principais de um classificador baseado em OPF aplicado à perda técnica de energia se resume em ser livre de parâmetros e são viáveis para utilização em aplicações de tempo real para detecção de fraude, pois executam a fase de treinamento mais rápido.

O algoritmo OPF atribui um caminho ótimo de cada amostra, formando uma floresta de caminhos ótimos. As atribuições são feitas de tal forma que não forme ciclos. Inicialmente, devem ser encontrados os protótipos (amostras que melhor representam essas classes, ou seja os exemplos mais importantes) que estão dispostos na região entre as classes e que estas regiões geralmente são sobrepostas. Para medir a distância pode ser utilizada uma função custo-caminho (função euclidiana (descrita em [Ferreira, 2008]) ou algoritmo de distância mais elaborado) devido às suas propriedades teóricas para estimar protótipos que possuem este comportamento.

A seguir, apresenta-se um exemplo de uma iteração do algoritmo OPF e a sua aplicação. Seja Z uma base de dados com um conjunto de amostras de todas as classes. A distância entre duas amostras é dada pela distância entre seus vetores de características. Para medir esta distância pode ser utilizada qualquer métrica, este caso usa a distância Euclidiana. O OPF foi projetado usando uma função que computa a distância máxima entre as amostras adjacentes.

Seja G um grafo completo onde os nós são amostras em Z e todo qualquer par de amostra tem um arco entre si, conforme a figura 2.1 .

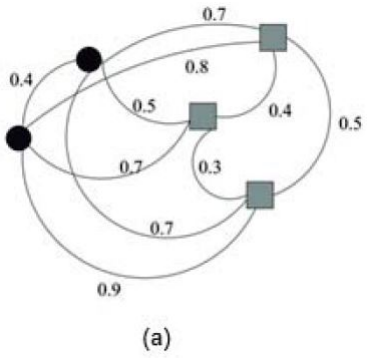

Figura 2.1: Passo (A) do algoritmo Floresta de Caminhos Ótimos

O Algoritmo OPF associa um caminho ótimo para todas as amostras, com isso formando uma floresta de caminhos ótimos. A figura 2.2 apresenta a floresta de caminhos ótimos resultante da 2.1 para os protótipos 1 e 2.

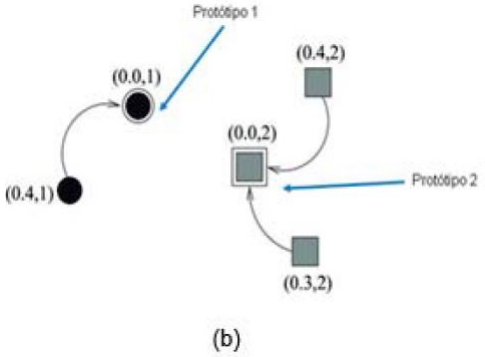

Figura 2.2: Passo (B) do algoritmo Floresta de Caminhos Ótimos

A entrada (x,y) dos nós representam o custo e o rótulo da amostra, já os arcos direcionados indicam os nós que os precedem no caminho ótimo. Na figura 2.3 o triângulo é a amostra de teste. Esta é a fase de classificação que vai ser calculado suas conexões (linhas tracejadas), a partir dos nós de treinamento.

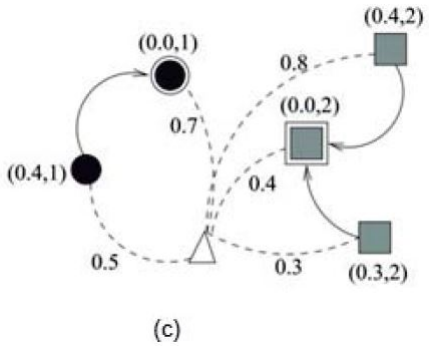

Figura 2.3: Passo (C) do algoritmo Floresta de Caminhos Ótimos

Os resultados da classificação se encontram na figura 2.4. O custo 0.3 de classificação vai ser associado a amostra de teste, juntamente com o rótulo 2, estes valores foram obtidos através do caminho ótimo do protótipo mais fortemente conexo.

Para realizar os experimentos [Ramos et al., 2011] utilizou dois conjuntos de dados: o primeiro composto por 5.190 perfis industriais e o segundo por 8.067 perfis comerciais, sendo que a rotulagem dos conjuntos de dados foi realizada por técnicos da empresa. Cada perfil é representado por quatro características:

- Demanda Contratada: o valor da demanda por disponibilidade contínua solicitada à empresa de energia que necessita ser paga independentemente de a energia elétrica ser ou

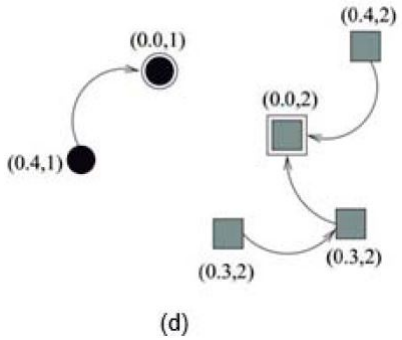

Figura 2.4: Passo (D) do algoritmo Floresta de Caminhos Ótimos

não utilizada pelo consumidor, medida em quilowatts (kW);

- Demanda Medida ou Máxima Demanda: a demanda real máxima de energia ativa é verificada por medição em intervalos de 15 min durante o período de faturamento, medida em quilowatts (kW);

- Fator de Carga: a relação entre a demanda média (DMédia) e a demanda máxima (DMax) da unidade consumidora, registrada no mesmo período de tempo. É um índice que mostra como a energia elétrica é utilizada de forma racional, sendo a relação entre a energia total e o período;

- Potência Instalada: é a soma da potência nominal de todos os equipamentos elétricos instalados e prontos para operar na unidade consumidora, em quilowatts (kW);

O autor compara redes neurais, SVM-RBF, SVM-LINEAR, ANN-MLP e SOM com o OPF para detecção de perdas não técnicas e avaliou sua robustez em relação às variações no conjunto de treinamento em dois conjuntos de dados distintos. Para isso autor utilizou dados de uma empresa brasileira de energia. Em todos os testes foi usado 50% da base para treino e 50% para teste, os resultados são média com desvio padrão de cada algoritmo, sendo que cada um foi executado dez vezes. Os experimentos usando redes neurais, OPF, SVM-RBF, SVM-LINEAR, ANN-MLP e SOM em conjuntos de dados industriais e comerciais demonstraram que essa abordagem pode superar o tempo de classificação das técnicas mais comumente usadas em termos de eficiência e eficácia, que pode ser observados nas tabelas 2.4, 2.2 e 2.3.

As métricas utilizadas por [Ramos et al., 2011] com intuito de avaliar o desempenho dos métodos de detecção da perda não técnica e compará-los foi o tempo de treinamento (*Training*

Time e acurácia) [1] *Accuracy*.

Classificador	Velocidade em B_i	Velocidade em B_c
SMV-RBF	458.23s	504.52s
SMV-LINEAR	90.39s	79.66s
ANN-MLR	82.27s	94.06s
SOM	133.73s	114.99s

Tabela 2.1: Velocidade em segundos do OPF em comparação com os outros classificadores [Ramos et al., 2011]

Classificador	Acurácia	Tempo de treinamento	Tempo de classificação
OPF	86.62 +-2.28	1.03s	0.9102s
SMV-RBF	78.72 +- 3.72	890.67s	0.3622s
SMV-LINEAR	50.52 +-2.34	175.76s	0.0073s
ANN-MLR	50.26 +- 0.78	159.99s	0.0007s
SOM	59.56 +- 0.72	259.73s	0.3070s

Tabela 2.2: Acurácia e média de treinamento e tempo de classificação para OPF, SVM-RBF, SVM-LINEAR, AND ANN-MLP para o *dataset* B_i[Ramos et al., 2011]

Classificador	Acurácia	Tempo de treinamento	Tempo de classificação
OPF	88.29 +-1.48	1.58s	1.3918s
SMV-RBF	81.10 +- 2.57	1500.66s	0.4960s
SMV-LINEAR	50.42 +-0.89	237.03s	0.0086s
ANN-MLR	50.99 +- 1.41	279.87s	0.0014s
SOM	62.92 +- 5.03	341.74s	0.4059s

Tabela 2.3: Acurácia e média de treinamento e tempo de classificação para OPF, SVM-RBF, SVM-LINEAR, AND ANN-MLP para o *dataset* B_c[Ramos et al., 2011]

É demonstrado também que o algoritmo de aprendizado OPF pode melhorar o desempenho do OPF tradicional. Além do mais, é utilizado algoritmo de poda para identificar as amostras mais relevantes do conjunto de treinamento e removê-las do processo de classificação. As experiências demonstraram sem afetar a precisão do conjunto de testes que esta remoção

[1] A acurácia é definida conforme a equação:

$$Acuracia = \frac{Tc}{Ts}.100$$

Sendo que Tc representa o número total de amostras classificados corretamente, já o Ts o número de amostras usadas para o teste.

pode diminuir em até 50% do conjunto de treinamento original e pode, até melhorá-lo em alguns casos.

