UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DIOGO VIANA ROCHA

APLICAÇÃO DE TECNICAS DE MINERAÇÃO DE DADOS PARA DETECTAR FRAUDES NA REDE DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA
2016

DIOGO VIANA ROCHA

APLICAÇÃO DE TECNICAS DE MINERAÇÃO DE DADOS PARA DETECTAR FRAUDES NA REDE DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

Trabalho de Diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – CSTADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Doutor Gloria Patricia Lopez.

Co-orientador: Prof. Doutor Hugo Andres Ruiz Florez.

MEDIANEIRA

2016

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Graduação e Educação Profissional
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

TERMO DE APROVAÇÃO

APLICAÇÃO DE TECNICAS DE MINERAÇÃO DE DADOS PARA DETECTAR FRAUDES NA REDE DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

Por

DIOGO VIANA ROCHA

Este Trabalho de Diplomação (TD) foi apresentado às 13:50 horas do dia 16 de novembro de 2016 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Analise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. O acadêmico foi arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com louvor e mérito.

Prof. Me. Gloria Patricia Lopez Sepúlveda
UTFPR – *Campus* Medianeira
(Orientador)

Prof. Dr. Claudio Leones Bazzi
UTFPR – *Campus* Medianeira
(Convidado)

Prof. Dr. Arnaldo Candido Junior
UTFPR – *Campus* Medianeira
(Convidado)

Prof. Dr. Hugo Andes Ruiz Florez
UTFPR – *Campus* Medianeira
(Co-Orientador)

Prof. Me. Jorge Aikes Junior
UTFPR – *Campus* Medianeira
(Responsável pelas atividades de TCC)

A folha de aprovação assinada encontra-se na Coordenação do Curso.

DEDICATÓRIA

Dedico este trabalho aos meus pais, que com muito carinho e paciência me apoiaram na realização deste sonho.

Ao meu irmão pelo apoio dispensado em todos os momentos que precisei.

AGRADECIMENTOS

É certo de que estas linhas não irão mencionar todas as pessoas que fizeram parte desta importante fase da minha vida. Porém elas podem estar certas de que fazem parte do meu pensamento e de minha gratidão.

Primeiramente, agradeço a Deus, que torna tudo possível.

Reverencio a Professora Gloria Patricia Lopez pela sua dedicação, orientação e apoio durante todo o trabalho desenvolvido.

Ao professor Hugo Andres Ruiz Florez, pela atenção e dedicação como co-orientador, sempre esteve disposto a ajudar.

Aos amigos que fiz durante o curso, pela amizade, incentivo e apoio constante.

A todos os docentes do curso de Análise e Desenvolvimento de Sistemas, pela dedicação e ensinamentos disponibilizados em sala de aula, cada um deles de forma especial contribuiu para a realização deste trabalho.

E por fim gostaria de agradecer aos amigos e familiares pelo carinho e apoio, a todos que contribuirão direta ou indiretamente para que esse trabalho fosse realizado.

RESUMO

ROCHA, Diogo Viana. Nesta Monografia (APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS PARA DETECTAR FRAUDES NA REDE DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA). É utilizada a inteligência computacional mediante o uso de um modelo de mineração de dados baseado no processo de descoberta de conhecimento em base de dados, também conhecido como KDD. Este modelo tem como objetivo auxiliar na resolução de um dos grandes problemas enfrentado pelas empresas concessionárias de energia elétrica em todo Brasil, que é o caso da identificação de usuários fraudulentos nas redes de distribuição de energia elétrica. Neste trabalho, o objetivo foi realizar a identificação de usuários que fazem algum tipo de fraude nas redes de distribuição de energia elétrica, visando reduzir as perdas de energias oriundas de fraudes e diminuir a quantia dos gastos com inspeções em campo. No desenvolvimento desta pesquisa, foram estudadas várias técnicas e algoritmos de mineração de dados, dentre eles foram escolhidos os algoritmos de maquinas de vetor de suporte e árvores de decisão, os quais foram gerenciados através do software de mineração de dados WEKA, este software permite realizar o treinamento e validação dos algoritmos de mineração de dados para serem utilizado no processo de identificação dos usuários que devem ser inspecionados. Neste trabalho são descritas as técnicas de mineração de dados utilizadas e, é descrito o estudo de caso de uma empresa concessionária de energia elétrica, onde foi praticada a seleção de usuários a serem inspecionados por suspeita de fraude segundo a indicação dos modelos de mineração de dados utilizados durante a pesquisa. Os resultados obtidos demostraram ser satisfatórios quando avaliados com as inspeções realizadas em campo.

Palavras chave: Inteligência Computacional, Mineração de Dados, Processo de KDD, Weka, Detecção de fraudes, Árvores de Decisão, Maquinas de Vetores de Suporte.

RESUMO EM LINGUA ESTRANGEIRA

Rocha, Diogo Viana. In this Monography (APLICATION OF TECHNIQUES OF DATA MINING TO DETECT FRAUD ON ELECTRICITY DISTRIBUTION NETWORKS) Computational intelligence is used through the application of a data mining model based on the process of knowledge discovery in databases, known as KDD. The model purpose is assist in solving one of the major problem faced by concessionaries of electric distribution throughout Brazil, which is the case of identifying fraudulent users on the electricity distribution networks. In this work, the objective is to identify user who do any type of fraud in the electricity distribution networks, seeking to reduce the losses due frauds and decrease the amount of spending time with inspections in the field. At the development of this research, were studied several techniques and algorithms of data mining, among them were chosen the algorithms of support vector machines and decision trees, which were managed through the WEKA data mining software, this software enables the training and validation of various data mining algorithms, to be used in the process of identifying user that must be inspected. This paper describes the data mining techniques that were used and also describes the case of study of an electric distribution company, where were applied the selection of users to be inspected due suspicion of fraud according to models of data mining used in this research. The results where satisfactory when evaluated with inspections in the field.

Key Words: Computational Intelligence, Data Mining, KDD, Weka, Detection of Frauds Support Vector Machines, Decisions Trees.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

As últimas décadas foram marcadas pelo avanço dos recursos computacionais, linguagens de programação e paradigmas computacionais. Devido a esse avanço e a disseminação das novas tecnologias no cotidiano, houve um crescimento na quantidade de dados coletados e armazenado em meios eletrônicos. O simples armazenamento e recuperação destas informações já trazem um grande benefício, porém não proporciona todas as vantagens possíveis (NAVEGA,2002).

De acordo com FAYYAD et al (1996 a), dados produzidos e armazenados em grande escala são inviáveis de serem lidos ou analisados por especialistas através dos meios convencionais. Portanto, é necessário o uso de uma ferramenta capaz de transformar grandes massas de dados em informações úteis, este processo é conhecido como *Knowledge Discovery in Databases* (KDD). Na visão de Piatetsky-Shapiro (1991), KDD se refere a todo o processo de descobrimento de informações úteis em bases de dados, mas seu maior foco é a etapa de mineração de dados (MD).

O processo de MD é responsável por identificar padrões novos, relevantes, potencialmente úteis e compreensíveis (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996 b). A MD faz uso de diversos algoritmos, como por exemplo, redes neurais artificiais (RNA), árvore de decisão (AD) e máquina de vetor de suporte (MVS), entre outros, capazes de processar dados e encontrar padrões novos e valiosos.

Cabe ressaltar que os algoritmos atuais, mesmo sendo capazes de descobrir padrões, ainda não conseguem determinar se o mesmo é importante (NAVEGA, 2002). Por este motivo a MD requer a interação de analistas, responsáveis por determinar a importância dos padrões encontrados.

O processo de KDD e a MD podem ser usados para ajudar a resolver problemas de diversas áreas, como por exemplo, a área de distribuição de energia elétrica.

Atualmente as empresas de distribuição de energia elétrica lidam com vários tipos de perdas de energia, um desses tipos são as perdas comerciais (furtos, anomalias e fraudes de energia elétrica). As perdas de energia são encaradas como um problema mundial que prejudica a sociedade, elas provocam aumentos na tarifa de fornecimento e causam acidentes (DELGADO, 2010). O problema de fraudes de energia tem se mostrado difícil de solucionar, e resulta em grandes perdas de receita as empresas concessionárias distribuidoras de energia elétrica (DELAIBA et al, 2004).

Duas das principais estratégias usadas para identificar as fraudes de energia são: fazer vistorias nas unidades consumidoras e realizar análises estatísticas de forma manual (ARAUJO, 2006). Dado que o processo de KDD tem como objetivo de descobrir informações úteis em bases de dados que auxiliem na tomada de decisões, é interessante executá-lo com o objetivo de ajudar as empresas concessionárias distribuidoras de energia elétrica no processo de descobrimento ágil e efetivo de usuários que estão realizando algum tipo de fraude na rede de distribuição de energia elétrica.

1.1 OBJETIVO GERAL

Aplicar técnicas de MD no histórico de consumo dos clientes das companhias de energia elétrica, visando a identificação de anomalias em redes de distribuição de energia elétrica.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste projeto são:

1. Criar um sistema detector de anomalias.
2. Avaliar o sistema criado.
3. Comparar os algoritmos utilizados.

1.3 JUSTIFICATIVA

No processo de distribuição de energia elétrica existem vários tipos de perdas de energia como, por exemplo: perdas nas linhas de transmissão, falhas nos registros, erros de medição e fraudes. De acordo com o relatório anual da companhia distribuidora de energia elétrica COPEL, no ano de 2013 as perdas não técnicas (Comerciais), que ocorrem devido a procedimentos não regulares, como por exemplo, ligações clandestinas feitas por consumidores, são equivalentes a 33% de todas as perdas na distribuição de energia, assim como ilustrado na Figura 1.

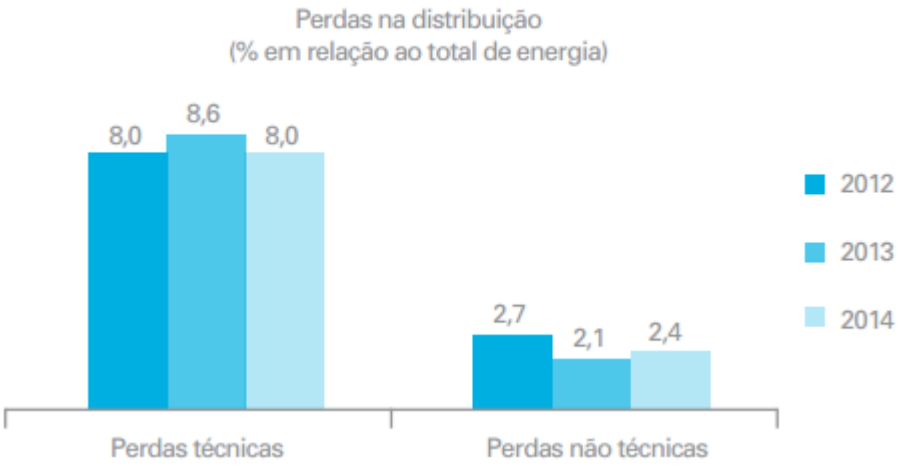

Figura 1 - Perdas na distribuição.
Fonte: Relatório de sustentabilidade (COPEL, 2014).

Além do prejuízo financeiro para as empresas de energia elétrica as ligações irregulares representam um risco para a segurança pública, pois podem causar acidentes graves e incêndios, que podem levar a morte de pessoas (DELGADO, 2010).

Algumas das estratégias utilizadas para a detecção de ligações clandestinas são: realização de inspeções nas unidades consumidoras, fiscalização de denúncias feitas por consumidores e a realização de analises estatísticas de forma manual baseada no consumo dos usuários e realização de inspeção nas unidades consumidoras. Porém essas estratégias acarretam em mais gastos para as empresas concessionárias distribuidoras de energia elétrica (ARAUJO, 2006).

Diante da grande quantidade de dados armazenados pelas empresas concessionárias distribuidoras de energia elétrica, e tendo em conta as diversas aplicações que usam MD para analises de grandes bases de dados, foi proposta uma maneira de utilizar a MD para auxiliar no processo de identificação de ligações irregulares no sistema de distribuição de energia elétrica.

1.4 ESTRUTURA DO TRABALHO

O trabalho é composto de seis capítulos, sendo que o primeiro é apresentado a introdução, objetivos gerais e específicos bem como a justificativa do trabalho.

No capítulo II é descrita a fundamentação teórica com os conceitos de KDD, MD, MVS, AD´s, Ferramenta WEKA, calibragem dos algoritmos de MD e o problema das perdas de energia elétrica.

O capitulo III descreve a metodologia proposta para auxiliar na identificação de fraudes no sistema distribuidor de energia elétrica.

No capítulo IV são descritos os materiais e métodos da pesquisa, é apresentada as ferramentas utilizadas, da base de dados inicial, descreve-se o pré-processamento e transformação dos dados e o treinamento e testes dos três algoritmos de MD.

O capítulo V apresenta os resultados obtidos nesse trabalho e também é feita uma análise dos resultados obtidos.

Por último, o capitulo VI compõem a conclusão e as considerações finais e algumas possibilidades de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capitulo são apresentados as principais definições, aplicações e características das tecnologias utilizadas durante todo o processo de extração de conhecimento.

2.1 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS (*KNOWLEDGE DISCOVERY IN DATABASE* - KDD)

O KDD é um processo de descoberta de conhecimento em bases de dados, seu principal objetivo é extrair conhecimento a partir de grandes bases de dados. Para isto ele usa conhecimento de diversas áreas, tais como: matemática, estatística, banco de dados e inteligência artificial (CASTANHELAS, 2008).

De acordo com Fayyed, Piatetsky-Shapiro e Smyth (1996 a), KDD é um processo interativo e iterativo, que envolvem vários passos com diversas decisões a serem tomadas pelo usuário. Esses vários estão ilustrados na Figura 2.

