MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

LUCAS RAMOS CARDOSO TINÔCO CORTEZ
RAFAEL FARIAS CAÇÃO

WORKFLOW PARA DESCOBERTA DE CONHECIMENTO EM SEGURANÇA CIBERNÉTICA

Rio de Janeiro
2017

INSTITUTO MILITAR DE ENGENHARIA

LUCAS RAMOS CARDOSO TINÔCO CORTEZ
RAFAEL FARIAS CAÇÃO

WORKFLOW PARA DESCOBERTA DE CONHECIMENTO EM SEGURANÇA CIBERNÉTICA

Projeto de Fim de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Engenheiro de Computação.

Orientadora: Prof[a]. Maria Claudia Reis Cavalcanti - D.Sc.
Co-Orientador: Frederico Tosta Oliveira - M.Sc.

Rio de Janeiro
2017

INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 – Praia Vermelha

Rio de Janeiro – RJ CEP: 22290-270

006	Cortez, Lucas Ramos Cardoso Tinôco
C818w	Workflow para descoberta de conhecimento em segurança cibernética / Lucas Ramos Cardoso Tinôco Cortez; Rafael Farias Cação; orientados por Maria Claudia Reis Cavalcanti; Frederico Tosta Oliveira – Rio de Janeiro: Instituto Militar de Engenharia, 2017.
	?p. : il.
	Projeto de Fim de Curso (PROFIC) – Instituto Militar de Engenharia, Rio de Janeiro, 2017.
	1. Curso de Engenharia de Computação – Projeto de Fim de Curso. 2. Botnet. I. Cação, Rafael Farias. II. Cavalcanti, Maria Claudia Reis. III. Oliveira, Frederico Tosta. IV. Título. V. Instituto Militar de Engenharia.

INSTITUTO MILITAR DE ENGENHARIA

LUCAS RAMOS CARDOSO TINÔCO CORTEZ
RAFAEL FARIAS CAÇÃO

WORKFLOW PARA DESCOBERTA DE CONHECIMENTO EM SEGURANÇA CIBERNÉTICA

Projeto de Fim de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Engenheiro de Computação.

Orientadora: Prof[a]. Maria Claudia Reis Cavalcanti - D.Sc.

Co-Orientador: Frederico Tosta Oliveira - M.Sc.

Aprovado em 5 de OUTUBRO de 2017 pela seguinte Banca Examinadora:

Prof[a]. Maria Claudia Reis Cavalcanti - D.Sc. do IME - Presidente

Frederico Tosta Oliveira - M.Sc. do IME

Prof. Ronaldo Goldschmidt - D.Sc. do IME

Rio de Janeiro
2017

Ao Instituto Militar de Engenharia, alicerce da minha formação e aperfeiçoamento.

AGRADECIMENTOS

Agradeço a todas as pessoas que me incentivaram, apoiaram e possibilitaram esta oportunidade de ampliar meus horizontes.

Meus familiares e mestres.

Em especial à Professora Orientadora Dra. Maria Claudia Reis Cavalcanti, por suas disponibilidades e atenções.

"Nenhuma localidade, nenhuma indústria ou organização é a prova de balas quando se trata do comprometimento dos dados "

VERIZON'S 2016 DATA BREACH INVESTIGATIONS REPORT

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

WfMS	Workflow Management System

RESUMO

A evolução dos sistemas computacionais e da internet fizeram surgir a necessidade de um tipo diferente de defesa, a defesa cibernética. Ela consiste na proteção de informações contra ameaças cibernéticas como por exemplo o terrorismo cibernético e espionagem. Aliado a isso, o advento de técnicas de aprendizado de máquina possibilitaram o desenvolvimento de sistemas de detecção ainda mais robustos. Neste contexto, aumenta a relevância de frameworks capazes de atender as diversas necessidades nos processo de negócio que envolvem a segurança de dados. Dito isso, este trabalho possui como finalidade implementar um workflow para a detecção de botnets para ser utilizado em um sistema de gerenciamento de workflows (WfMS) e assim gerar conhecimento para ser utilizado em decisões relativas a segurança de redes, através da experimentação em conjunto de treino com recolhimento de estatísticas deste processo.