Pode se observar também que a utilização algoritmo de poda pode acelerar a fase de teste do classificador baseado em OPF, pois isso é de maneira direta proporcional a quantidade de amostras no conjunto de treinamento.

2.1.2 Máquina de vetores de suporte (*Support Vector Machines* - SVM)

As máquinas de vetores de suporte tem como objetivo pré-selecionar consumidores para serem inspecionados com base em irregularidades no comportamento de consumo, com a finalidade de identificar a perda não técnica de energia elétrica [Nagi et al., 2010]. Para atingir os objetivos propostos utilizou-se o algoritmo SVM, que foi introduzido por Vapnik no final dos anos 1960 [Cristianini and Shawe-Taylor, 2000]. Um ponto positivo é que o SVM é bastante resistente ao problema de desequilíbrio de classes.

O principal objetivo do algoritmo SVM é construir uma função de decisão ótima, de modo que preveja com precisão dados não vistos em duas classes e minimizando o erro de classificação. O SVM encontra um hiperplano entre duas classes de dados, para isso segundo [Messinis and Hatziargyriou, 2018] o mesmo tem a capacidade de atribuir diferentes pesos a vários tipos de erros de classificação. Isso pode ser feito se atribuir um alto custo a erros de classificação da classe minoritária, o que pode assim levar a métricas de desempenho mais altas.

As SVMs podem ser confiáveis quando aplicadas à detecção de perdas não técnicas (*Non-technical losses-NTL)*, porém o autor ressalta que pode ser bastante difícil e demorado ajustá-las. Isso pode aumentar o tempo de construção do modelo para os casos com grandes conjuntos de dados. Devido a essa característica se torna uma solução não eficiente para aplicações em tempo real.

Segundo [Messinis and Hatziargyriou, 2018], o SVM pode ser combinado com outros classificadores (por exemplo, sistema de inferência difusa - FIS, Árvores de Decisão, Redes Neurais) com o intuito de melhorar os resultados de classificações, tornando-se assim uma solução básica comum para comparar técnicas de seleção de características e classificações.

Resultados experimentais indicam que o algoritmo proposto pode ser usado confiavelmente para a detecção de anormalidade e atividades fraudulentas. Desta forma, utilizar o SVM para detecção de clientes que praticam fraudes demonstrou ser promissor. Isso se dá pois o SVM

possui hiper-superfícies de divisão não linear, o que lhe dá alta discriminação. Outro ponto forte é que ele fornece uma boa capacidade de generalização para classificação de dados invisíveis. Essas propriedades permitem que o SVM conduza problemas complexos de classificação com facilidade e boa precisão.

Levando em consideração que pressupõe-se que os perfis do cliente contêm irregularidades quando ocorre um evento de fraude, o SVM classifica os perfis de clientes em diferentes categorias: normal e fraude. Existem diferentes tipos de fraude que podem ser identificados, porém os autores concentraram-se apenas em cenários em que há alterações bruscas nos perfis do cliente, indicando possíveis eventos de fraude.

Os resultados dos testes-piloto obtidos para inspeção no local indicou que uma taxa de acerto de detecção média (porcentagem de clientes detectados com anormalidades e atividades de fraude dos suspeitos da lista curta) de 26%, sendo eles 7% de anormalidades e 19% de atividades de fraude. As anormalidades identificadas foram: necessidade de substituição dos medidores, casas abandonadas, mudança de inquilinos e fiação do medidor com defeito.

Com os resultados da classificação do SVM, os autores [Nagi et al., 2010] aprimoraram um sistema que é utilizado para selecionar apenas clientes com altas possibilidades de fraude, conseguindo assim determinar as características mais comuns entre os que cometem fraudes. Com esta abordagem a taxa de detecção nos testes piloto melhorou de 26% para 64%.

A base de dados utilizada continha 105.525 casos de fraude, detectados a partir de inspeção no local em uma determinada área, no período de dezembro de 2000 a julho de 2008. A inspeção nos dados mostrou que houve casos em que clientes foram detectados mais de uma vez por fraude. As ações implementadas para detecção de NTL alcançam uma taxa de detecção de 3% (sem aplicação de aprendizado de máquina) para 60% (com aplicação de aprendizado de máquina).

Após encontrar os parâmetros ótimos, representados por C = 1 e γ = 0.92, foi obtida uma acurácia = 86,43%.

ações implementadas beneficiam não apenas a melhor gestão NTLs, mas complementam as práticas contínuas já existentes para a detecção, com previsão de que grandes economias sejam feitas utilizando este sistema.

Os autores ressaltam que a única limitação do sistema de detecção de fraude é que os clientes que cometeram atividades de fraude antes do período de dois anos podem não serem detectados pois o SVM não é treinado para tais instâncias.

2.1.3 Árvore de Decisão

Para realização dos estudos, [Cody et al., 2015] utilizou dados de aproximadamente 5000 medidores residenciais e 650 comerciais anônimos cedidos pelo Centro de Arquivo de Dados de Ciências Sociais da Irlanda. Os valores dos dados de consumo de energia foram registrados em intervalos de 30 minutos durante um período de aproximadamente dois anos entre 2009-2011. Devido à diversidade e à granularidade, foram aplicadas duas etapas de pré processamento: Remoção de informações nulas, devido a quantidade de leituras foi feito um agrupamento por dia, semana e mês para facilitar os experimentos, que segundo os autores o mês se demonstrou irrelevante no treinamento.

Os autores seleciona um conjunto de medições a partir de um único medidor inteligente durante um certo período de tempo. Para validação é utilizado outro período de tempo. Após a seleção do conjunto de dados de treinamento, esse conjunto de dados é usado para gerar as regras de decisão representativas do modelo de comportamento de consumo de energia normal para o cliente em questão.

Foram realizados diversos experimentos utilizando a ferramenta *WEKA*, e para treinamento e classificação o algoritmo de árvore de decisão.

Com os experimentos constatou-se que o *overfiting* é um dos problemas que mais ocorre no aprendizado com árvore de decisão, sendo extremamente importante escolher de forma adequada o tamanho do conjunto de dados para o treinamento, de forma à garantir que os dados não se sobreponham ou subestime.

Após análise de tentativa e erro feito pelo autor o tamanho do conjunto para o treinamento da árvore de decisão com um agrupamento de leituras de 3 semanas. Devido à falta de rótulos, as atividades fraudulentas foram simuladas para ilustrar cenários do mundo real. Para esta simulação de uma fraude o autor subtraiu um valor aleatório 0 a 0,5 kWh em todas as medições originais do conjunto de dados para validação.

Para detecção de anomalia, é calculado a Raiz do Erro Quadrático Médio (*Root Mean Squared Error - RMSE)*, que é utilizado para medir as diferenças entre os valores previstos e os valores reais. Qualquer valor acima do limite serve como uma indicação de possível fraude de energia.

O cálculo do RMSE é usado para diferenciar entre o comportamento de consumo normal e a possível atividade fraudulenta. Se o valor do RMSE estiver abaixo do limite, o conjunto de

dados será considerado normal. Já se o RMSE estiver acima do limite (que neste caso é de 0,4 kWh), o conjunto de dados é uma possível fraude.