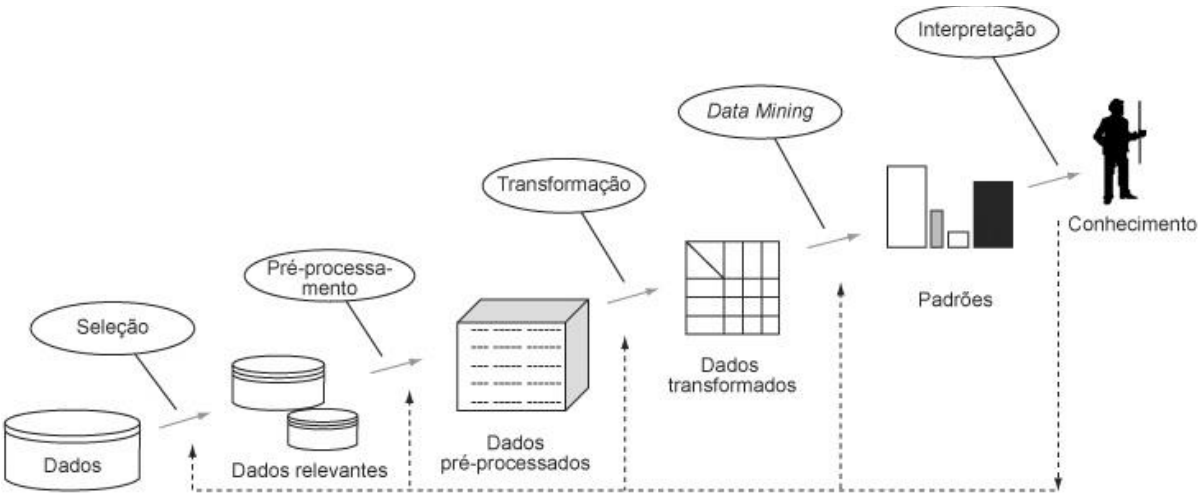

Figura 2 - Processo de KDD.
Fonte: Fayyad et al (1996 b).

Segundo Brachman e Anand (1996), os passos básicos para uma boa execução do processo de KDD podem ser descritos da seguinte maneira:

1. Identificar o objetivo do processo de KDD seguindo o ponto de vista do cliente.
2. Criar um conjunto de dados: fazer um agrupamento dos dados relevantes que sejam relacionados ao objetivo do KDD. Os dados podem ser qualitativos ou quantitativos.

3. Limpar os dados e fazer o pré-processamento: são operações básicas tais como a remoção de ruído e execução de estratégias que lidem com campos vazios. Essa etapa tem como objetivo assegurar a qualidade dos dados selecionados.
4. Transformar: os dados pré-processados passam por outra transformação, que os armazena em outra forma, com o objetivo de facilitar a execução das técnicas de MD.
5. Escolher o método para MD: é necessário escolher um método que combine com o objetivo do KDD, alguns dos possíveis métodos são: *clustering*, regressão, classificação e sumarização.
6. Escolher o algoritmo para MD: é a etapa onde se escolhe o algoritmo a ser utilizado na busca de padrões na base de dados. Essa etapa inclui decidir quais modelos e parâmetros são adequados, e combinar o algoritmo de MD com o objetivo do KDD.
7. Minerar os dados: aplicar os algoritmos de MD na base de dados em busca de padrões de interesse em uma forma representacional em forma de: regras de classificação, árvores de regressão, *clustering*, e assim por diante. O usuário pode auxiliar significativamente na eficiência do algoritmo de MD, executando corretamente as etapas anteriores.
8. Interpretar os padrões minerados, e possivelmente retornar a qualquer um dos passos 1-7 para mais uma iteração. Este passo pode também envolver a visualização dos padrões extraídos, ou a visualização dos dados extraídos através dos modelos.
9. Consolidar conhecimento descoberto, incorporando esse conhecimento a outro sistema, realizar novas ações, ou simplesmente documentá-lo e relatá-lo às partes interessadas. Isto também inclui a verificação de informações obtidas com o que se acreditava anteriormente, assim podendo resolver potenciais conflitos com o conhecimento que se acreditava anteriormente.

Cada uma das etapas do processo de KDD representa tarefas de grande importância, porém a etapa de MD pode ser estimada como a tarefa de maior relevância, dado que esta etapa permite encontrar padrões dentro das bases de dados mediante o uso de diferentes algoritmos.

2.2 MINERAÇÃO DE DADOS

Nesta sessão são fornecidos conceitos da etapa de mineração de dados dentro do processo de descoberta de conhecimento em bases de dados. São relatadas as tarefas mais usadas, citando aplicações e características.

2.2.1 INTRODUÇÃO

Existem várias definições de MD que podem ser encontradas na literatura, algumas delas são:

"MD é uma ferramenta utilizada para descobrir novas correlações, padrões e tendências entre as informações de uma empresa, através de grandes quantidades de dados armazenados em Data Warehouse usando técnicas de reconhecimento de padrões, estatística e matemática" Nimer e Spandri (1998, p 32).

"Data Mining, ou Mineração de Dados, pode ser entendido como o processo de extração de informações, sem conhecimento prévio, de um grande banco de dados e seu uso para tomada de decisões" Corrêa e Sferra (2003, p 04).

"Data Mining tem por objetivo oferecer estratégias automatizadas para a análise de grandes bases de dados de empresas e outras áreas de estudo" Araujo e Pereira (2011, p 09).

"Mineração de dados é a técnica de exploração de grandes conjuntos de dados, com o objetivo de estabelecer relações, associações e padrões de difícil visualização, transformando dados brutos em informações de alto valor" Bueno e Viana (2012, p 02).

"É o processo não-trivial de identificar em dados, padrões válidos, novos, potencialmente úteis e ultimamente compreensíveis", Fayyad, Piatetski-Shapiro e Smyth (1996 b, p 03).

Pode-se dizer que a MD consiste na existência de um algoritmo que encontra informações implícitas e úteis em uma base de dados. Essa é a fase onde dados são transformados em conhecimento. Uma vez escolhido o algoritmo que será utilizado no processo de MD, será necessário implementá-lo e adaptá-lo ao problema proposto.

A MD se diferencia das técnicas estatísticas porque ao invés de procurar padrões hipotéticos, ela faz uso dos próprios dados para descobrir estes padrões. De acordo com Thearling et al. 1999, cerca 5% de todas as relações podem ser encontradas por métodos estatísticos. As técnicas de MD podem descobrir as outras relações que não seriam descobertas, os 95% restantes.

Várias ferramentas distintas, como redes neurais artificiais (RNA), árvores de decisão (AD), Maquinas de Vetores de Suporte (MVS), Regressão Linear, Regressão Logística, *Clustering*, Apriori, K-means, programas estatísticos, tanto isoladamente, quanto em combinação, podem ser aplicados durante a realização do processo de MD.

É mostrado na pratica que nenhum algoritmo é o mais eficiente em todos os objetivos das tarefas de MD (Cohavi et al, 1997). Por isso é comum realizar o processo de MD com mais de um algoritmo. Neste trabalho são utilizados algoritmos de MVS e as AD´s.

De acordo com Navega (2002), é possível distinguir dois tipos de objetivos para a MD:

a) Verificação, onde o sistema é limitado a verificar as hipóteses do usuário;
b) Descobrimento, onde o sistema de forma automática encontra novos padrões.

Ainda é possível subdividir o objetivo de descobrimento em duas partes:

a) Predição, onde o sistema encontra padrões com o objetivo de poder predizer futuros comportamentos de algumas entidades;
b) Descrição, onde o sistema busca padrões com o objetivo de apresenta-os de maneira compreensível ao ser humano.

2.3 PRINCIPAIS TAREFAS DA MINERAÇÃO DE DADOS

A MD vem sendo desenvolvida visando à solução de problemas de diversas áreas, logo suas tarefas também são diversificadas. Essas tarefas são classificadas de acordo com sua capacidade de resolver problemas de determinadas áreas.

Estas tarefas podem descobrir diferentes tipos de conhecimento, portanto, faz-se necessário determinar o tipo de conhecimento que o algoritmo deve extrair no início do processo de MD.

2.3.1 CLASSIFICAÇÃO

A classificação é uma função de aprendizado que visa identificar a qual classe um determinado registro pertence. Nesta tarefa, o modelo analisa dados de entrada, ou conjuntos de registros fornecidos, com cada registro já contendo à qual classe pertence, com o propósito de aprender como classificar um novo registro. (CASTANHELAS, 2008)

Em outras palavras o objetivo de um algoritmo de classificação é encontrar alguma correlação entre os atributos e uma classe, para possibilitar que o processo de classificação esses atributos para prever a classe de um exemplo novo e desconhecido.

São exemplos de tarefas de classificação: separar pedidos de créditos em alto, médio e baixo risco ou determinar quando uma transação de um cartão de credito pode ser uma fraude.

2.3.2 REGRESSÃO

A regressão também é conhecida por predição funcional, predição de valor real, função de aproximação, ou ainda, aprendizado de classes contínuas. Sua tarefa é similar a de classificação, a principal diferença é que o atributo a ser identificado é um valor numérico e não um categórico. Essa tarefa possibilita estimar o valor de uma determinada variável analisando os valores das demais (CAMILO e SILVA, 2009).

Alguns exemplos de regressão são: estimar a pressão ideal de um paciente se baseando na pressão, idade, massa corporal e sexo, ou estimar o aumento no gasto mensal de uma família com duas crianças no período de volta as aulas.

De acordo com Indurkhya e Weiss (1999), nas áreas MD e aprendizagem de maquinas, a maior parte das pesquisas é voltada para problemas de classificação, pois é menos comum encontrar problemas de regressão na vida real.

2.3.3 AGRUPAMENTO

Agrupamento é um processo que visa à partição de um grupo de dados heterogênea em vários subgrupos mais homogêneos. No agrupamento, não existem classes pré-definidas ou valores a serem estimados (AGRAWAL e SRIKANT, 1994).

Normalmente a tarefa de agrupamento é realizada antes de alguma outra forma de MD. Por exemplo, em uma aplicação de oferta de credito bancário, pode-se primeiro dividir os clientes em grupos que tenham comportamento similar, para posteriormente aplicar uma classificação.

2.3.4 REGRAS DE ASSOCIAÇÃO

A tarefa de associação consiste em identificar o quanto a presença de um conjunto de itens nos registros de uma base de dados implica na presença de outro conjunto de itens nos mesmos registros (AGRAWAL e SRIKANT, 1994). O objetivo das regras de associação é encontrar tendências na base de dados. Por exemplo, observando as vendas de um supermercado, é constatado que 85% dos consumidores que compram os produtos X e Y também adquirem, na mesma compra o produto Z.

Regras de associação são muito usadas em mercados para planejar ofertas e a distribuição dos produtos nas prateleiras, de modo que os produtos próximos ente si chamem a atenção dos clientes.

2.3.5 DESVIO

A tarefa de desvio tem o objetivo de encontrar conjunto que não segue o padrão dos demais dados. Para poder executar essa tarefa é necessário adotar padrões antecipadamente (CASTANHELAS, 2008).

Essa tarefa pode ser usada para identificar vários tipos de fraudes, baseando-se em elementos que fogem dos padrões ou são exceção as regras.

2.3.6 CONCLUSÃO

No processo de MD, para obter o melhor resultado é muito importante identificar qual tarefa se enquadra melhor no objetivo do processo de MD. Como o objetivo é identificar um usuário de acordo com seu histórico de consumo, aparentemente a tarefa que se enquadra na base de dados seria REGRESSÃO. Porém, após serem executadas as etapas de pré-processamento e transformação dos dados, cada consumo de cada cliente que se encontram em forma numérica serão usados para gerar uma nova base de dados, com o código do cliente e quatro atributos descritivos. De acordo com essa nova base de dados a tarefa mais adequada para ser utilizada no processo de MD é a de classificação.

A partir da escolha feita, serão estudadas ferramentas, métodos e algoritmos apropriados, visando concluir o objetivo deste trabalho. De acordo com resultados obtidos,

serão estudadas as AD's e MVS para classificação no processo de MD. Os estudos das técnicas, algoritmos e base de dados estão nos capítulos seguintes.

2.4 MÁQUINAS DE VETORES DE SUPORTE

As MVS conseguem resolver problemas de regressão e classificação através de sua capacidade de generalização. MVS são capazes de separar instancias das suas classes a partir de uma função, que é obtida a partir dos dados usados na etapa de treinamento. Seu objetivo é criar um classificador que funcione adequadamente em dados que não foram usados em seu treinamento, desta maneira sendo capaz de predizer as saídas de futuras entradas de dados (JUNIOR, 2010).

De acordo com RUSSEL e NORVING (1994), existem três propriedades que tornam as MVS atraentes:

1. As MVS constroem um separador de margem máxima com um limite de decisão com maior distância possível a pontos de exemplo. Isso os ajuda a generalizar futuras entradas de dados.
2. As MVS criam uma separação linear em hiperplano (separam as classes que estão em um plano cartesiano utilizando uma reta), mas tem a capacidade de adicionar um ou mais espaços de dimensão, usando um algoritmo de *kernel*.
3. As MVS são um método não paramétrico, elas mantêm exemplos de treinamento e podem precisar armazenar todos eles. Mas na pratica, acabam mantendo uma pequena porção do número de exemplos. Assim as MVS têm flexibilidade para representar funções complexas, e são resistentes a super adaptação.

Uma MVS cria sua função (classificador) a partir dos padrões encontrados na sua base de dados de treinamento. Considerando a ilustração da Figura 3, existem classificadores lineares que separam as duas classes, porém apenas uma que possui a mesma distância para os elementos de ambas as classes, o hiperplano com margem máxima recebe o nome de hiperplano ótimo (GUNN, 1998). Os hiperplanos são definidos por um conjunto de pontos, **x:w * x** + b = 0 (RUSSELL e NORVING, 1994).