ABSTRACT

The evolution of computer systems and the internet have raised the need for a different kind of defense, the cyber defense. It consists of protecting information against cyber threats such as cyber terrorism and espionage. In additiopn to this, the advent of machine learning techniques enabled the development of even more robust detection systems. In this context, the relevance of frameworks capable of meeting the diverse needs in the business processes that involve data security arose. This work aims to implement a workflow for the detection of botnets to be used in a workflow management system (WfMS) and thus generate knowledge to be used in decisions regarding network security, through experimentation in a training set while collecting statistics of this process.

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Com o crescimento do uso das redes de computadores, um elevado número de serviços governamentais e privados passaram a ser oferecidos através da Internet. Assim, grande parte da população e das empresas passou a utilizar essa rede para realizar atividades e serviços, tais como, matrículas em escola, declaração de impostos, emissão de notas fiscais, entre outros. Dentro desse contexto, o grau de importância da Internet cresceu vertiginosamente, assim como a necessidade do funcionamento pleno de todos os seus serviços. Como consequência, a Internet tornou-se alvo em potencial para atividades ilícitas, tais como, roubo de informações, paralisação de serviços, envio de spam, ciber terrorismo e hacktivismo. O exemplo mais recente desse tipo de atividade foi o conjunto dos ciberataques de escala global realizados no dia 12 de Maio de 2017 pelo programa malicioso chamado WannaCry. Esse programa sequestrou, através de um sistema de criptografia, dados de diversas organizações do mundo, cobrando um resgate pela chave.

Segundo pesquisa da Cybersecurity Ventures os custos relativos a crimes cibernéticos irão crescer de 3 para 6 trilhões de dólares anualmente até 2021, e os gastos com defesa irão acumular o valor de 1 trilhão de dolares nos próximos 5 anos (VENTURES, 2017). O número de incidentes cresceu nos últimos anos nos Estados unidos, como mostram os dados presentes no gráfico da Figura 1.1, e espera-se um crescimento ainda maior (GAO, 2017).

Além disso, segundo o relatório The global Risks Report 2016 do Fórum Econômico Mundial, uma parcela significativa de crimes cibernéticos não é detectada. Os dados são ainda mais agravantes no contexto das organizações de porte pequeno e médio. Uma pesquisa realizada pela Keeper Security, apontou que metade das empresas consultadas sofreram violação de dados num período de 12 meses anterior à consulta (SECURITY, 2016). Diante deste cenário, é evidente a importância de pesquisas e investimento na área de defesa cibernética.