Segundo os autores [Cody et al., 2015] os resultados obtidos a partir dos experimentos conduzidos com o intuito de avaliar a eficácia do uso da aprendizagem da árvore de decisão para aprender o modelo do consumidor e para aplicação do modelo com meios estatísticos para prever a detecção de fraude de energia.

Um dos experimentos, a árvore de decisão foi treinada usando medições de energia de agosto de 2009 e validada usando medições de energia com e sem atividades fraudulentas simuladas a partir de agosto de 2010. O resultado demonstrou que o modelo foi capaz de identificar possíveis comportamentos fraudulentos, sendo que mesmo experimento foi repetido para outros meses, reproduzindo resultados semelhantes.

Em outro experimento foi projetado a avaliação o quão preciso era o modelo para prever valores futuros. Neste experimento os conjuntos de dados fraudulentos foram relatados, assim como outros resultados com experimentos em outras datas.

Segundo [Cody et al., 2015], os resultados dos experimentos realizados revelam a capacidade de prever com precisão os valores de consumo de energia a partir do mesmo mês de um ano, semanas subsequentes e dentro da mesma estação do tempo. Com esse resultado é possível utilizar o algoritmo de aprendizado da árvore de decisão M5P em grupos de dados sob investigação.

2.2 Não supervisionado

2.2.1 Floresta de Caminhos Ótimos

O método de [Júnior et al., 2016] tem como objetivo avaliar agrupamento (*clustering*) do OPF para identificação de perdas não técnicas em sistemas de distribuição de energia. As principais contribuições são voltadas para o uso do OPF não supervisionado na detecção de perdas não técnicas e na modelagem do problema da perda não técnica como sendo uma tarefa de detecção de anomalias.

O trabalho tem como objetivo modelar o problema da identificação de perdas não técnicas como uma detecção de anomalias. O classificador é treinado com o consumo regular, utilizando uma base de dados de onde foram removidos anteriormente os consumidores fraudulentos. Em vista disso quando uma nova amostra surge para ser classificada, identifica-se se

esta amostra diz respeito ao padrão "normal" aprendido pelo classificador, caso contrário, tal amostra é classificada como sendo uma anomalia, ou seja, um perfil de usuário suspeito.

Tendo em vista que a obtenção de conjuntos de dados rotulados neste contexto geralmente é uma tarefa difícil, muitas vezes é preciso avaliar técnicas não supervisionadas para identificação de perda não técnica. Os autores [Júnior et al., 2016] ressaltam que apenas um trabalho [Ramos et al., 2011] que empregou OPF para identificação de NTL não supervisionada até aquele momento.

Duas rodadas de experimentos foram realizadas com o intuito de avaliar a robustez do classificador de OPF para o reconhecimento não supervisionado de NTL e detecção de anomalias, comparando com as técnicas k-means, GMM, AP e Birch com a finalidade de identificar perda não técnica não supervisionada, também foi testado o SVM de uma classe para a tarefa de detecção de anomalias. Nos comparativos o OPF obteve os resultados mais precisos considerando ambas as aplicações em dois conjuntos de dados, compostos por perfis comerciais e industriais de consumidores regulares e irregulares.

Classificador	B_i	B_c
OPF	81.57% +- 2.48	78.30% +- 3.11
GMM	74.64% +- 3.90	74.80% +- 4.35
k-Means	81.51% +- 3.71	77.88% +- 3.48
Affinity propagation	82.56% +- 3.00	79.51% +- 2.33
Birch	51.65% +- 2.23	72.15% +- 12.21

Tabela 2.4: Acurácia média para cada técnica de agrupamento considerando conjuntos de dados B_c e B_i [Júnior et al., 2016]

Para avaliação, os autores utilizam duas bases de dados denominados B_i e B_c, composto por perfis comerciais e outro composto por consumidores industriais, fornecidos por uma empresa de energia elétrica brasileira. O B_i é um conjunto de dados composto por 3178 perfis industriais, e o B_c contém 4948 perfis comerciais. Cada perfil industrial e comercial é representado por oito características:

- Demanda faturada: é o valor da demanda da energia ativa que será considerada para fins de faturamento, medida em quilowatts (kW);
- Demanda Contratada: o valor da demanda por disponibilidade contínua solicitada à empresa de energia, que deve ser paga independentemente ser ou não utilizada, medida em quilowatts (kW);

- Demanda Medida ou Máxima: é verificada por medição em intervalos de 15 minutos durante o período de faturamento, medida em quilowatts (kW);
- Energia Reativa: energia que flui através dos campos elétrico e magnético de um sistema de CA, em quilovolt-amperes horas reativas (kVArh);
- Transformador de Potência: transformador de potência instalado para os consumidores, em quilovolt-amperes (kVA);
- Fator de Potência: a relação entre a potência ativa e aparente consumida em um circuito, sendo que o fator de potência indica a eficiência de um sistema de distribuição de energia;
- Potência Instalada: soma da potência nominal de todos os equipamentos elétricos instalados e prontos para operar na unidade consumidora, em quilowatts (kW);
- Fator de Carga: a razão entre a demanda média e a demanda máxima da unidade consumidora, desta forma o fator de carga mostra como a energia elétrica é usada de maneira racional.

O conjunto de dados comerciais contém 4680 consumidores regulares (94,58%) e 268(5,12%) perfis irregulares, já o conjunto de dados industriais contém 2984 amostras (93,89%) as mesmas representam consumidores regulares e 194(6,11%) amostras que são consumidores irregulares.

O autor ressalta que trabalho contribuiu para a literatura relacionada à detecção de NTL não supervisionada, que carece desse tipo de trabalho. Também é proposto olhar para o problema do reconhecimento de NTL como um problema de detecção de anomalias. Este trabalho não foi comparado com a versão supervisionada do algoritmo Floresta de Caminhos Ótimos.

2.2.1.1 Mapa de auto-organização (*Self Organizing Map* - SOM)

Segundo [Messinis and Hatziargyriou, 2018], o algoritmo Mapa de auto-organização pode ser utilizado para classificação não supervisionada, atuando como um método para redução de dimensionalidade pois geralmente produz uma melhor representação do conjunto de dados.

O SOM tem como saída um conjunto de agrupamentos que pode produzir uma representação visual que fica fácil para compreensão dos dados. O número de agrupamentos resultantes podem ser mais do que dois, se forem considerados comportamentos atípicos ao ponto de formar um novo agrupamento.

Desta forma, para os agrupamentos resultantes, é necessário passar por avaliação de especialistas ou servir de entrada para um nível secundário de lógica, pois não necessariamente estes agrupamentos sejam fraudes.

Um dos pontos negativos é que em sua natureza o SOM não supervisionado tem uma precisão reduzida, levando a aparecer poucas vezes em literaturas de detecção de NTL.

2.2.1.2 Detecção de *outliers* (*outlier detection*)

A detecção de NTL pode ser identificada utilizando conceitos da detecção de *outliers*, que é identificar um dado que tem suas características muito diferente da maioria de um dado conjunto de dados.

Segundo [Messinis and Hatziargyriou, 2018] os algoritmos que utilizam este conceito são a Distribuição Gaussiana Multivariada (*Multivariate Gaussian Distribution* - MGD), fator de *outlier* local (*Outlier Factor* - LOF), K-médias (*K-means*) e Maximização de Expectativas (*expectation–maximization* E-M) e divergência de Kullback-Leibler (*Kullback–Leibler divergence* - KLD).

A Distribuição Gaussiana Multivariada é usada por [Júnior et al., 2016]. Sendo que o mesmo utiliza um conjunto de dados livre de fraudes e cada grupo é modelado como uma distribuição gaussiana. Ao incluir uma nova amostra é calculada a probabilidade da mesma pertencer a uma das distribuições existentes. Posteriormente a maior destas probabilidades é comparada com um valor limite estabelecido, a partir deste ponto é decidido se a amostra é dada ou não como um anomalia. Esta abordagem apresenta um desafio que é identificar a quantidade de agrupamentos e os parâmetros das distribuições.