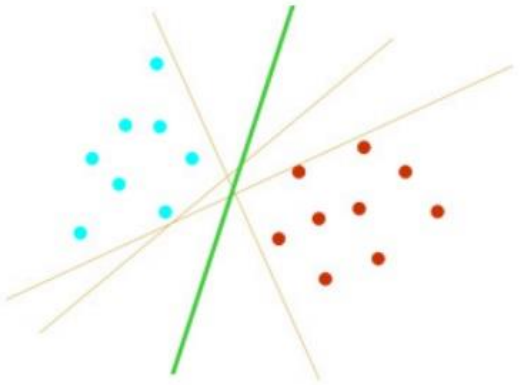

Figura 3 - Possíveis hiperplanos de separação e hiperplano ótimo.
Fonte: (OLIVEIRA JUNIOR, 2010).

Porém existem casos em que é impossível separar as classes usando apenas uma reta, nestes casos são necessários executar funções de maior complexidade para tentar contornar o problema.

Essas são as funções de *Kernel*, que tem a finalidade de projetar os vetores de características de entrada em um espaço de características de alta dimensão para a classificação dos problemas que se encontram em espaços não linearmente separáveis. Pois a medida que se aumenta o espaço da dimensão do problema, também aumenta a probabilidade do problema se tornar linearmente separável. Entretanto, para alcançar uma boa distribuição é necessário um conjunto de treinamento com uma quantidade elevada de instâncias (GONÇAVES, 2011).

A Figura 4 ilustra o processo de transformação de um domínio não linearmente separável, em um domínio linearmente separável através do aumento da dimensão onde é feito do mapeamento.

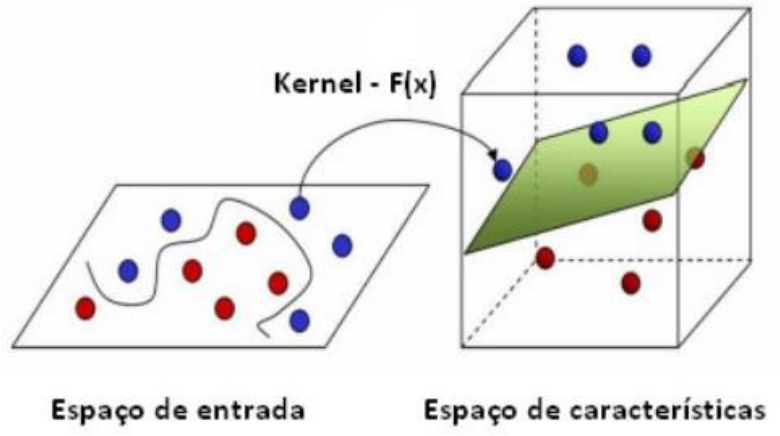

Figura 4 - Processo de transformação de um domínio linearmente não separável.
Fonte: (OLIVEIRA JUNIOR, 2010).

A representação alternativa para casos não separáveis linearmente pode ser chamada de representação dual, alguns exemplos de funções de *Kernel* usados em algoritmos de MVS são: *Kernel* Polinomial, *Kernel* Linear, *Radial Basis Functiopn*, entre outros (RUSSELL e NORVING, 1994).

2.5 ÁRVORES DE DECISÃO

Uma AD representa uma função que toma como entrada um vetor com valores de atributos e retorna um valor de saída único (RUSSELL e NORVING, 1994). As AD consistem em nós que representam os atributos; em arcos provenientes destes nós que recebem os possíveis valores para estes atributos; e de nós folhas que representam as diferentes classes de um conjunto de treinamento (INGARGIOLA, 1996).

No contexto da MD, AD são algoritmos de classificação de dados, que alcançam sua decisão executando uma sequência de testes. Os algoritmos de aprendizagem que utilizam as AD usam uma estratégia chamada dividir-para-conquistar, em outras palavras, um problema complexo é divididos em vários subproblemas menores, que podem ser resolvidos de maneira recursiva (LEMOS; NIMEOLAS; STEINER, 2005).

Existem vários algoritmos que utilizam a estratégia de AD, os principais são: *UserClassifier*, *SimpleCart*, *RandomTree*, *DecisionStump* e J48. Nesse projeto é usado o algoritmo J48, que surgiu da necessidade de recodificar o algoritmo C4.5, que foi implementado na linguagem C, para JAVA (QUINLAN, 1993). O J48 gera um modelo de árvore de decisão baseada em uma base de dados de treinamento, e usa esse modelo para classificar futuras entradas de dados.

A utilização de algoritmos de AD apresenta as seguintes vantagens: não assumem nenhuma distribuição particular para os dados; os dados podem ser categóricos (qualitativos) ou numéricos (quantitativos); podem-se construir modelos para qualquer função desde que o conjunto de treinamento tenha um grande número de exemplos; e possua um elevado grau de compreensão. (LEMOS; NIMEOLAS; STEINER, 2005)

A Figura 5 ilustra um exemplo de AD, na qual os dados relatam as condições para uma pessoa receber um empréstimo. Nesse caso existem duas classes: Sim (receber empréstimo) e Não (não receber empréstimo). Os atributos são montante, salário e conta. O atributo montante pode assumir os valores de médio, alto ou baixo; o atributo salário pode assumir qualquer valor do tipo inteiro; e o atributo conta pode ser sim ou não. A classificação,

nesse caso, resulta numa estrutura de árvore, que pode ser usada para todos os objetos do conjunto (BRADZIL, 1999).

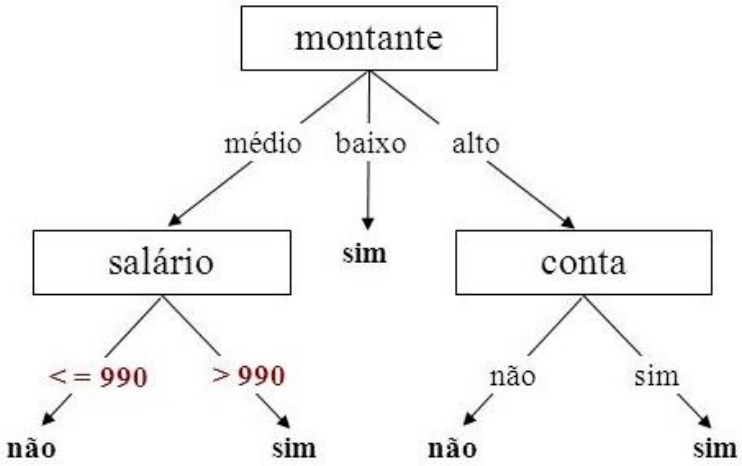

Figura 5 - Exemplo de uma árvore de decisão.
Fonte: (LEMOS; NIMEOLAS; STEINER, 2005).

Após o treinamento de um algoritmo de MD sempre é importante avaliá-lo. No caso das AD não é diferente, está avaliação consiste em utilizar a árvore de decisão em um conjunto de dados que não tenha sido usado no seu treinamento. Esta estratégia permite julgar como a árvore se adapta a novas situações, além de revelar os erros e acertos ocorridos na construção da árvore (BRADZIL, 1999).

2.6 WEKA

Para o gerenciamento eficiente dos algoritmos de mineração de dados pode ser utilizado o *software* WEKA (Waikato *Enviroment for Knowledge Analysis*). Este *software* foi desenvolvido usando linguagem de programação Java na universidade de Waikato, Nova Zelândia. O WEKA gerencia uma grande variedade de algoritmos de aprendizado de máquina e técnicas que agregam algoritmos de diferentes paradigmas da área de inteligência artificial (HALL, 2015). É por isto que o WEKA foi escolhido como a ferramenta a ser utilizada no treino e validação dos algoritmos de MD utilizados na resolução do problema estudado.

2.6.1 ESTRUTURA DO WEKA

A tela inicial do WEKA solicita ao usuário a escolha de uma interface, entre quatro possíveis, cada uma delas possui características e objetivos específicos.

a) *Explorer* - proporciona um ambiente gráfico para seleção de um conjunto de dados e aplicação de filtros e algoritmos para MD. É uma interface simples de usar, que possui diversos algoritmos que podem ser utilizados. Essa interface impede que o usuário faça escolhas não aplicáveis, e apresenta informações sobre os preenchimentos de seus campos (HALL, 2015).

b) *Experimenter* – é um ambiente gráfico que permite testar técnicas diferentes em classificação ou regressão, com o objetivo de compará-las. Isso também é possível usando tela *Explorer* e *KnowledgeFlow*, porém na tela *Experimenter* é possível escolher vários conjuntos de dados e varias técnicas a serem executadas em um experimento (REUTMAN e SCUSE, 2007).

c) *KnowledgeFlow* – permite o desenvolvimento de projetos de MD em um ambiente gráficos com fluxo de informações. Suas principais vantagens são *layout* intuitivo, componentes pode ser selecionados usando uma *tool bar*, permite processamento em *batch*, permite aplicação de conjuntos de dados de elevada dimensão e processamento paralelo (HALL e REUTMAN, 2008).

d) *Simple CLI* – é uma interface de linha de comandos onde é possível executar os comandos do WEKA. Usando está interface é possível utilizar todas as funcionalidades, porém requer um conhecimento sobre os comandos a serem utilizados (HALL, 2015).

2.6.2 INTERFACE EXPLORER

A interface mais usada nesse projeto é a *Explorer*, essa interface permite executar tarefas que estão divididas em seis separadores, segundo FRANK, HALL e WITTEN (2011) essas tarefas podem ser descritas da seguinte maneira:

1. *Preprocess*: aqui é definida a base de dados a ser usada, também é permitido aplicar uma grande quantidade de filtros sobre os dados.
2. *Classify*: seu objetivo é classificar algo, primeiramente é necessário escolher um classificador e configurá-lo, depois se deve escolher o modo de treino. O WEKA oferece 4 modos de teste: *Use Training set*, o método é treinado e testado com todos os dados disponíveis; *Supplied Test set*, é possível escolher uma parte da base dados

para treinar o método e toda a base de dados é usada para testar o método; *Cross-Validation*: a base de dados é dividida em N partes, uma para treinar e N-1 para testar, esse processo é repetido N vezes usando uma parte diferente para teste; *Percentage split*, é definido uma porcentagem dos dados para treinar o método e o resto é usado para testá-lo. Depois de o método ser treinado e testado o WEKA mostra um relatório com estatísticas sobre o desempenho do método.

3. *Clustering*: seu funcionamento é similar a interface de Classificação, é necessário escolher um cluster, configurá-lo e escolher um modo de treino. Esse modo também mostra um relatório detalhado sobre o desempenho do método.
4. *Associate*: permite aplicar associações orientadas entre métodos de pesquisa de dados. Nessa interface só é necessário escolher um método e configurá-lo, após sua execução é possível ver um relatório.
5. *Select Atributes*: o objetivo dessa interface é determinar quais atributos possuem mais peso em determinar se os dados são de um tipo ou de outro. Primeiramente faz-se necessário selecionar o método de avaliação de atributos, esse método é responsável por avaliar cada um dos atributos e dizer seu peso. O segundo passo consiste em escolher o método responsável por gerar o espaço de testes, selecionar o método de ensaio e o atributo que representa a classificação conhecida.
6. *Visualize*: Esse modo mostra a distribuição dos dados graficamente. Esse modo permite visualizar correlações e associações entre os atributos de uma forma gráfica.

2.6.3 INTERFACE EXPERIMENTER

O ambiente de *Experimenter* do Weka permite ao usuário criar, executar, modificar, e analisar experiências de uma forma mais conveniente do que é possível quando processando os algoritmos individualmente. Por exemplo, o usuário pode criar uma espécie de rotina que executa vários esquemas contra uma série de conjuntos de dados e, em seguida, analisar os resultados para determinar se um dos esquemas é (estatisticamente) melhor do que os outros esquemas (BOUCKAERT et all, 2010).

Ao se iniciar um experimento simples na interface *Experimenter*, é necessário realizar os seguintes passos.

- Selecionar o destino do arquivo com os resultados do experimento, por padrão é gerado um arquivo no formato .ARFF, mas também é possível escolher entre o formato .CSV ou salvar em uma banco de dados JDBC;

- O tipo do experimento, por padrão é feito um experimento do tipo *Cross Validation*, mas também é permito dividir uma porcentagem dos dados para treinamento e outra para teste;
- Escolher a tarefa da MD, é possível escolher entre Classificação ou Regressão;
- Adicionar um ou mais conjuntos de dados;
- O número de repetições, que por padrão é 10;
- A prioridade de execução, por padrão a prioridade é das bases de dados, mas pode ser dada aos algoritmos. Isso é útil quando se tem mais de uma base de dados e algoritmos configurados, e se quer que o algoritmo seja aplicado a uma base de dados o mais rápido possível; e
- Configurar os algoritmos, pode ser adicionada um ou mais algoritmos de diferentes tipos, e calibrados de forma diferente. Cada um desses algoritmos é aplicado a cada uma das bases de dados.

Após configurado é possível que de fato se inicie o experimento, durante a execução é possível acompanhar o *status* do experimento. Caso tudo ocorra bem é possível acessar o arquivo gerado pelo Weka com os resultados do teste.

2.7 CALIBRAGEM DOS ALGORITMOS

Uma questão muito importante do processo de MD de dados é a calibragens dos algoritmos usados. O algoritmo j48 possui 17 atributos que podem ser modificados, na Tabela 1 são exibidos os parâmetros associados ao algoritmo J48, a descrição e valor padrão dos mesmos.

Parâmetro	**Descrição**	**Valor padrão**
Seed	A semente usada para randomizar os dados quando a poda de erros reduzida é usada.	1
Unpruned	Se a poda é realizada.	False
ConfidenceFactor	O fator de confiança utilizado para a poda (valores menores implica em mais poda).	0,25
NumFolds	Determina a quantidade de dados utilizados para a poda de erros reduzida. Uma dobra é usada para a poda, o resto para o crescimento da árvore.	3
Numdecimalplaces	O número de casas decimais a ser usado para a saída de números no modelo.	2
Batchsize	O número preferido de instâncias para processar se a previsão lote está sendo realizada. Mais ou menos casos pode ser proporcionado, mas isto dá implementações a oportunidade de especificar um tamanho de lote preferida.	100
ReducedErrorPruning	Se a poda de erros reduzido é utilizado em vez de poda	False

	C.4.5.	
UseLaplace	Se a contagem em folhas é suavizada usando a correção Laplace.	False
DoNotMakeSplitPointActualValue	Se for verdade, o ponto de divisão não é transferido para um valor de dados reais. Isso pode aumentar a velocidade do algoritmo.	False
Debug	Se definido como verdadeiro, classificador pode saída informações adicionais para o console.	False
SubtreeRaising	Se deve considerar a operação de sub elevação quando a poda.	True
SaveInstanceData	Se salvar os dados de treinamento para visualização.	False
BinarySplits	Se usar divisões binárias em atributos nominais ao construir a árvore.	False
DoNotCheckCapabilities	Se for definido, as capacidades de classificadores não são verificadas antes do classificador ser construído.	False
MinNumObj	O número mínimo de casos por folha.	2
UseMDLcorrection	Se a correção MDL é utilizado quando encontrar racha nos atributos numéricos.	True
CollapseTree	Remover as peças que não reduzem o erro de treinamento.	True

Tabela 1 - Parâmetros do algoritmo J48
Fonte: Adaptado de (SEPULVEDA, 2016).

Na Tabela 2 estão ilustrados os nomes dos parâmetros que devem ser calibrados ao usar o algoritmo de MSV LibSVM, suas descrições e valores padrão.

Parâmetros	Descrição	Valor padrão
SVM Type	O tipo de SVM a ser usada.	C-SVC (Classification)
Batch Size	O número preferido de exemplos para processar se a previsão do lote estiver a ser realizada. Mais ou menos casos pode ser proporcionado, mas isto dá implementações a oportunidade de especificar um tamanho de lote preferida.	100
Cache Size	O tamanho do cache em MB.	40
Coef0	O coeficiente usado para calcular a margem do separador.	0
Cost	O parâmetro C custo para C-SVC, epsilon-SVR e nu-SVR.	1
Debug	Se definido como verdadeiro, classificador pode exibir informações adicionais no console.	False
Degree	O grau do Kernel.	3
Do Not Check Capabilities	Se definido, as capacidades de classificadores não são verificadas antes do classificador ser construído (Deve ser usado com cuidado para reduzir o tempo de execução).	False
Do Not Replace Missing Values	Desativa reposição automática de valores em falta. AVISO: definido como verdadeiro somente se os dados não contêm campos sem valores.	False
Eps	Tolerância do critério de terminação (Não dever ser modificado).	0,001
Gamma	O gama a ser usado, deve variar entre 0 e 1.	0,0
Kernel Type	O tipo de kernel para usar	Radial Bassis Function
Loss	O epsilon para a função de perda de epsilon-SVR.	0,1
Model File	A pasta interna onde é salvo o modelo LIBSVM	Weka 3.8
Normalize	Deve normalizar os dados.	False

Nu	O valor de nu para nu-SVC, SVM uma classe e nu-SVR.	0,5
Num Decimal Places	O número de casas decimais a ser usado para a saída de números no modelo.	2
Seed	O número aleatório para ser utilizado.	False
Probability Estimates	Se necessário, é usado para gerar estimativas de probabilidade em problemas de classificação, deve ter o valor 1 ou -1.	1
Shrinking	Se pretende utilizar a heurística Shrinking.	True
Weights	Os pesos de usar para as classes (lista em branco-separados, por exemplo, "1 1 1" para um problema com 3 classes), se for vazio é usado 1 por padrão.	Vazio

Tabela 2 - Descrição e valor padrão dos atributos do LibSVM.
Fonte: Adaptado de (Bouckaert Et all, 2010).

Os atributos do algoritmo SMO são muito similares ao LibSVM, pois os dois são algoritmos de MVS. A tabela 3 ilustra a lista de atributos usados pelo SMO, sua descrição e valor padrão.

Parâmetros	Descrição	Valor padrão
Batch Size	O número preferido de exemplos para processar se a previsão do lote estiver a ser realizada. Isto dá implementações a oportunidade de especificar um tamanho de lote preferida.	100
Build Calibration Models	Ajustar modelos de calibração para as saídas do SVM (Para estimativas de probabilidade próprias).	False
C	O parâmetro complexidade C.	1
Calibrator	O método de calibração a ser usado. Permite que seja usado atributos de outros algoritmos de MVS em algumas estimativas.	Logistic
Checks Turned Off	Se ativado, desativa as verificações que consomem muito tempo.	False
Debug	Se definido como verdadeiro, o classificador pode fornecer informações adicionais no console.	False
Do Not Check Capabilities	Se ativado, as capacidades dos classificadores não são verificadas antes do classificador ser construído (Deve ser usado com cuidado para reduzir o tempo de execução).	False
Epsilon	O epsilon de erro de arredondamento (não deve ser alterado).	1.0E-12
Filter Type	Determina como e se os dados serão transformados.	Normalize Training Data
Kernel	O kernel a ser usado.	Poly Kernel
Num Decimal Places	O número de casas decimais a ser usado para a saída de números no modelo.	2
Num Folds	O número de dobras para validação cruzada, é usado para gerar dados de treinamento para modelos de calibração (-1 significa que os dados de treinamento serão usados para validação).	-1
Random Seed	Aleatória semente número para a validação cruzada.	1
Tolerance Parameter	O parâmetro de tolerância (não deve ser alterado).	0,001

Tabela 3 - Descrição e valor padrão dos parâmetros do algoritmo SMO.
Fonte: Adaptado de (SEPULVEDA, 2016).

Como pode ser visto nos quadros e figuras acima, cada algoritmo de MD possui diversos atributos a serem calibrados para se obter bons resultados. Foram executados mais de 50 testes em cada algoritmo, alternando as configurações dos atributos. Os primeiros testes foram realizados com os valores padrões dados pelo Weka, e de acordo com o desempenho de cada teste os atributos foram sendo modificados.

2.8 PROBLEMA DE PERDAS E FRAUDES NAS RDEE'S

As perdas de energia elétrica constituem um grave problema que acontece em vários países, de acordo com Departamento de Monitoramento do Sistema Elétrico (DMSE), no Brasil, no ano de 2015, 17,80% de toda energia elétrica inserida nas redes de distribuição foi perdida. A Figura 6 ilustra um gráfico que representa o consumo de energia elétrica de todo o Brasil dividido em cinco classes.

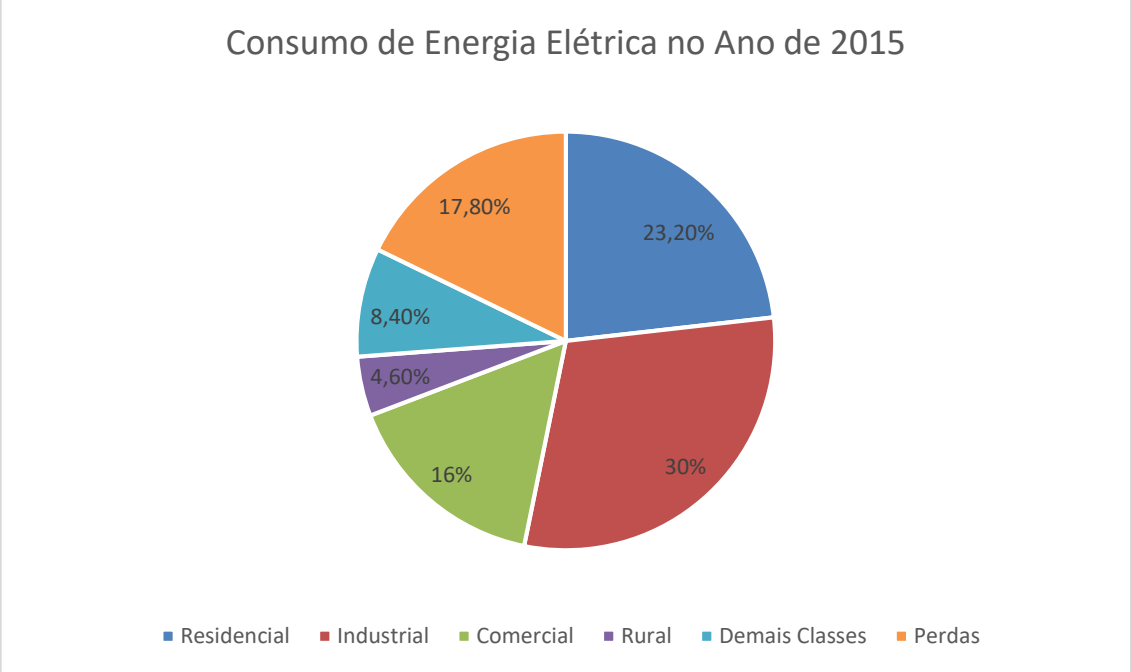

Figura 6 - Consumo de energia elétrica do ano de 2015.
Fonte: adaptado do (boletim mensal da DMSE, janeiro de 2016).

Como já se infere pelo próprio nome, perdas de energia elétrica é diferença negativa entre à energia consumida pelos usuários que é registrada pelos padrões e a energia inserida no sistema distribuidor (Copel, 2014). Existem dois tipos de perdas, as técnicas, que acontecem naturalmente e estão relacionadas ao próprio sistema de distribuição de energia elétrica. Enquanto as perdas técnicas estão ligadas as características dos sistemas de

distribuição, comerciais estão ligadas principalmente a anomalias nas leituras, inadimplência, fraudes ou problemas nos medidores (ARAUJO, 2006). A Figura 7 ilustra as subdivisões dos tipos de perdas de energia elétrica.

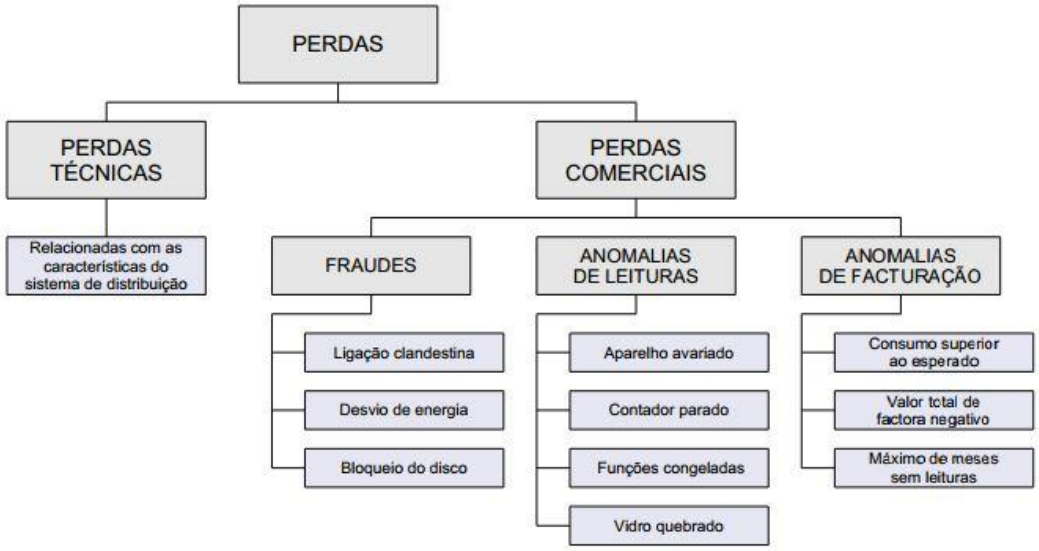

Figura 7 - Tipos de perdas.
Fonte: (FRANCO; LÓPES; QUEIROZ, 2015).

Em geral a maior preocupação em relação as perdas comerciais estão relacionadas a duas modalidades de fraudes, as ligações clandestinas e as modificações nos medidores. As ligações clandestinas são reconhecidas pelo desvio de energia feito através de ligações ilegais por um consumidor, o que faz com que a energia usada não seja contabilizada, resultando em perdas. No caso das modificações nos medidores, o consumidor possui registro, porém faz alterações no aparelho responsável por registrar a energia usada, fazendo com que parte da energia consumida por sua residência/comercio/indústria não seja contabilizada, pagando por menos energia do que de fato foi usada (DELGADO, 2010).

No estado do Espírito Santo, na área de concessão da companhia EDP Escelsa, foi calculado uma perda comercial de cerca de 17% no ano de 2013, esse prejuízo pode chegar a cerca de R$ 48 milhões ao ano. Já em Minas Gerais, a CEMIG registrou uma perda de 11,28% da energia injetada no sistema. O prejuízo causado atinge aproximadamente R$ 950 milhões ao ano.

A Agência Nacional de Energia Elétrica (ANEEL) tem autorizado as distribuidoras de energia elétrica em todo o país a aumentarem as tarifas cobradas nas faturas de energia. Em São Paulo, os consumidores da AES Eletropaulo, viram em julho de 2015 suas contas aumentarem em 15,23%. O que torna a situação um pouco pior para os consumidores é que

em março de 2015 já havia ocorrido uma revisão extra, que elevou em 31,9% as contas de luz dos clientes da referida concessionária.

Uma das preocupações de especialistas do setor e empresas de distribuição é de que o aumento acelerado gere dificuldade para alguns consumidores no pagamento de suas contas, resultando no aumento da inadimplência, e talvez no aumento das perdas relacionadas a fraudes e furtos (SEPULVEDA, 2016).