O Fator de *outlier* local é usado por [Mashima and Cárdenas, 2012], diferente do MGD e do LOF é usado para calcular a densidade local de uma amostra e equiparar com a densidade local média dos vizinhos mais próximos da amostra.

Amostras com densidade local menor que suas vizinhas podem ser identificadas como anomalias. Um ponto positivo é que o método não exige um conjunto de dados sem fraudes para a etapa de treinamento, porém um valor alto não necessariamente é uma fraude. Desta forma a tarefa de escolha do limite e as regras adicionais do algoritmo são extremamente importante para uma maior assertividade.

No algoritmo de Divergência de *Kullback-Leibler* (KLD) é tida como base para o trabalho do [Krishna et al., 2016] e tem como proposta detectar ataques inteligentes que disfarçam

o uso malicioso como benigno, usado em redes *smart Grid*. O KLD é uma medida de distância entre duas distribuições de probabilidade e pode ser usado para comparar a distribuição de um conjunto de medições com uma linha de base, obtida a partir da distribuição histórica. Em seu conjunto de treinamento podem conter fraudes que o algoritmo consegue ter uma boa identificação, se tornando assim um ponto positivo.

Os algoritmos não supervisionados têm uma maior assertividade e aplicação em redes *smart Grid* e são assim pouco utilizadas em redes tradicionais. Por isso, não serão abordadas em detalhes neste trabalho.

3 ANÁLISE EXPLORATÓRIA

Neste capítulo, serão apresentadas as configurações dos experimentos, os métodos analisados e os resultados obtidos.

3.1 Base de Dados

Inicialmente, foram contatados os autores dos artigos [Trevizan et al., 2015], [Ramos et al., 2011] e [Messinis and Hatziargyriou, 2018] porém todos responderam que não podiam disponibilizar as bases de dados utilizadas em suas pesquisas devido aos dados serem privados da concessionária.

A base de dados utilizada para a realização dos experimentos foi obtida através de uma solicitação para [for Energy Regulation , CER], e disponibilizada em outubro de 2018. Os dados são leituras de medidores de energia elétrica correspondente a um período entre os anos de 2009 e 2012, contendo 485 consumidores industriais, 4225 residenciais e outros 1735 clientes, dispostos em 6 arquivos. A estrutura dos arquivos pode ser vista na Tabela 3.1.

	Consumidor	Data/hora codificados	Consumo de energia em kWh
leitura 1	1392	19503	0.14
leitura 2	1392	19504	0.138
...	...	...	...
leitura 98	1392	19643	0.516
leitura 99	1392	19644	1.875

Tabela 3.1: Estrutura do arquivo de dados.

O campo *consumidor* tem como função identificar unicamente o usuário. Data e hora representam o momento da leitura, como se trata de uma rede *Smart Grid*(Rede Inteligente) [2] a leitura é feita com um intervalo de 30 minutos, sendo que esse consumo é medido em kWh.

3.2 Pré processamento

Os dados de leitura estão originalmente organizados conforme a Tabela 3.1, na qual cada linha representa uma nova leitura. Tal formato não é adequado para o treinamento dos métodos

[2] Conforme definido por [de Souza et al., 2013] uma rede inteligente é um sistema complexo de ponta a ponta, composto por vários subsistemas de energia interconectados e inter-relacionados entre si por meio de diversos protocolos com camadas adicionais de tecnologia (energia, comunicações, controle/automação e TI).

pois os algoritmos utilizados esperam os atributos conforme Tabela 3.2. Ou seja,os dados foram reorganizados de maneira que cada linha representa uma unidade consumidora e cada coluna uma leitura.

	leitura 1	leitura 2	...	leitura 98	leitura 99
consumidor 01	0.14	0.138	...	0.516	1.875
...	...	...	...	...	...
consumidor 100	0.237	0.238	...	0.232	0.229

Tabela 3.2: Estrutura reorganizada do arquivo de dados, sendo que as colunas representam as leituras e as linhas cada um dos consumidores, representado pelo número identificador da unidade consumidora.

Como a base de dados não possui rótulos referentes a consumidores que cometeram fraudes, foi necessário criá-los sinteticamente. Com base na metodologia proposta em [Cody et al., 2015], foi subtraído randomicamente valores de todas as leituras de alguns consumidores para geração das fraudes. Desta forma, foram geradas seis novas bases com uma nova coluna contendo o rótulo de cada cliente conforme a tabela 3.3.

	leitura 1	leitura 2	...	leitura 98	leitura 99	Rótulo
consumidor 01	0.14	0.138	...	0.516	1.875	Sem Fraude
...	...	...	...	...	...	...
consumidor 100	0.237	0.238	...	0.232	0.229	Fraude
consumidor 101	0.42	1.88	...	0.62	0.197	Sem Fraude

Tabela 3.3: Base de dados com rótulos sintéticos adicionados.

Com base na quantidade de leituras, as 6 bases foram classificadas em dois tipos: BaseA com 10% de fraudes sintéticas; e a base BaseB com 30% de fraudes sintéticas. Em cada uma das bases de dados foi subtraído um valor para geração de ruído. Por exemplo, (todos os campos da BaseA e da BaseB foram gerados com alterações no valores variando de 0,1 à 0,5, 0,6 à 1,5 e 2,5 à 3,5). Isto de fato resultou em 6 bases de dados com variações de ruído que pode se mapeada através da Tabela 3.4. Tais bases foram geradas com intuito de analisar o comportamento de cada algoritmo dependendo da quantidade de ruído adicionado. Desta forma, os novos registros de consumo foram classificados como fraude de energia.

	BaseA	BaseB
Sub1	0,1 - 0,5	0,1 - 0,5
Sub2	0,6 - 1,5	0,6 - 1,5
Sub3	2,6 - 3,5	2,6 - 3,5

Tabela 3.4: Bases de dados criadas com fraudes sintéticas.

3.3 Métricas e Configurações

Esta seção apresenta as bibliotecas, ferramentas e configurações que foram utilizadas no pré-processamento e execuções. As etapas do processo foram codificadas utilizando a linguagem de programação *Python* [3]. Para executar os algoritmos de classificação, foi utilizada a biblioteca lib_opf [4] e *scikit-learn* [5] que possuem várias ferramentas para utilização com algoritmos de aprendizado de máquina, bem como ferramentas para leitura e manipulação dos dados. Todos os códigos das ferramentas utilizadas para os testes estão disponíveis publicamente e *open-source*.

Os algoritmos e as configurações utilizados para os experimentos foram:

- OPF - Utilizando a lib_opf, configuração utilizada para a função *(learning="default", metric="euclidian", precomputed_distance=False.)*
- SVM - foi utilizado a *GridSearchCV* [6] do *scikitlearn* que retorna os melhores parâmetros na base de dados. Para obter os melhores parâmetros o algoritmo espera amostras dos melhores candidatos para cada parâmetro que se deseja encontrar. O *GridSearchCV* considera exaustivamente todas as combinações e resulta na melhor para a base de dados específica, por isso foi necessário executar o algoritmo para cada das base de dados que foram criadas. Os candidatos utilizados no algoritmo foram:
  - Parâmetro C = [0.000000001, 0.00000001, 0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 20]
  - Parâmetro Gamma = [0.000000001, 0.00000001, 0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1]
  - Kernel = ['linear','rbf']

[3] https://www.python.org
[4] https://github.com/dhaalves/LibOPF
[5] https://scikit-learn.org
[6] https://scikitlearn.org/stable/modules/grid_search

A Tabela 3.5 contém as melhores configurações para cada base de dados:

	C	Gamma	Kernel
BaseA Sub1	0.000000001	0.000000001	'linear'
BaseA Sub2	20.0	0.000001	'rbf'
BaseA Sub3	10.0	0.000001	'rbf'
BaseB Sub1	20.0	0.00001	'rbf'
BaseB Sub2	20.0	0.000001	'rbf'
BaseB Sub3	20.0	0.000001	'rbf'

Tabela 3.5: Configuração ideal para o SVM de acordo com a Base de Dados.