3 METODOLOGIA DA SOLUÇÃO PROPOSTA

Com o objetivo de melhorar a etapa de pré-processamento e o resultado da mineração de dados, foram feitos estudos na base de dados a fim de selecionar um subconjunto de dados, contendo o que realmente seja necessário. O conjunto de dados foi definido em um modelo que está divido em três conjuntos: informações do registro do usuário, informações sobre o histórico dos usuários e informações sobre as anomalias de leituras encontradas. Para atender as necessidades da análise, foi necessário considerar as seguintes suposições: um só tipo de usuário (com consumo inferior a 130 kW); usuários com um histórico de medições de no mínimo 12 meses; usuários com todas as medições maiores que zero; e usuários com medições consideradas corretas (sem erro de leitura).

Propõem-se o uso de um modelo de extração de conhecimento apresentado por FRANCO, LÓPES e QUEIROZ, no ano de 2015, que avalia as características dos clientes que são identificados pelo atributo matricula (MAT) e, a partir delas, a criação de regras que serão aplicadas sobre os dados dos clientes. Foram criadas quatro regras que avaliam os usuários de acordo com seus históricos de consumo. Estas regras são:

- Classificação do consumo (CLA_CON);
- Risco (RIS);
- Oscilação (OSC);
- Anomalias (ANOM)

Essas regras são processadas de acordo com parâmetros de referência que indicam qual a variação esperada para a classificação dos clientes. A seguir são apresentadas as definições das regras anteriores.

Regra CLA_CON: Os usuários são classificados de acordo com a energia consumida, em consumo baixo (CB), em consumo médio (CM), em consumo alto (CA), onde, C_{Min} é o menor consumo do usuário, C_{Max} é o maior consumo e $C(j)$ é o consumo do mês atual. A estrutura de consumo para CB, CM e CA são apresentadas nas expressões (1), (2) e (3) respectivamente.

$$Cmin \leq C(j) \leq \frac{Cmax - Cmin}{3} \quad (1)$$

$$\frac{Cmax - Cmin}{3} < C(j) \leq 2\frac{Cmax - Cmin}{3} \quad (2)$$

$$2\frac{Cmax - Cmin}{3} < C(j) \leq Cmax \quad (3)$$

Regra RIS: Usando um índice de risco, é determinada a possibilidade de fraude de acordo com a comparação do consumo do mês atual, *j*, e a média dos três messes anteriores. Para as três primeiras faturas o índice de risco é igual a zero, caso contrário:

$$IndRis(j) = \frac{C(j) - \frac{(C(j-3))+(C(j-2))+(C(j-1))}{3}}{\frac{(C(j-3))+(C(j-2))+(C(j-1))}{3}} * C(j) \qquad (4)$$

Fazendo o uso do índice de risco é determinado o nível de risco que cada cliente tem de estar cometendo fraude como: alto risco (AR), médio risco (MR), baixo risco (BR) e sem risco (SR), de acordo com as expressões (5) (6) (7) e (8) respectivamente:

$$IndRis(j) < \left(-0.2 * C(j)\right) \qquad (5)$$

$$\left(-0.2 * C(j)\right) \leq IndRis(j) \leq 0 \qquad (6)$$

$$0 \leq IndRis(j) \leq (C(j) * 0.2) \qquad (7)$$

$$\left(0.2 * C(j)\right) \leq IndRis(j) \qquad (8)$$

Regra OSC: Usando o índice de oscilação, é determinada a possibilidade de fraude de acordo com a comparação da medição de consumo do mês atual, j, e a medição do mês anterior, j-1, levando em conta que para o primeiro mês de consumo do usuário o risco de oscilação é igual a zero, caso contrario.

$$IndOsc(j) = \frac{C(j) - C(j-1)}{C(j-1)} * C(j) \qquad (9)$$

Fazendo uso do índice de oscilação é classifica o usuário de acordo com o tipo de variação no consumo como de oscilação descendente (OsD), oscilação normal (OsN) e oscilação ascendente (OsA), de acordo com as expressões (10), (11) e (12) respectivamente.

$$IndOsc(j) < \left(-0.15 * C(j)\right) \qquad (10)$$

$$(-0.15 * C(j)) \leq IndOsc(j) \leq \left(0.2 * C(j)\right) \qquad (11)$$

$$\left(0.2 * C(j)\right) \leq IndOsc(j) \qquad (12)$$

Regra ANOM: Esta regra é utilizada para determinar que medições são suspeitas e poderiam indicar usuários que devem ser inspecionados. A classificação correspondente a esta regra é determinada através do uso das regras CLA_CON, RIS e OSC. Com está regra podem ser determinados os valores abaixo.

Inspecionar (Ins): Se CLA_CON é CB, RIS é AR e OSC é OsD, ou, se CLA_CON é CB, RIS é MR e OSC é OsD, então é preciso inspecionar.

Não Inspecionar (NIsn): Em caso contrário aos descritos acima.

4 MATERIAIS E MÉTODOS

Este trabalho propõe uma maneira de identificar fraudes na rede de distribuição de energia elétrica através da análise dos dados relacionados ao consumo dos clientes da empresa distribuidora de energia elétrica, diminuindo o tempo e o dinheiro gasto no combate às fraudes.

Neste capitulo apresenta-se uma descrição detalhada sobre a base de dados usada e as ações necessárias para realização das etapas de pré-processamento, transformação e mineração dos dados bem como seus resultados.

4.1 FERRAMENTAS UTILIZADAS

Para o desenvolvimento desse trabalho foram usadas as seguintes ferramentas e tecnologias:

- Linguagem de programação JAVA;
- Sistema operacional Windows;
- Ambiente de desenvolvimento integrado, IDE Eclipse;
- Weka;
- Apache Open Office;
- Astah Community;
- Office Word;

O sistema operacional escolhido foi o Windows, pois o mesmo permite controlar a IDE Eclipse e o software Weka através do uso de interfaces simples e execuções por linha de comando.

A ferramenta Open Office, uma vez que além de ser um software livre, ele permite a edição de documentos no formato de planilhas, tornando mais simples o gerenciamento e manipulação da base de dados.

A IDE Eclipse foi utilizada no desenvolvimento de um software para facilitar a etapa de limpeza e pré-processamento dos dados, ela está na versão Mars.2, sendo uma das mais atuais lançadas. Em conjunto com essa IDE foi utilizado o JAVA na versão 1.8.

A ferramenta Astah foi usada na modelagem do diagrama de classes apresentado nesse trabalho. E o software Word do pacote Office, foi usado para escrever este documento.

4.2 BASE DE DADOS INICIAL

A base de dados utilizada nesse trabalho contém dados reais, disponibilizados por uma companhia distribuidora de energia elétrica. Os dados inicialmente vieram divididos em duas planilhas no formato Excel, uma com os dados respectivos aos anos de 2009 e 2010, outra com os dados dos anos de 2011 até 2013. As duas planilhas estão em padrões diferentes e continham dados utilizados pela empresa distribuidora de energia elétrica que não serão necessários na neste estudo.

O arquivo que contém os dados de 2009 e 2010, contém 72.779 registros. Além do consumo de cada cliente essa planilha contém as seguintes informações, que são usadas pela companhia de distribuidora de energia elétrica: digito de checagem, o número da conta, o código do departamento e do município a qual o cliente se encaixa, dois códigos relativos a atividades realizadas pelo cliente, classificação de serviço, nó e grupo de qualidade a qual o cliente pertence.

A planilha com os dados dos anos de 2011, 2012 e 2013, contém os registros de 462.435 clientes. Suas colunas além de conter os consumos desde janeiro de 2011 até junho de 2013, contém os seguintes campos: código do usuário, código do alimentador e do transformador usado pelo cliente, as fazes utilizadas pelo cliente, e uma classificação do tipo de consumo do cliente.

As Tabelas 4 e 5 ilustram as primeiras linhas das duas planilhas que foram recebidas da empresa distribuidora de energia elétrica, e que foram utilizadas para montar a base de dados deste projeto.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1													2009					
2	MATRICULA	CHEQUEO	CUENTA	DEP	MUNICIPIO	UB	ESTRATO	SIE	CIIU	CLASE	NODO	CALIDAD	1	2	3	4	5	6
3	101102	229	101102229	66	683	U	2	1A	11	FM	D23516	2	116	112	131	151	129	145
4	101103	6	101103006	66	683	U	3	1A	11	FM	D23516	2	150	145	177	201	212	245
5	101104	893	101104893	66	683	U	2	1A	11	FM	D23516	2	0	0	0	0	0	3
6	101105	570	101105570	66	683	U	2	1A	11	FM	D23516	2	14	122	118	124	105	87
7	101106	357	101106357	66	683	U	3	1A	11	FM	D23516	2	106	96	96	85	96	118
8	101110	534	101110534	66	683	U	1	1A	11	FM	D23516	2	120	116	118	120	117	117
9	101112	108	101112108	66	683	U	2	1A	11	FM	D23516	2	100	91	69	55	46	45
10	101113	985	101113985	66	683	U	1	1A	11	FM	D23516	2	112	98	49	118	120	116
11	101114	762	101114762	66	683	U	2	1A	11	FM	D23516	2	53	54	50	50	52	53
12	101118	890	101118890	66	683	U	1	1A	11	FM	D23516	2	82	74	88	87	85	79
13	101119	677	101119677	66	683	U	2	1A	11	FM	D23516	2	103	89	100	102	107	115
14	101158	486	101158486	66	683	U	1	1A	11	FM	D20330	2	97	106	112	0	0	144
15	101167	578	101167578	66	683	U	2	1A	11	FM	D23516	2	38	41	39	31	7	18
16	101170	978	101170978	66	683	U	3	1A	11	FM	D23516	2	91	79	81	16	24	31
17	101171	755	101171755	66	683	U	3	1A	11	FM	D23516	2	161	157	152	144	152	158
18	101172	532	101172532	66	683	U	2	1A	11	FM	D23516	2	330	398	430	474	448	475
19	101173	319	101173319	66	683	U	2	1A	11	FM	D23516	2	84	87	76	77	68	68
20	101183	298	101183298	66	683	U	3	1A	11	FM	D20299	2	239	269	297	257	279	272
21	101184	75	101184075	66	683	U	3	1A	11	FM	D20299	2	124	109	85	115	101	125
22	101190	716	101190716	66	683	U	3	1A	11	FM	D20299	2	175	161	176	175	172	225
23	101191	503	101191503	66	683	U	3	1A	11	FM	D20299	2	95	96	103	101	93	106
24	101192	380	101192380	66	683	U	3	1A	11	FM	D20299	2	176	154	173	202	149	175
25	101193	167	101193167	66	683	U	3	1A	11	FM	D20299	2	100	89	79	95	104	0

Tabela 4 - Dados dos anos de 2009 e 2010.
Fonte: Autoria Própria.

	A	B	C	D	E	F	G	H	I	J	K
1	USUARIO	ALIMENTADOR	TRANSFORMADOR	CONEXIO	CLIENTE	2011					
2	U_CODIGO	A_CODIGO	T_CODIGO	FASES	TIPO	1	2	3	4	5	6
3	362448307	AGU23L12	N40003	ABN	RS3	254	266	229	191	222	224
4	344772937	AGU23L12	N40003	BN	RS3	46	51	54	55	51	63
5	338096133	AGU23L12	N40003	AN	RS2	82	93	93	73	88	73
6	340132043	AGU23L12	N40003	AB	RS3	56	13	91	73	88	76
7	340273283	AGU23L12	N40003	AB	RS3	48	42	126	135	131	99
8	342046091	AGU23L12	N40003	AB	RS3	42	40	93	105	133	126
9	344771150	AGU23L12	N40003	AB	RS3	111	78	70	66	63	67
10	342912319	AGU23L12	N40003	BN	RS3	126	62	98	83	0	8
11	342911532	AGU23L12	N40003	AN	CR6	0	70	80	36	18	11
12	319996571	AGU23L12	N40003	AN	RS3	29	37	31	41	36	52
13	319997358	AGU23L12	N40003	AN	RS3	174	173	163	138	149	148
14	319999912	AGU23L12	N40003	AN	RS2	219	203	221	191	131	272
15	337432970	AGU23L12	N40003	ABN	RS2	0	133	150	111	123	131
16	320183701	AGU23L12	N40003	ABN	CR6	2	37	65	49	54	56
17	320184588	AGU23L12	N40003	ABN	RS3	200	209	178	174	197	176
18	320182914	AGU23L12	N40003	BN	RS3	75	58	78	61	24	15
19	320185285	AGU23L12	N40003	AN	RS3	127	108	109	97	87	73
20	320186042	AGU23L12	N40003	BN	RS3	21	3	11	1	1	1
21	320187829	AGU23L12	N40003	ABN	CR6	0	45	53	44	44	44
22	320188606	AGU23L12	N40003	AN	RS3	65	41	42	45	51	56
23	320189493	AGU23L12	N40003	ABN	CR6	0	52	41	30	39	41
24	320190229	AGU23L12	N40003	BN	RS2	15	63	53	46	53	53
25	320000301	AGU23L12	N40003	ABN	CR6	0	54	72	65	68	68

Tabela 5 - Dados do período de 2011 a 2013.
Fonte: Autoria Própria**.**

4.3 PRÉ-PROCESSAMENTO E TRANSFORMAÇÃO

Para realizar as etapas de pré-processamento e transformação dos dados foi implementado um *software* utilizando a linguagem de programação JAVA. Esse software lê a base de dados que se encontra em duas planilhas que devem estar no formato CSV, processa os dados, aplica as regras do modelo de extração e gera dois resultados: o primeiro é a junção dos dados que se encontram separados em duas planilhas, e o segundo é a base de dados que será usado na etapa de MD.

Como está ilustrado na Figura 8, o *software* que realiza as etapas de pré-processamento e transformação da base de dados e é dividido em seis classes.

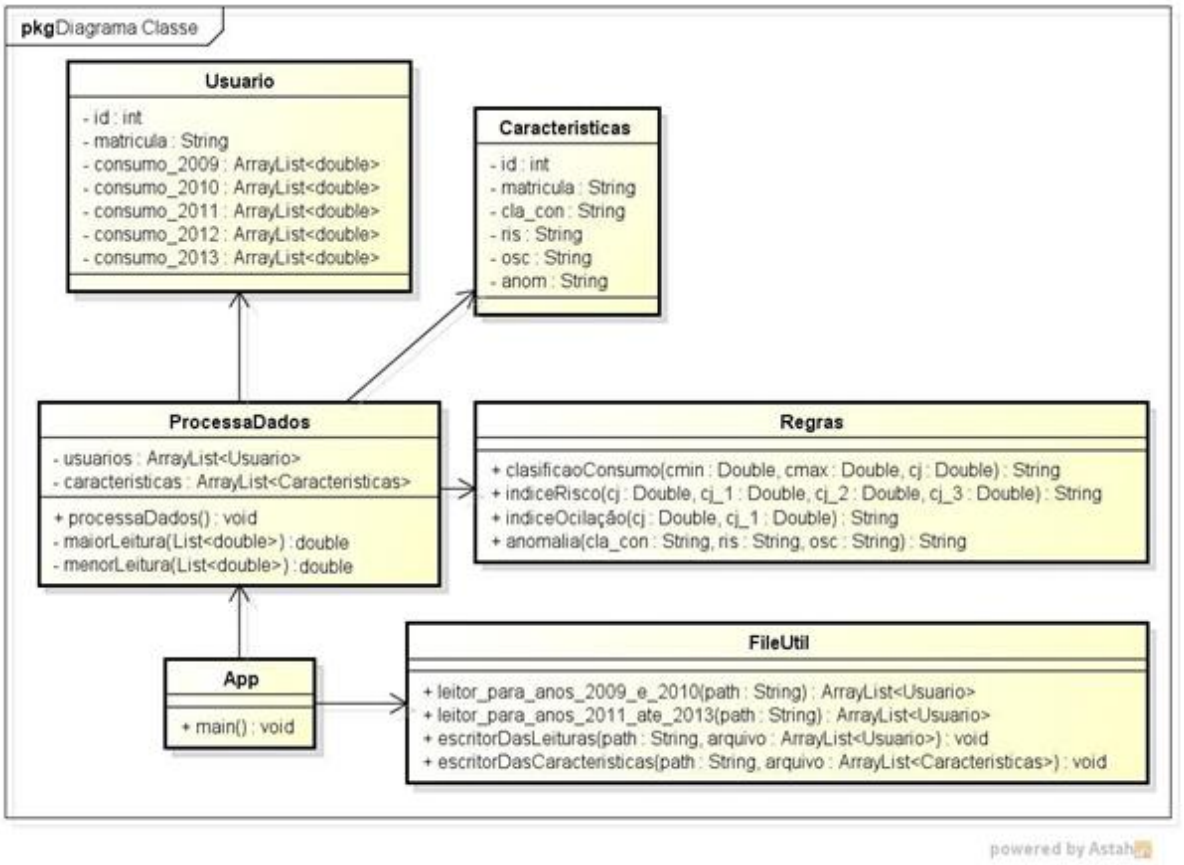

Figura 8 - Diagrama de classes do software.
Fonte: Autoria própria.

Os parágrafos abaixo descrevem os objetivos e características que motivaram a criação de cada uma das classes.

A classe Usuário é utilizada para manter todos os consumos de cada um dos usuários da Companhia Distribuidora de Energia Elétrica, ela possui sete atributos: um número único que é usado como identificador pelo sistema, a matricula do usuário e cinco listas, cada uma delas é usada para manter o consumo respectivo a um ano entre 2009 e 2013.

A classe Características tem o propósito de guardar os resultados da execução das regras que avaliam o usuário de acordo com o seu consumo, ela é composta por 6 atributos: Numero identificador, matricula do usuário, classificação do consumo, risco, oscilação e anomalia.

Na classe Regras é implementado as quatro regras que classificam o usuário de acordo com seu histórico de consumo, ela possui quatro métodos, cada um responsável por executar o cálculo de uma regra e retornar o resultado, os métodos são respectivamente responsáveis por aplicar a regra de classificação do consumo, índice de risco, índice de oscilação e por último a regra de anomalia.

A classe FileUtil é responsável pela conversão da base de dados que se encontra no formato CSV, em uma lista de objetos Java do tipo Usuário. Ela também é responsável por criar dois arquivos, um no formato de texto, contendo todas as leituras de cada usuário considerado valido, e outro no formato de uma planilha .CSV, contendo as características de cada consumo de todos os Usuários.

A classe ProcessaDados é necessária para guardar as informações que vem da base de dados, e os resultados gerados após a aplicação das regras definidas, possibilitando que esses dados sejam escritos no disco rígido do computador e usados nas outras etapas do processo de mineração de dados.

A classe principal do projeto é chamada de App, sua função é chamar os métodos das outras classes em sua devida ordem, e manter um *log* informações sobre o andamento das operações.

A execução do *software* gera dois resultados finais, o primeiro deles é uma planilha que se encontra no formato .CSV, ela contém o histórico de leituras do subconjunto de usuários validos que serão utilizados na etapa de mineração de dados. Essa planilha é usada para ter uma visão diferenciada dos dados que compõem a base de dados usada no processo de MD. O primeiro resultado é ilustra na Tabela 6.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		2009												2010
2	MATRICULA	1	2	3	4	5	6	7	8	9	10	11	12	1
3	101112108	100	91	69	55	46	45	48	53	56	51	26	26	79
4	101113985	112	98	49	118	120	116	116	122	108	107	109	121	101
5	101114762	53	54	50	50	52	53	61	68	62	47	78	74	55
6	101118890	82	74	88	87	85	79	83	87	84	85	92	86	93
7	101167578	38	41	39	31	7	18	11	21	23	27	17	24	23
8	101170978	91	79	81	16	24	31	42	65	84	93	16	60	61
9	101173319	84	87	76	77	68	68	72	70	71	74	71	81	83
10	101191503	95	96	103	101	93	106	109	107	98	106	99	102	108
11	101194944	46	44	56	50	49	81	59	42	43	42	68	61	47
12	101200422	53	51	56	45	39	47	56	64	62	59	62	51	60
13	101204650	81	71	76	75	68	84	72	80	75	71	68	8	70
14	101205337	87	76	85	85	82	85	80	84	81	80	82	81	75
15	101207901	62	55	55	56	60	57	52	51	48	48	50	55	53
16	101226962	108	117	108	100	94	94	93	118	97	87	102	7	111
17	101228526	63	58	60	59	59	57	59	60	49	73	69	67	63
18	101231926	70	68	70	65	63	67	66	70	63	64	71	78	70
19	101251774	17	62	77	73	69	72	68	48	18	28	24	49	32
20	101259020	69	65	55	64	65	72	69	70	66	73	80	9	75
21	101260866	18	18	19	19	21	22	14	25	20	21	18	16	17
22	101273186	51	35	44	42	43	33	30	32	32	36	56	52	62
23	101280604	21	15	22	20	20	19	17	19	17	17	30	35	28
24	101285519	84	87	92	88	86	85	91	93	75	78	94	89	85
25	101288960	92	78	91	70	12	70	79	83	86	87	85	88	89
26	101296275	28	23	28	25	23	21	22	23	17	17	23	41	39
27	101309322	90	81	77	70	96	97	93	108	103	100	91	84	76
28	101316850	96	101	114	91	83	84	77	101	77	64	84	49	92
29	101320037	112	120	107	106	99	99	109	116	103	105	89	76	67
30	101323488	93	86	92	95	88	93	83	90	82	90	93	89	96
31	101328393	121	112	108	102	57	77	106	14	50	71	110	100	79
32	101329170	92	86	101	95	99	88	100	121	97	113	108	19	72
33	101330906	119	96	111	106	124	129	107	98	97	93	92	102	90
34	101341662	78	79	80	79	84	80	99	101	87	81	93	103	92
35	101343226	83	79	83	82	82	87	95	93	83	81	86	84	85

Tabela 6 - Histórico de consumo dos usuários após pré-processamento.
Fonte: Autoria Própria.

O segundo resultado é a nova base de dados a ser utilizada no trabalho, ela também se encontra no formato .CSV e possui as seguintes colunas:

- Um código único para cada leitura mensal dos clientes, este código é composto da matricula do usuário, a abreviação do ano e do mês que fatura representa, como por exemplo, 192586659_12_1 onde 192586659 é a matricula do usuário, 12 é a abreviação do ano da fatura (2012), e o número 1 que representa o mês (Janeiro).
- Outro código único composto apenas de um número inteiro.
- O resultado da Classificação do Consumo (CLA_CON).
- O resultado do índice de Risco (RIS).
- O resultado da regra de Oscilação (OSC).
- E o resultado da regra de Anomalia (ANOM).

O segundo resultado pode ser visto na Tabela 7. Os dados contidos nessa planilha são usados nos processos de treinamento e testes dos algoritmos de MD.

	A	B	C	D	E	F
1	MATRICULA	CODIGO	CLA_CON	RIS	OSC	ANOM
2	101167578_12_1	13121	CA	BR	OsN	NAO
3	101167578_12_2	13122	CA	MR	OsD	NAO
4	101167578_12_3	13123	CA	AR	OsN	NAO
5	101167578_12_4	13124	CA	MR	OsN	NAO
6	101167578_12_5	13125	CA	BR	OsN	NAO
7	101167578_12_6	13126	CA	MR	OsN	NAO
8	101167578_12_7	13127	CM	MR	OsN	NAO
9	101167578_12_8	13128	CA	BR	OsA	NAO
10	101167578_12_9	13129	CA	BR	OsN	NAO
11	101167578_12_10	131210	CM	AR	OsD	NAO
12	101167578_12_11	131211	CB	AR	OsD	SIM
13	101167578_12_12	131212	CB	AR	OsD	SIM
14	101167578_13_1	13131	CB	AR	OsA	NAO
15	101167578_13_2	13132	CB	MR	OsD	SIM
16	101167578_13_3	13133	CB	MR	OsN	NAO
17	101167578_13_4	13134	CB	MR	OsN	NAO
18	101167578_13_5	13135	CM	SR	OsA	NAO
19	101167578_13_6	13136	CA	SR	OsA	NAO
20	101170978_09_4	14094	CB	AR	OsD	SIM
21	101170978_09_5	14095	CB	AR	OsA	NAO
22	101170978_09_6	14096	CB	AR	OsA	NAO
23	101170978_09_7	14097	CM	SR	OsA	NAO
24	101170978_09_8	14098	CM	SR	OsA	NAO
25	101170978_09_9	14099	CA	SR	OsA	NAO
26	101170978_09_10	140910	CA	SR	OsN	NAO
27	101170978_09_11	140911	CB	AR	OsD	SIM
28	101170978_09_12	140912	CM	MR	OsA	NAO
29	101170978_10_1	14101	CM	BR	OsN	NAO
30	101170978_10_2	14102	CM	BR	OsD	NAO
31	101170978_10_3	14103	CB	AR	OsD	SIM
32	101170978_10_4	14104	CB	AR	OsN	NAO
33	101170978_10_5	14105	CB	AR	OsN	NAO
34	101170978_10_6	14106	CM	SR	OsA	NAO
35	101170978_10_7	14107	CM	SR	OsN	NAO

Tabela 7 - Base de dados.
Fonte: Autoria Própria.

4.4 TREINAMENTO E TESTE DOS ALGORITMOS DE MD

Esse subtítulo descreve os processos desenvolvidos durante o treinamento dos algoritmos de MD.

Foi realizado o treinamento e testes de três algoritmos de MD, o primeiro deles é um algoritmo de AD conhecido como J48, o segundo e terceiro foram dois algoritmos de MVS, o primeiro deles foi o algoritmo LibSVM e o segundo foi o algoritmo SMO. Nessa etapa foram utilizadas as interfaces *Explorer* e *Experimenter* da ferramenta Weka.

Foram usadas duas abas usadas da interface *Explorer*, a primeira aba é chamada de *Pre-process* e tem vários usos, entre eles: visualizar algumas características dos dados, carregar, salvar e aplicar filtros na base de dados. A segunda usada se chama *Classify*, esse é o local onde os algoritmos são treinados e testados individualmente, possibilitando que sejam vistos os resultados, parâmetros de execução sejam modificados e modelos sejam salvos ou carregados.

A etapa de mineração de dados foi repedida várias vezes para cada algoritmo, aplicando filtros nos dados, modificando atributos dos algoritmos, alternando a tipos dos

dados e removendo as colunas com os códigos dos dados, porém era gasto muito tempo pois cada iteração era feita de maneira manual. Após observar como os resultados das MD se comportavam de acordo com as diferentes configurações, e com o objetivo de diminuir o tempo ocioso gasto entre cada teste, foi resolvido fazer uso da interface *Experimenter*.

Fazendo uso dessa interface foi possível executar o treinamento e teste do algoritmos de MD de forma dinâmica, quando após a escolha de uma ou mais base de dados, foi possível criar uma fila de algoritmos configurados de forma diferente a ser executado um após o outro, para posteriormente gerar uma planilha com os resultados e características da execução de cada algoritmo. Essa funcionalidade foi muito usada durantes o treinamento dos algoritmos LibSVM e SMO, que devido a sua grande quantidade de atributos tinham muitas possibilidades de configurações.

A aba *Setup* é a primeira tela da interface *Experimenter*, essa é a aba responsável pela configuração de uma rotina de treinamentos e testes, onde é escolhida a base de dados e vários algoritmos configurados de forma diferentes.

Após configurada a rotina é o momento de executa-la, a aba *Run* é responsável por executar a rotina e manter um *feedback* de sua execução. Caso ocorra qualquer problema essa aba possui um sistema de log que avisa o usuário sobre qualquer problema que resulte no cancelamento da rotina, caso não ocorra nenhum problema essa aba possibilita que visualizemos o andamento da rotina.