Os outros parâmetros não são informados, desta forma utiliza o padrão, como por exemplo *cache_size=200, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False*

- Árvore de Decisão - Utilizando o *scikit-learn* configuração utilizada para o algoritmo *(criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False).*

As métricas mais básicas e frequentemente utilizadas para avaliar os resultados segundo [Manning et al., 2008] é a precisão e revocação, na qual serão utilizadas neste trabalho para medição e comparar os resultados encontrados.

- Precisão (P) - Precisão é a fração de consumidores recuperados que são relevantes, calculada conforme:

$$P = \frac{ItensRelevantesRecuperados(tp)}{TotaldeItensRecuperados(tp + fp)}$$
$$P = tp/(tp + fp)$$

- Revocação (R) - Revocação consumidores relevantes que são recuperados, calculada conforme:

$$R = \frac{ItensRelevantesRecuperados(tp)}{TotaldeItensRelevantes(tp + fn)}$$
$$R = tp/(tp + fn)$$

	Relevante	**Não Relevante**
Recuperado	Verdadeiro Positivo (tp)	Falso Positivo (fp)
Não Recuperado	Falso Negativo(fn)	Verdadeiro Negativo (tn)

Tabela 3.6: Matriz de confusão.

Para facilitar o entendimento do F1 foi montado a Tabela 3.6.

A métrica de precisão é frequentemente usada para avaliar problemas de classificação. Se for utilizado somente a precisão, uma fração de suas classificações estão corretas já que existem duas classes possíveis, relevantes e não-relevantes. De acordo com [Manning et al., 2008] por esta razão a precisão não é uma medida ideal para ser utilizada sozinha, há risco de ter alta taxa de falsos positivos, em contrapartida rotular todos como não relevantes é completamente insatisfatório. Desta forma, as medidas de precisão e revocação juntas concentram em avaliar o retorno de verdadeiros positivos.

- F1 - O F1 complementa as medidas de precisão versus revocação por uma média harmônica e ponderada de ambas, calculada conforme:

$$F1 = \frac{2PR}{(P + R)}$$

Para implementar essas métricas foi utilizado o pacote *sklearn.metrics*, para Precisão a função *precision_score*, revocação a função *recall_score* e F1 a função *f1_score*. Os parâmetros utilizados nestas funções foram labels=None, pos_label=1, average='binary', sample_weight=None.

Os experimentos foram realizados fazendo validação cruzada com nove variações no tamanho da base de treinamento e base de teste. Para cada variação do tamanho os algoritmos foram executados dez vezes e a partir disso seus resultados foram computados a média e desvio padrão das métricas. Desta forma, fica visível o comportamento de cada algoritmo e facilitando a escolha da melhor maneira de executar.

3.4 Execução dos Experimentos

3.4.1 Algoritmos Supervisionados

Os experimentos com algoritmos supervisionados foram divididos em três partes, levando em consideração o tipo da base, conforme Tabela 3.4.

Os experimentos inicialmente foram realizados utilizando todas as leituras conforme Tabela 3.7. Porém ao decorrer dos experimentos fazendo testes com agrupamentos diferentes, o agrupamento diário se mostrou mais eficiente conforme Tabela 3.8, utilizando a metodologia de pré processamento descrita na seção 3.2. Os gráficos apresentados são com somas de leituras equivalente a um dia.

	leitura 1	leitura 2	...	leitura 98	leitura 99
consumidor 1	0.14	0.138	...	0.516	1.875
...	...	...	...	...	...
consumidor 99	0.237	0.238	...	0.232	0.229

Tabela 3.7: Base de dados com todas as leituras.

	Agrupamento 0	Agrupamento 1	...	Agrupamento 20	Agrupamento 21
consumidor 1	31.85	35.56	...	52.02	49.90
...	...	...	...	...	...
consumidor 99	38.01	17,89	...	13.72	31.77

Tabela 3.8: Base de dados com as leituras agrupadas por dia.

Nas figuras 3.1 (BaseA sub1) e 3.2 (BaseB Sub1) são os resultados das base de dados onde é subtraído randomicamente valores de 0 à 0,5. No OPF, SVM e árvore de decisão, os resultados reportam poucas fraudes identificadas, visto que este é o cenário onde houve a menor subtração nas leituras, indicando uma leve alteração no comportamento do usuário, o que leva o algoritmo a não identificar a fraude. Para o gráfico na figura 3.1 o SVM não classificou nenhum caso como fraude, já o OPF e árvore de decisão tiveram o F1, com menos de 40%. Já no Gráfico 3.1, a precisão foi maior, mas o F1 ficou entre 40 a 60%, devido a quantidade maior de ruído inseridos na mesma.

Nos Gráficos 3.3 (BaseA sub2) e 3.4 (BaseB Sub2) foi usado o mesmo percentual de inserção de ruídos, porém foi alterada a quantidade de ruído para um intervalo de 0,6 à 1,5. Nestes casos, todos os algoritmos tiveram um aumento na precisão, revocação e F1.

Para o Gráfico 3.3, por exemplo, todos os métodos tiveram resultados entre 40% a 80% em suas métricas. No entanto, devido ao alto desvio padrão todos ficaram com desempenho semelhantes para o F1, mas estatisticamente não ficaram empatados, com o SVM sobressaindo aos outros, 19% melhor do que o OPF e 10% da árvore de decisão. Para o gráfico na figura 3.4 houve uma melhora promissora, pois todas as métricas ficaram acima de 80%, diminuindo

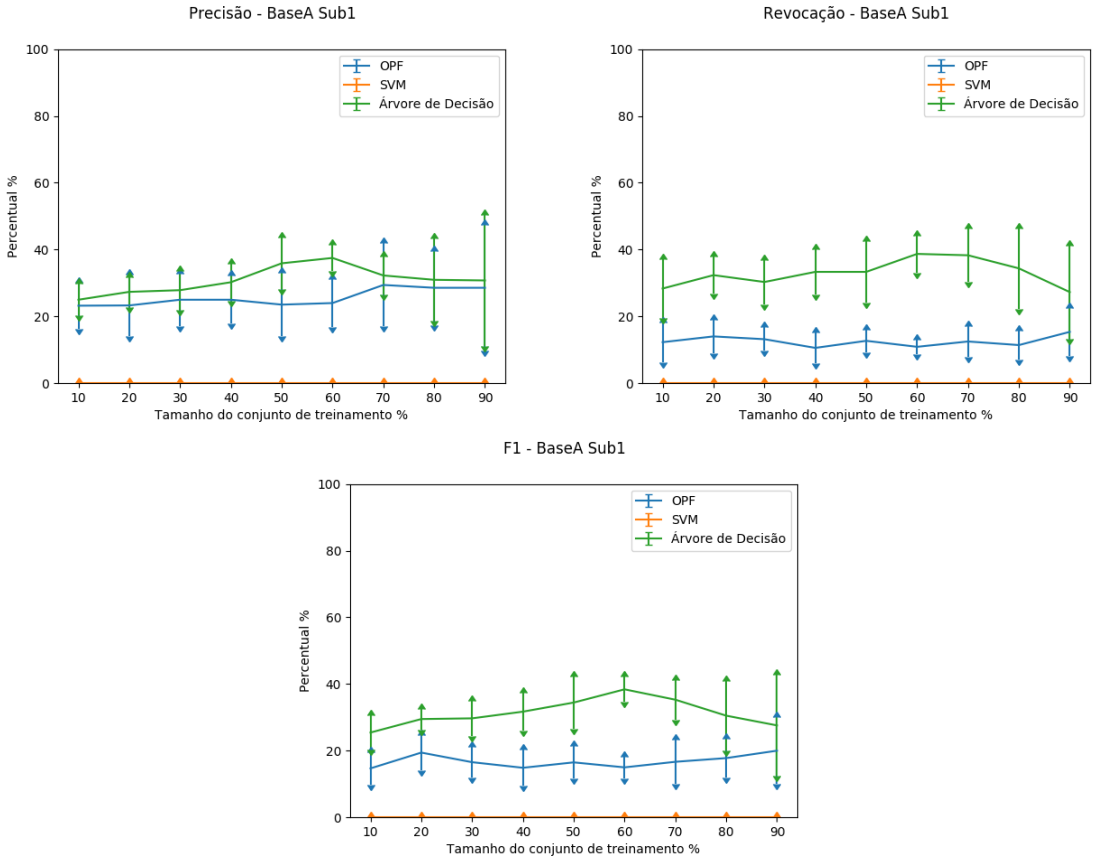

Figura 3.1: BaseA Sub1

significativamente o desvio padrão das execuções. Conforme foi incrementado o tamanho da base de treinamento todos os algoritmos foram melhorando.