Quando o Weka acaba de executar o treinamento e testes dos os algoritmos colocados na fila da rotina, eles o salvam em uma planilha que possibilita o uso da última aba da interface *Experimenter*, essa aba é chama de *Analyse*, ela serve para fazermos comparações entre os resultados obtido durante o treino e teste de cada algoritmo da rotina.

Porém essa não é a única maneira de analisar os resultados gerados nessa interface, também é possível acessar diretamente a planilha gerada após a execução da rotina, ela permite que analisemos 60 atributos obtidos na execução de cada *fold* de cada algoritmo treinado e testado, algumas dessas informações são: Quantidade de classificações corretas e incorretas, tempos gasto no treinamento e teste, *true positives*, *true negatives*, *false positives*, *false negatives*, instancias de treino e teste.

5 RESULTADOS

Neste capitulo é apresentado dados sobre o desempenho e resultados obtidos após a aplicação do processo de MD.

5.1 APLICANDO O ALGORITMO J48 USANDO A REGRA ANOM

Ao aplicar o algoritmo J48 usando a regra ANOM, foi obtido a matriz de confusão exibida na Tabela 9 e a AD da Figura 9. A partir da matriz de confusão é possível observar que, todas as 3.242.541 instâncias do total de 3.242.541 foram classificadas corretamente, para os valores SIM (Possui anomalias) e NÃO (Não possui anomalias), foram classificados corretamente 100.392 e 3.142.149 (100% do total dos dados) respectivamente. De acordo com as taxas de acertos obtidas pelo modelo, pode-se afirmar que há confiabilidade “ALTA” nesta classificação.

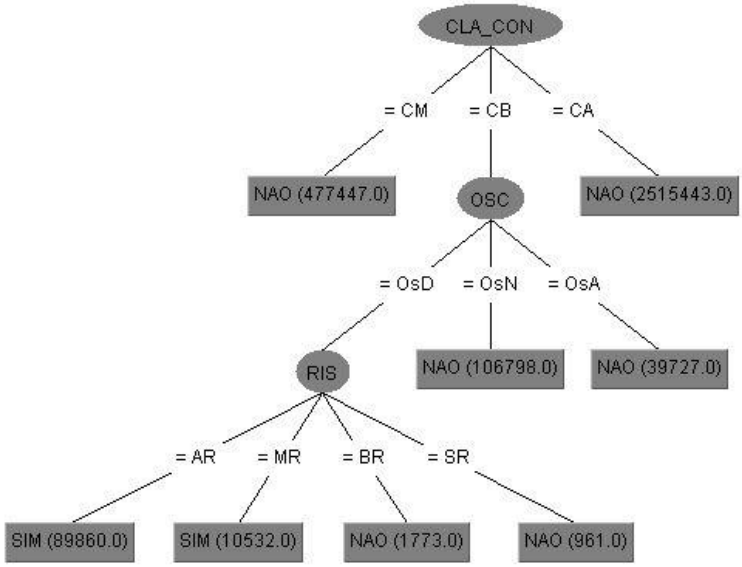

Figura 9 - AD do algoritmo J48 aplicado a regra ANOM
Fonte: Autoria Própria.

	Não	Sim
Não	3142149	0
Sim	0	100392

Tabela 8 - Matriz de Confusão do algoritmo J48 aplicado a regra ANOM
Fonte: Autoria Própria.

Observando a AD ilustrada na Figura 9 é possível fazer as seguintes afirmações:

- Se CLA_CON tem o valor CB e OSC tem o valor OsD e RIS tem o valor AR, então a classificação indica que 89.860 classificações possuem fraudes.
- Se CLA_CON tem o valor CB e OSC tem o valor OsD e RIS tem o valor MR, então a classificação indica que 10.532 classificações possuem fraudes.
- Se CLA_CON tem o valor CB e OSC tem o valor OsD e RIS tem o valor BR, então a classificação indica que 1.773 classificações não possuem fraudes.
- Se CLA_CON tem o valor CB e OSC tem o valor OsD e RIS tem o valor AR, então a classificação indica que 89.860 classificações possuem fraudes.
- Se CLA_CON tem o valor CB e OSC tem o valor OsD e RIS tem o valor SR, então a classificação indica que 961 classificações não possuem fraudes.
- Se CLA_CON tem o valor CB e OSC tem o valor OsN, então a classificação indica que 106.798 classificações não possuem fraudes.
- Se CLA_CON tem o valor CB e OSC tem o valor OsA, então a classificação indica que 39.727 classificações não possuem fraudes.
- Se CLA_CON tem o valor CB e OSC tem o valor OsA, então a classificação indica que 39.727 classificações não possuem fraudes.
- Se CLA_CON tem o valor CM, então a classificação indica que 447.447 classificações não possuem fraudes.
- Se CLA_CON tem o valor CA, então a classificação indica que 2.515.443 classificações não possuem fraudes.

5.2 APLICANDO OS ALGORITMOS DE MVS USANDO A REGRA ANOM

Os treinamentos realizados empregando as MVS permitiram modificações em diversos parâmetros. No algoritmo LibSVM, alguns dos atributos calibrados foram: o tipo de MVS, custo, degrau, Kernel, entre outros. E no algoritmo SMO foram: Kernel, Complexidade C, Filtro dos dados, entre outros. Porém as alterações dos parâmetros não mudaram a taxa de acerto, houve mudança apenas nos tempos gastos durante o treinamento e testes de cada modelo, e no tamanho final dos modelos.

A porcentagem de acerto obtida na criação dos modelos com aplicação das MVS foi de 100%. Tanto no algoritmo LibSVM quanto no SMO não foi encontrado nenhum padrão

diferente ao mudar as configurações do algoritmo. As Tabelas 10 e 11 exibem os resultados obtidos no processo de criação e testes do algoritmo LibSVM e SMO respectivamente.

Kernel	Degre	Coef	Custo	Instancias de treinamento	Instancias de Teste	Tempo de treinamento em segundos	Tempo de teste em segundos	Acertos
RBF	3	0	1	2.918.287	324.254	71,402	3,869	100%
Linear	3	0	1	2.918.287	324.254	25,928	2,652	100%
Polynomial	3	0	1	2.918.287	324.254	71,542	6,583	100%
Sigmoid	3	0	1	2.918.287	324.254	77,61	4,649	100%
Linear	3	0	5	2.918.287	324.254	24,638	2,467	100%
Linear	3	0	15	2.918.287	324.254	27,586	2,358	100%
Polynomial	3	0	2	2.918.287	324.254	52,245	4,196	100%
Polynomial	3	0	3	2.918.287	324.254	45,926	3,432	100%
Polynomial	3	0	4	2.918.287	324.254	42,339	3,276	100%
Polynomial	3	2	3	2.918.287	324.254	62,166	2,48	100%
Polynomial	3	4	5	2.918.287	324.254	96,83	2,371	100%
Polynomial	1	0	1	2.918.287	324.254	41,333	3,26	100%
Polynomial	3	0	1	2.918.287	324.254	114,598	8,518	100%
Polynomial	5	0	1	2.918.287	324.254	535,269	44,304	100%
Polynomial	1	0	1	2.918.287	324.254	40,065	3,151	100%
Polynomial	5	0	1	2.918.287	324.254	535,269	44,304	100%
Sigmoid	3	0	10	2.918.287	324.255	27,023	1,966	100%
Sigmoid	3	0	30	2.918.287	324.254	55,084	2,824	100%
RBF	3	10	1	2.918.287	324254	69,185	2,517	100%
RBF	3	30	1	2.918.287	324254	66,6725	2,665	100%
RBF	10	0	1	2.918.287	324254	66,595	3,11	100%
RBF	20	0	1	2.918.287	324254	68,0475	2,775	100%
RBF	15	8	10	2.918.287	324254	66,2675	2,605	100%
Linear	3	0	25	2.918.287	324.254	25,334	2,469	100%
Linear	3	0	50	2.918.287	324.254	30,807	2,201	100%

Tabela 9 - Comparação dos resultados do algoritmo LibSVM.
Fonte: Autoria Própria.

Comple-xidade C	Kernel	Filtro	Instancias de Treino	Instancias de Teste	Tempo de treino em segundos	Tempo de teste em segundos	Acertos
0,1	Puk	Normalize	2.918.286	324.255	467,517	14,82	100%
0,2	Polynomial	Normalize	2.918.286	324.255	40,902	11,156	100%
0,5	Normalized Poly Kernel	Normalize	2.918.286	324.255	225,557	14,277	100%
0,8	Normalized poly Kernel	Normalize	2.918.286	324.255	302,345	16,99	100%
2	Normalized poly Kernel	Normalize	2.918.286	324.255	167,412	13,843	100%
0,7	Polynomial	Normalize	2.918.286	324.255	20,484	10,109	100%

1	Poly Kernel	Normalize	2.918.286	324.255	20,576	10,483	100%
1	Normalized poly Kernel	Normalize	2.918.286	324.255	280,254	6,615	100%
1	Puk	Normalize	2.918.286	324.255	386,631	18,018	100%
1	RBF Kernel	Normalize	2.918.286	324.255	1201,842	32,947	100%
1	RBF Kernel	Standardize	2.918.286	324.255	1229,471	31,808	100%
1,5	Puk	Normalize	2.918.286	324.255	431,699	18,767	100%
2	Normalized Poly Kernel	Normalize	2.918.286	324.255	171,977	14,71	100%
3	RBF Kernel	Normalize	2.918.286	324.255	801311	22,464	100%
6	RBF Kernel	Normalize	2.918.286	324.255	634531	23,276	100%
6	RBF Kernel	Standardize	2.918.286	324.255	870,451	22,51	100%
8	Puk	Normalize	2.918.286	324.255	420,296	18,579	100%
8	RBF Kernel	Standardize	2.918.286	324.255	1265,443	33,322	100%
10	RBF Kernel	Normalize	2.918.286	324.255	1010,335	23,884	100%
20	Normalized Poly Kernel	Normalize	2.918.286	324.255	167,412	13,843	100%
20	Normalized poly Kernel	Standardize	2.918.286	382.255	189,372	15,526	100%
20	RBF Kernel	Normalize	2.918.286	324.255	1129,676	33,57	100%
50	Polynomial	Normalize	2.918.286	324.255	20,745	10,904	100%

Tabela 10 - Comparação dos resultados do algoritmo SMO
Fonte: Autoria Própria.

Em relação aos algoritmos de MVS, como já era esperado, o atributo que tem mais influência sob o resultado é o *Kernel*, atributo responsável pela formula a ser usada aumentar a dimensão do problema a ser solucionado. Em relação ao algoritmo LibSVM observa-se que o *Kernel* mais veloz é o Linear, e o mais lento é o *Polynomial*. Já no algoritmo SMO, o *Kernel* mais veloz é o *Polynomial*, enquanto o mais lento é o *Radial Basis Function* (RBF).

Também podemos dizer que em média, os resultados do algoritmo LibSVM possui o treinamento e teste mais veloz quando comparado aos resultados do SMO, porém se for observar o melhor caso de cada um, o algoritmo SMO é quem possui o melhor tempo de treinamento do modelo, mas seu melhor tempo de treinamento é muito inferior que o do algoritmo LibSVM resultado. Em outras palavras, podemos afirmar que o *Kernel Polynomial* aplicado em conjunto com o algoritmo SMO possui baixo tempo de treino, mas é lento ao testar o modelo, em contrapartida, o *Kernel* Linear quando aplicado em conjunto com o algoritmo LibSVM gasta bastante tempo no treinamento, porém é mais veloz que todos os outros *Kernel's* no momento de testar o modelo criado.

5.3 ANALISE DOS RESULTADOS

Tendo em mão os modelos criados durante o desenvolvimento deste trabalho, e os dados dos consumos dos clientes subdivido pelo ano do consumo, foi feita uma comparação com os modelos dos algoritmos treinados para encontrar anomalias (Regra ANOM) aos dados dos anos de 2009, 2010, 2011, 2012 e 2013 separadamente. O resultado dessa comparação é exibido na tabela 12.

	Modelo J48		Modelo LibSVM		Modelo SMO	
	Tempo Gasto	Acertos	Tempo Gasto	Acertos	Tempo Gasto	Acertos
Criação do Modelo	1.23 segs.	100%	2.55 segs.	100%	1.71 segs.	100%
Classificação dos dados de 2009	33.07 segs.	100%	0.91 segs.	100%	32.01 segs.	100%
Classificação dos dados de 2010	49.87 segs.	100%	0.58 segs.	100%	50.6 segs.	100%
Classificação dos dados de 2011	309.31 segs.	100%	3.1 segs.	100%	259.69 segs.	100%
Classificação dos dados de 2012	383.51 segs.	100%	4.02 segs.	100%	339.43 segs.	100%
Classificação dos dados de 2013	176.71 segs.	100%	3.48 segs.	100%	168.7 segs.	100%

Tabela 11 - Comparação entre os modelos J48, LibSVM e SMO.
Fonte: Autoria Própria.

Observando a tabela acima, é possível afirmar que todos os algoritmos usados conseguem identificar com 100% de certeza os usuários que devem ser inspecionados por suspeita de anomalias. Porém o algoritmo LibSVM tem o melhor desempenho. Mesmo sendo o mais lento ao criar o modelo, ele é muito mais veloz na hora de classificar os dados.

A gráfico da Figura 10 ilustra a curva de consumo de um consumidor que os algoritmos classificaram como sem anomalias. Observa-se que a linha de consumo do período de 2009 a 2013 se comportam de forma similar.