O SVM se saiu estatisticamente melhor do que os demais mas a árvore de decisão ficou muito próximo em alguns casos até superior. Ao se fazer o teste estatístico, observou-se que o SVM manteve-se superior, com segundo melhor resultado a árvore de decisão e por último OPF.

Nos gráficos das figuras 3.5 (BaseA sub3) e 3.6 (BaseB Sub3) foi ampliada a quantidade de ruídos, com intervalo randômico de 2,5 à 3,5. Para ambas bases foram identificadas melhorias em todas as métricas se comparado aos experimentos anteriores. Para os gráficos da figura 3.5, o tamanho da base de treinamento influenciou significativamente nos resultados, tendo uma variação de somente 10% nos resultados se comparar da primeira até a última execução. O SVM ficou superior aos outros algoritmos, mas o OPF e árvore de decisão se mostraram estatisticamente empatados.

Para os gráficos na figura 3.6 ao observar as 9 execuções tem uma dentre elas, apresentando um baixo desvio padrão e com diferença menor de 10% entre elas, tendo isso em vista e

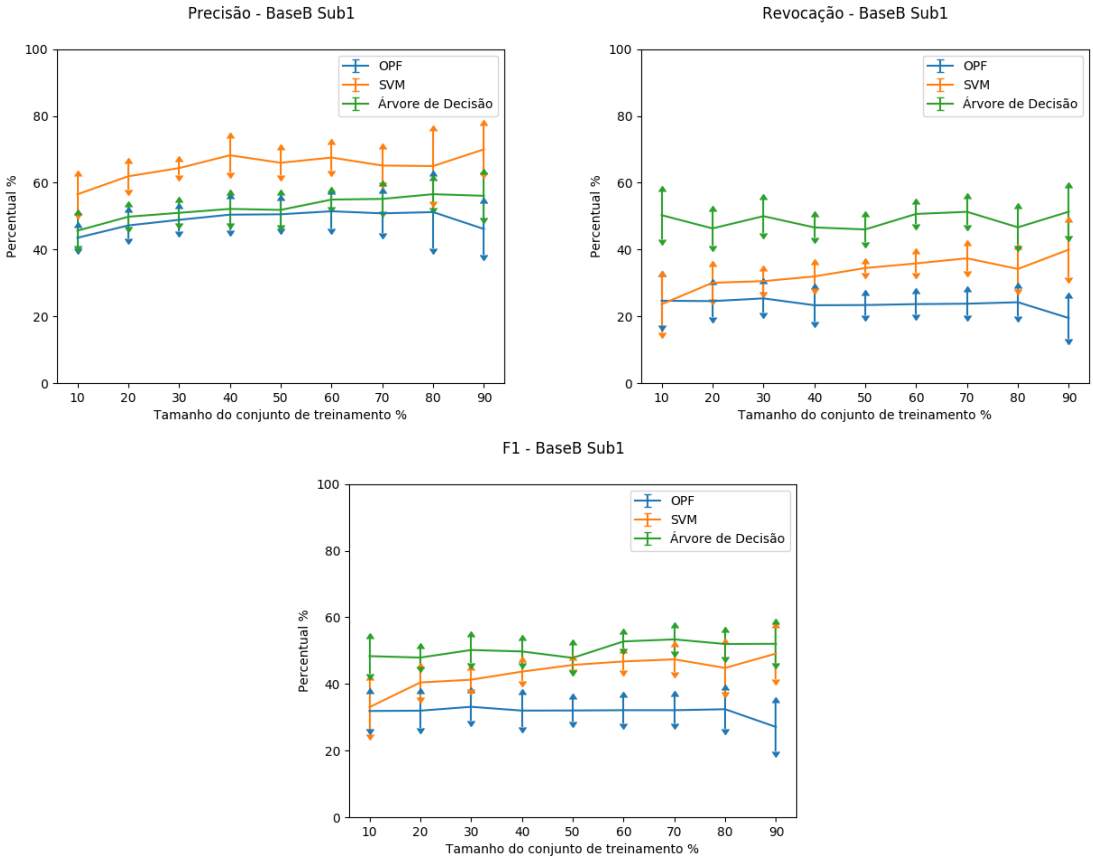

Figura 3.2: BaseB Sub1

fazendo o *t-test*, o SVM ficou 4% superior ao OPF, mas estatisticamente empatado com a árvore de decisão.

3.4.2 **Algoritmo não-supervisionado**

Foram executados experimentos utilizando o algoritmo Floresta de Caminhos Ótimos (OPF) não supervisionado com as mesmas bases de dados utilizadas nos algoritmos supervisionados.

Antes de executar o algoritmo com as base de dados da Tabela 3.4, foi executado o algoritmo Floresta de Caminhos Ótimos (OPF) utilizando variações de diferentes de parâmetros e agrupamentos na sua execução, mas em todo os casos o algoritmo não conseguiu identificar as fraudes.

Desta forma, tendo em vista os experimentos demonstrados através das Figuras 3.1 à 3.6, foi executado o OPF com bases contendo agrupamentos de leituras com outras variações (agrupamento Semanal, quinzenal, mensal, bimestral), mas sem a obtenção de resultados significativos.

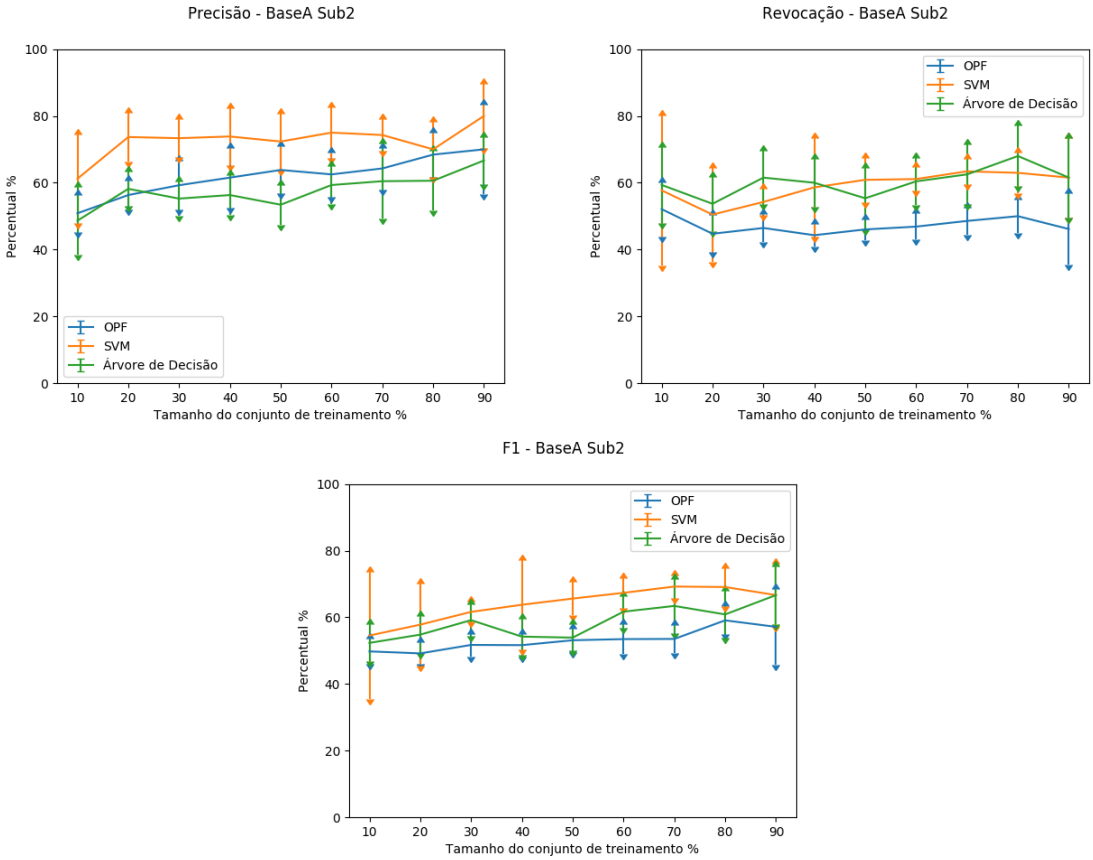

Figura 3.3: BaseA Sub2

A Tabela 3.9 contém os parâmetros utilizados e os resultados obtidos.

Tam. Treino	Verdadeiro Positivo	Verdadeiro Negativo	Falso Positivo	Falso Negativo
90%	4	803	283	17
80%	3	843	287	11
70%	1	867	299	15
60%	7	848	314	50
50%	6	872	320	58
40%	6	907	332	49
30%	0	1018	351	0

Tabela 3.9: Base de dados com as leituras agrupadas por dia.

Como a proposta deste trabalho é comparar os algoritmos supervisionados e o não supervisionado nas mesmas condições, parametrizações e mesmo pré processamento, foi possível concluir que o resultado do método não supervisionado não foi considerado satisfatório em relação aos algoritmos supervisionados, pois não foram identificados os ruídos sintéticos de fraude em nenhuma das bases de dados descritas na Tabela 3.4

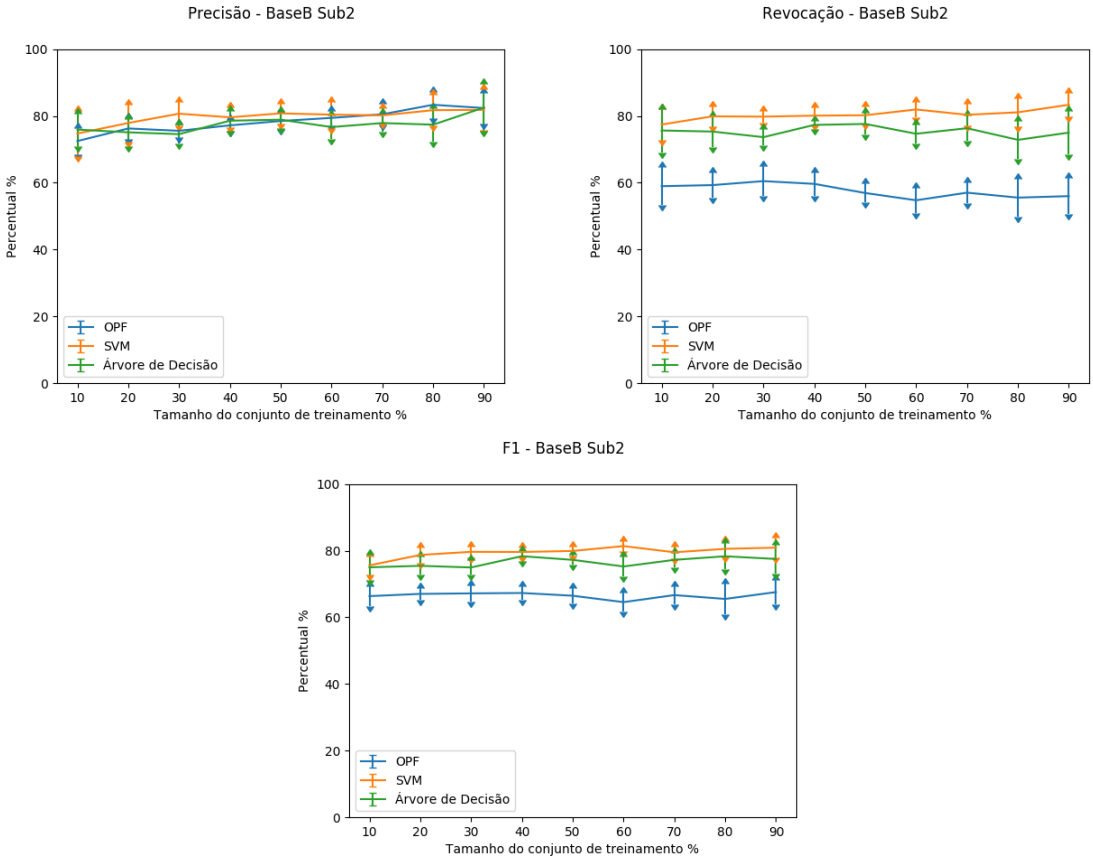

Figura 3.4: BaseB Sub2

Se for utilizar o algoritmo OPF não supervisionado para identificação de perda não técnica com esta base de dados, deve ser estudado de forma mais aprofundada o comportamento do OPF, com intuito de fazer um pré processamento da base mais eficiente.