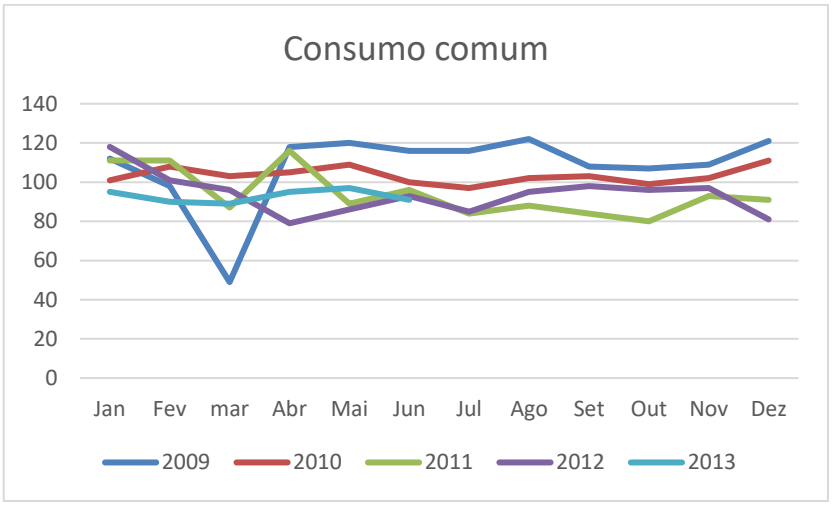

Figura 10 - Gráfico com as linhas de consumo de usuário sem anomalias
Fonte: Autoria Própria.

Já o gráfico da Figura 11, ilustra a curva de consumo de um consumidor que os algoritmos classificaram como usuário com anomalia. Nota-se que o padrão das curvas se comportam de uma maneira regular até o mês de janeiro de 2010, depois há três meses de queda até que o consumo aumente de forma incomum no mês de maio de 2010, continua com o consumo baixo, mas de forma mais regular até o mês de outubro de 2011, onde há outra queda brusca, o consumo se regularizar novamente, tendo o consumo baixíssimo, com algumas faturas muito próxima de zero.

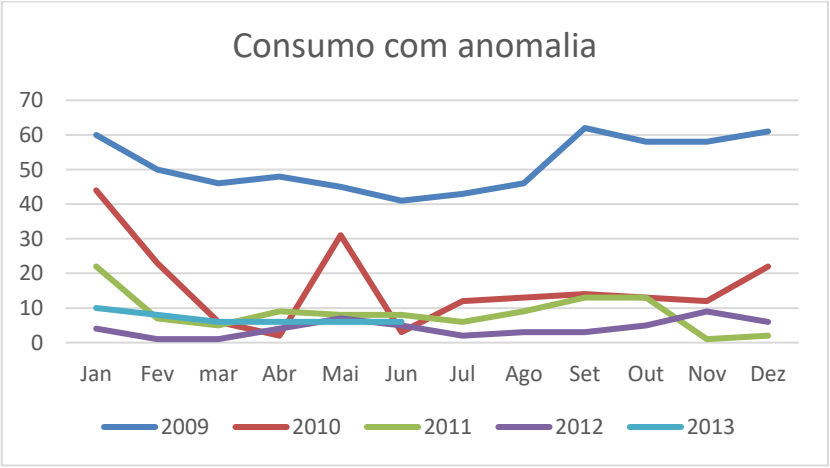

Figura 11 - Gráfico de curva de consumo com anomalia
Fonte: Autoria Própria.

Vale lembrar que o padrão das curvas de usuários com anomalia, ou não, variam de acordo com cada consumidor. Levando isso em consideração, ao usar as ferramentas corretas, é possível detectar os padrões nas curvas de consumo que podem representar algum tipo de fraude ou furto nos sistemas de distribuição de energia elétrica.

6 CONSIDERAÇÕES FINAIS

Com base no estudo realizado e nos resultados obtidos, é possível chegar a algumas conclusões e sugestões para a continuação do trabalho.

6.1 CONCLUSÃO

Por meio do estudo do TDD, foi possível conhecer maneiras de executar a tarefa de MD de acordo com o processo de KDD. Foi visto que as etapas de pré-processamento e transformação da base de dados são muito importantes e difíceis de serem realizadas, e que por meio do desenvolvimento de um software, foi possível automatizar o processo, resultando em ganho de tempo e garantia de que o processo foi realizado de forma otimizada.

Que a aplicação dos dados pré-processados a metodologia proposta neste trabalho teve efeito positivo na etapa da MD, as taxas de 100% de acertos só foram alcançadas devido a simplicidade da base de dados usada pelos algoritmos de MD. Se não fosse aplicada a metodologia, a complexidade do problema seria maior e provavelmente os resultados seriam ruins.

Na medida em que os algoritmos foram criados e testados, se obteve grande intimidade com a ferramenta Weka. Inicialmente seria usada apenas a interface *Explorer*, porém ao aprender a utilizar a interface *Experimenter*, surgiu a possibilidade de criar rotinas com vários algoritmos calibrados de diversas formas diferentes, simplificando a tarefa de identificar as melhores calibragens para cada algoritmo.

O desenvolvimento desse TDD mostrou que é viável aplicar a MD na resolução do problema de perdas nos sistemas de distribuição de energia elétrica, os algoritmos treinados são capazes de detectar possíveis usuários fraudulentos, e pode auxiliar no gerenciamento das vistorias em campo.

6.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

Existe uma gama muito grande de problemas que podem ser resolvidos através do uso da MD. Qualquer atividade que tenha uma grande quantia de dados pode aplicar a MD na descoberta de padrões em diversas áreas, como por exemplo: Bancos, Empresas de Empréstimos, Mercados, entre outros.

Mantendo o foco no problema de fraudes no sistema de distribuição de energia elétrica, podem ser feitos diversos estudos usando mineração de dados, entre eles estão:

a) Aplicar outros algoritmos de MD na tarefa de identificação de usuários fraudulentos, como por exemplo, as Redes Neurais Artificiais (RNA);
b) Ao invés de aplicar o processo de MD com o objetivo de identificar os usuários fraudulentos, também é viável aplicar MD para detectar os transformadores onde há mais perda de energia, consequentemente achando quais locais devem ser vistoriados;
c) Empregar MD no histórico de consumo diário dos consumidores, podendo identificar possíveis fraude ou anomalias rapidamente, diminuindo a quantia de energia perdida pelas concessionárias; ou
d) Criar uma nova metodologia que busque encontrar usuários fraudulentos que tenham um perfil diferente do utilizado neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

Agencia Nacional de Energia Elétricas ANEEL, **Condições Gerais de Fornecimento de Energia Elétrica**, Resolução 456. 1° edição. ANEEL, 2000.

AGRAWAL, R.; SRIKANT, R. **Fast Algorithms for Mining Association Rules.** IBM Almaden Research Center, Santiago, Chile, 1994.

ARAUJO, Antonio Carlos Marques de. **Perdas e Inadimplência na atividade de distribuição de energia elétrica no Brasil.** 2006. Tese (Pós-Graduação em Engenharia) - Programa de pós-graduação de engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

ARAUJO J. B.; PEREIRA S. E. Aplicação de Data Mining na área de CRM como ferramenta gerencial para tomada de decisão em empresas modernas, Rio janeiro, revista UNIABEU, abril de 2011.

BRACHMAN, R.; ANAND, T. The Process f Knowledge Discovery in Databases: A Human Centered Approach. In ADKKM, AAAI/MIT, pag. 37 a 58, 1996.

BOUCKAERT R. R.; FRANK E.; HALL M.; KIRKBY R. REUTEMANN P.; SEEWALD A.; SCUSE D. **WEKA Manual for Version 3-6-2**, University of Waikato, Hamilton, New Zealand, Janeiro de 2010.

BUENO, M. F.; VIANA, M. R. **Mineração de Dados: Aplicações, Eficiência e Usabilidade.** Anais do Congresso de Iniciação Científica do INATEL. INCITEL, 2012.

BRADZIL, P. B. **Construção de modelos de decisão a partir de dados**. 1999. Disponível em: <http://www.nacc.up.pt/~pbrazdil/Ensino/ML/ModDecis.html>. Acessado em: 15 de março de 2015.

CAMILO, C. O.; SILVA, J. C. **Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas.** Relatório Técnico do Instituto de Informática da Universidade Federal de Goiás, agosto de 2009.

CASTANHELAS, Luciana Gomes. **Aplicação de Técnicas de Mineração de Dados em Problemas de Classificação de Padrões**. 2008. Dissertação (Pós-Graduação em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, 2008.

CORRÊA, H. H.; SFERRA, Â. M. C. J. **Conceitos e Aplicações de Data Mining.** Revista de Ciências e Tecnologia, V. 11, Nº 11, p 19-34, dezembro de 2003.

Companhia Paranaense de Energia COPEL, **Relatório de Sustentabilidade de 2014**, Curitiba, 30 de abril de 2015.

DELAIBA, A. C.; REIS FILHO, J.; GONTIJO, E. M.; MAZINA, E.; CABRAL, J.E.; PINTO, J. O. P. **Fraud Identification in Electricity Company Costumers Using Decision Tree**. Netherlands, International Conference on System, Man and Cybernetics, outubro de 2004.

DELGADO, Jair José Lopes. **Sistema de Informação de Apoio à Detecção de Perdas de Energia Elétrica – O Caso da Electra**. 2010. Dissertação (Departamento de Eletrônica, Telecomunicações e Informática - UA) Programa para obtenção de grau mestre em Engenharia Eletrônico e Telecomunicação vertente e Sistema de Informação, Universidade de Aveiro 2010.

Departamento de Monitoramento do Sistema Elétrico DMSE, **Boletim Mensal de Monitoramento do Sistema Elétrico Brasileiro**, janeiro de 2016.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. **Advances in knowledge discovery & data mining.** Menlo Park, CA, USA: AAAI/MIT, 1996 a.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **Knowledge Discovery and Data Mining: Toward a Unifying Framework.** Redmond, WA, USA, AAAI, 1996 b.

FRANCO E. M. C.; LÓPEZ G. P.; QUEIROZ A. S. **Algoritmos de Inteligência Computacional Utilizados na Detecção de Fraudes nas Redes de Distribuição de Energia Elétrica**, XI LATIN AMERICAN CONGRESS ELECTRICITY GENERATION AND TRANSMISSION - CLAGTEE, 2015.

FRANK, E; HALL, M. WITTEN, I. H. **Pratical Data Mining Tutorial 01: Introduction to the WEKA Explorer**. University of Waikato, Nova Zelândia, 2011.

GARCIA, S. C. **O uso de árvores de decisão na descoberta de conhecimento na área da saúde.** In: SEMANA ACADÊMICA. Rio Grande do Sul: Universidade Federal do Rio Grande do Sul, 2000.

GUNN, S. R. **Support Vector Machine for Classification and Regression.** Reporte Técnico, University of Southampton, maio de 1998.

GONÇALVES, Andre Ricardo. **Otimização em Ambientes Dinâmicos com Variáveis Continuas empregando algoritmos de estimação de distribuição**. 2011. Dissertação (Mestrado em Engenharia Elétrica) - Departamento de Engenharia de Computação e Automação Industrial, Universidade Estadual de Campinas, Campinas, 2011.

HALL, Mark. **WEKA.** Department of Computer Science, University of Waikato, Nova Zelândia, 2015.

HALL, M.; REUTMAN, P. **Weka KnowledgeFlow Tutorial for Vesion 3-5-8**, University of Waikato, Nova Zelândia, 2008.

INDURKHYA, N.; WEISS, S.M. **Estimating Performance Gains for Voted Decision Trees**. 1999. IBM Research Division Technical Report in Intelligent Data Analysis.

INGARGIOLA, G. **Building classification models: ID3 and C4.5**. 1996. Disponível em: <http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html>. Acesso em: 15 de março de 2015.

LEMOS, E. P; STEINER, M. T. A; NIEVOLA, J. C. **Análise de credito bancário por meio de árvores de decisão.** Revista Administração USP, São Paulo, p. 255 a 234, setembro de 2005.

NAVEGA, S. **Princípos Essenciais do Data Mining**. Cenadem, Anais da InfoImagem, novembro de 2002.

NIMER, F.; SPANDRI, L. C. **Data Mining.** Revista Developers, Fevereiro de 1998, página 32.

OLIVEIRA JUNIOR, Gilson Medeiros de. **Máquina de Vetores Suporte: estudo e análise de parâmetros para otimização de resultado**. 2010. Trabalho de Graduação (Centro de Informática - UFPE) Graduação em Ciências da Computação, Universidade Federal do Pernambuco, 2010.

PIATETSKY-SHAPIRO, G. **KnowledgeDiscovery in real Databases**, AI Magazine, Janeiro de 1991.

QUINLAN, J.C. **C4.5: Programs for machine learning.** San Mateo, Morgan Kaufmann, 1993.

Resende T. **Perdas na distribuição: baixa tensão, altos prejuízos.** Publicado na Associação Brasileira de Distribuidores de Energia Elétrica (ABRADEE), Brasília, outubro de 2013.

REUTMAN, P.; SCUSE, D. **Weka Experimenter Tutorial for Vesion 3-5-5**, University of Waikato, Nova Zelândia, 2007.

RUSSELL, Stuart J; NORVING, Peter. **Inteligência Artificial**. Editora Hall, 1994.

SEPÚLVEDA, G.; P.; L, **Abordagem de Problemas da Área de Engenharia Elétrica Através de Inteligência Computacional**, publicado na Universidade Estadual Paulista - UNESP, campus Ilha Solteira, São Paulo, 2016.

THEARLING, Kurt.; BERSON, Alex.; SMITH, Stephen. **Building Data Mining Application for CRM**. Editora Mc Graw Hill, 22 de dezembro de 1999.

KOHAVI, R.; SOMMERFIELD, D.; ODUGHERTY, J. **Data Mining Using MLC++ A Machine Learning Library in C++**. Abril de 1997. Disponível em: <http://ai.stanford.edu/~ronnyk/mlcj.pdf >. Acesso em: 25 de agosto de 2016.