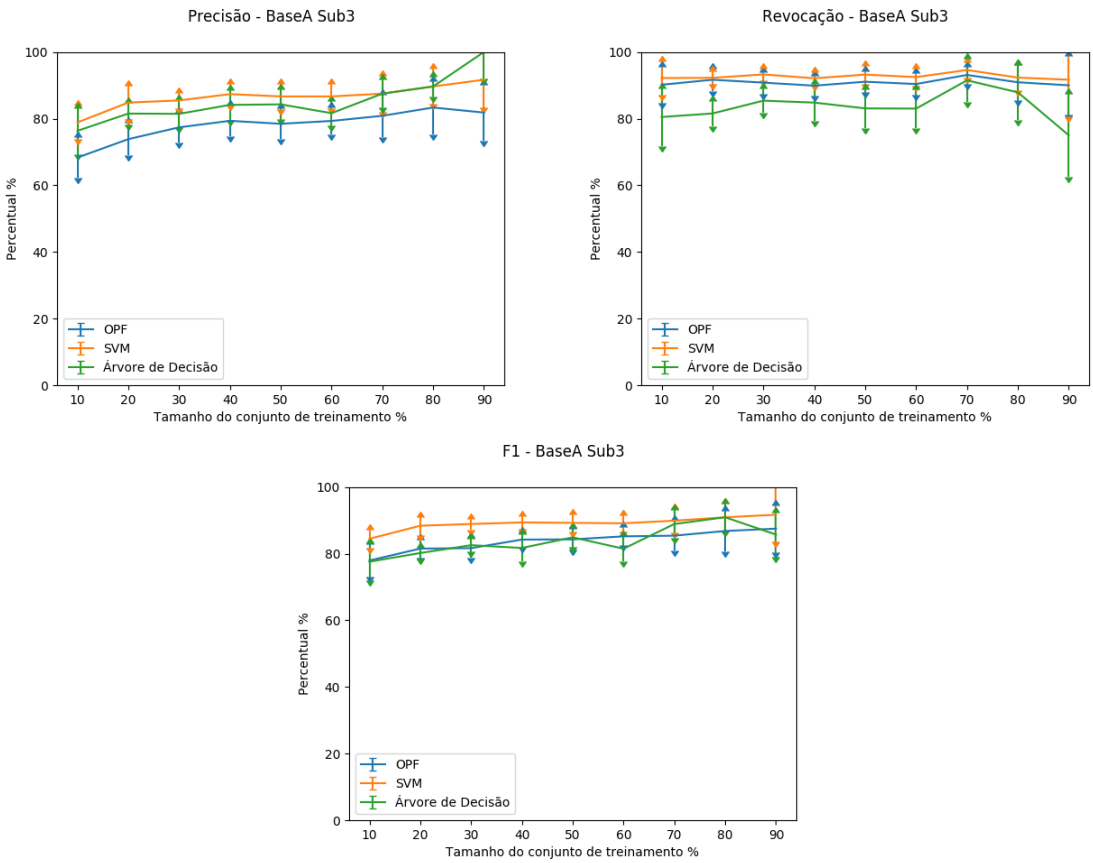

Figura 3.5: BaseA Sub3

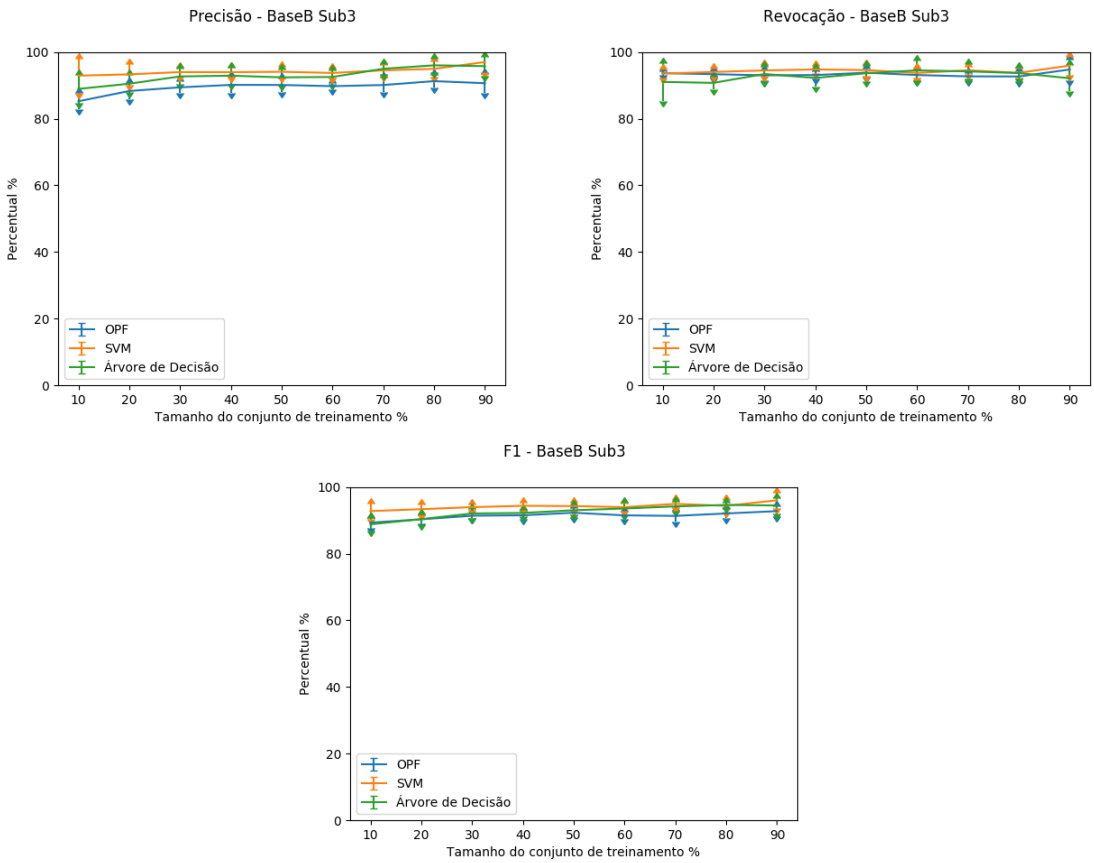

Figura 3.6: BaseB Sub3

4 CONCLUSÃO

Esse trabalho teve como objetivo comparar algoritmos supervisionados e não supervisionado com a finalidade de identificar consumidores com perfil fraudulento. Para tal, foram utilizados algoritmos de aprendizagem de máquina OPF, SVM e árvore de decisão supervisionado e o OPF não-supervisionado.

Os experimentos foram realizados com diversas combinações de parâmetros e agrupamento de leituras, com a finalidade de entender o comportamento dos algoritmos mediante a variação nas bases de dados. Os algoritmos de aprendizado de máquina supervisionado OPF, SVM e Árvore de decisão se mostraram bem próximos, porém no melhor caso o SVM se mostrou 4% superior ao OPF e 2% superior à árvore de decisão. Estes experimentos foram realizados com a base de dados que apresentaram quantidades maiores de ruído em sua composição. Já em bases com pouco ruído, o SVM se mostrou melhor do que os outros algoritmos, 6% superior ao OPF e 7% superior à árvore de decisão.

Utilizando a mesma abordagem de pré processamento, o algoritmo OPF não supervisionado não resultou em dados satisfatórios. Por exemplo, no melhor caso identificou somente 6 unidades consumidoras como fraude, e 58 unidades que não são fraudulentas, de um total de 1256 unidades consumidoras. Dessa forma, podemos concluir que o método OPF não supervisionado não deve ser usado com essa abordagem de pré processamento de dados e parametrizações.

4.1 Trabalhos Futuros

Uma forma de complementar o trabalho é utilizar uma outra abordagem para o pré processamento para o algoritmo OPF não supervisionado e comparar com outros métodos não supervisionados. Além disso, aplicar estes algoritmos em uma base de dados com rótulos reais para a posterior comparação dos resultados.

REFERÊNCIAS

[1] G. E. d. A. P. Batista et al. *Pré-processamento de dados em aprendizado de máquina supervisionado*. PhD thesis, Universidade de São Paulo, 2003.

[2] C. Cody, V. Ford, and A. Siraj. Decision tree learning for fraud detection in consumer energy consumption. *IEEE Transactions on Power Systems*, pages 1175–1179, 2015.

[3] N. Cristianini and J. Shawe-Taylor. An introduction to support vector machines, 2000.

[4] J. N. de Souza, P. Lorenz, and A. Jamalipour. Ultimate technologies and advances for future smart grid: utasg [guest editorial]. *IEEE Communications Magazine*, 51(1):66–67, 2013.

[5] S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni. Electricity theft: Overview, issues, prevention and a smart meter based approach to control theft. *Energy Policy*, 39(2):1007–1015, 2011.

[6] D. F. Ferreira. *Estatística multivariada*. Editora Ufla Lavras, 2008.

[7] C. for Energy Regulation (CER). Cer smart metering project - electricity customer behaviour trial, 2009-2010 [dataset]. 1st edition. irish social science data archive. sn: 0012-00., 2012. URL www.ucd.ie/issda/CER-electricity. Disponibilizada em outubro de 2018.

[8] L. A. P. Júnior, C. C. O. Ramos, D. Rodrigues, D. R. Pereira, A. N. de Souza, K. A. P. da Costa, and J. P. Papa. Unsupervised non-technical losses identification through optimum-path forest. *Electric Power Systems Research*, 140:413–423, 2016.

[9] V. B. Krishna, K. Lee, G. A. Weaver, R. K. Iyer, and W. H. Sanders. F-deta: A framework for detecting electricity theft attacks in smart grids. In *Dependable Systems and Networks (DSN), 2016 46th Annual IEEE/IFIP International Conference on*, pages 407–418. IEEE, 2016.

[10] S. G. Lakshmi and S. S. Kumar. Integrated substation automation and monitoring system. *IEEE Transactions on Power Systems*, pages 1–4, 2013.

[11] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

[12] D. Mashima and A. A. Cárdenas. Evaluating electricity theft detectors in smart grid networks. In *International Workshop on Recent Advances in Intrusion Detection*, pages 210–229. Springer, 2012.

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

[14] M. C. Monard and J. A. Baranauskas. Conceitos sobre aprendizado de máquina. In *Sistemas Inteligentes Fundamentos e Aplicações*, pages 89–114. Manole Ltda, Barueri-SP, 1 edition, 2003. ISBN 85-204-168.

[15] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and M. Mohamad. Nontechnical loss detection for metered customers in power utility using support vector machines. *IEEE transactions on Power Delivery*, 25(2):1162–1171, 2010.

[16] C. C. Ramos, A. N. Souza, D. S. Gastaldello, R. Y. Nakamura, and J. P. Papa. Identificacao de perdas nao-técnicas utilizando agrupamento de dados por floresta de caminhos otimos. *X SBAI – Simpósio Brasileiro de Automação Inteligente*, 2011.

[17] C. C. O. Ramos, A. N. de Sousa, J. P. Papa, and A. X. Falcao. A new approach for nontechnical losses detection based on optimum-path forest. *IEEE Transactions on Power Systems*, 26(1):181–189, 2011.

[18] R. D. Trevizan, A. S. Bretas, and A. Rossoni. Nontechnical losses detection: A discrete cosine transform and optimum-path forest based approach. In *North American Power Symposium (NAPS), 2015*, pages 1–6. IEEE, 2015.