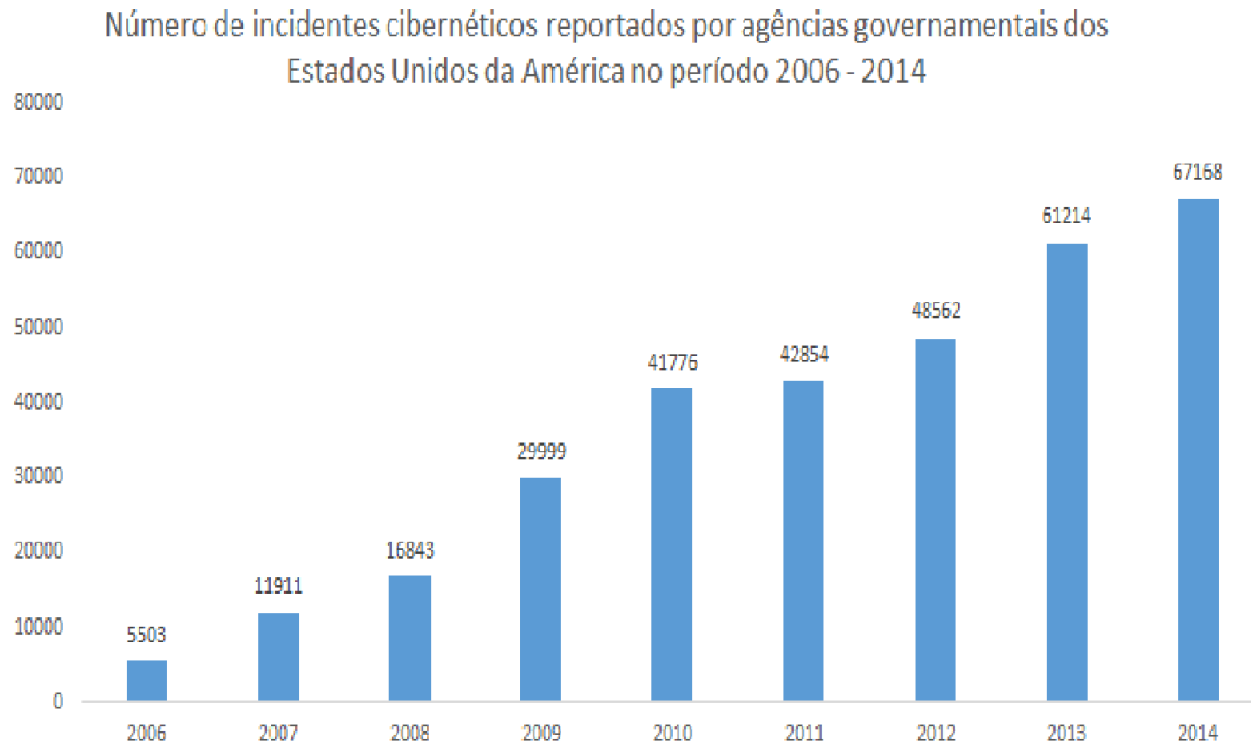

FIG. 1.1: Crescimento de incidentes cibernéticos nos Estados Unidos de 2004 a 2014 - (Gao, 2017)

1.2 OBJETIVOS

O objetivo deste trabalho é implementar um workflow de detecção de botnets utilizando árvores de decisão, uma técnica de aprendizado de máquina. Mais especificamente, os objetivos são:

a) Analisar bibliografia específica relativa à defesa cibernética, aprendizado de máquina e mineração de dados no que compreende os assuntos de detecção de botnets, problemas de classificação e algorítmos de aprendizado de máquina para classificação;

b) Implementar estrutura do workflow com algoritmo de aprendizado de máquina da ferramenta Spark com gerenciamento pela ferramenta Vistrails;

c) Criar ambiente simulado para a experimentação dos algoritmo e recolhimento de resultados;

d) Recolher dados de desempenho;

e) Calcular estatísticas de desempenho a fim de comparação.

1.3 JUSTIFICATIVA

O trabalho justifica-se com base na Estratégia Nacional de Defesa que em 2011 implementou o Plano Estratégico de Defesa Cibernética sob responsabilidade do Exército. Cita-se novamente as vulnerabilidades expostas pelo ataque do malware WannaCry ocorridos em 12 de Maio de 2017 bem como as estatísticas apresentadas na motivação deste trabalho.

1.4 METODOLOGIA

A metodologia adotada constituiu-se inicialmente na implementação de um workflow específico a ser apresentado no texto deste trabalho. Após implementado, um subset de parâmetros foi escolhido e variado segundo um processo de direcionamento, onde valores esparsos foram selecionados através de um iterador para produzir uma visão geral do cenário de resultados, com posterior seleção de um região para refinamento. A metodologia adotada para a realização do projeto foi a seguinte:

a) Contextualização do projeto e recolhimento de informações e bibliografia

b) Análise de material teórico;

c) Escolha de algorítmos de apredizado de máquina do Spark;

d) Desenvolvimento do projeto da interface entre as ferramentas e implementação no workflow;

e) Experimentação dos algorítmos em base de dados pronta;

f) Cálculo de estatísticas de desempenho e comparação dos resultados.

A metodologia adotada para a implementação do workflow seguiu as seguintes etapas:

a) Captura de informações da rede que está sendo analizada;

b) Tratamento dos dados da rede afim de produzir formatação específica;

c) Classificação dos dados da rede com ferramentas de aprendizado de máquina;

d) Apresentação dos dados tratados em formatação específica para cálculo das estatísticas e posterior analise de dados.

1.5 ESTRUTURA DO TEXTO

O texto apresenta-se em cinco partes principais:

a) Fundamentação teórica que consiste na apresentação dos conceitos básicos de segurança de redes, mineração de dados e aprendizado de máquina;

b) Sistema de gerenciamento de Workflows que consiste na apresentação do sistema ao qual o workflow está relacionado;

c) Apresentação e implementação dos algoritmos de aprendizado de máquina adotados e estrutura no workflow proposto;

d) Experimentos e Resultados que consiste na apresentação dos dados referentes aos experimentos realizados;

e) Estatísticas que consiste na apresentação de estatísticas de desempenho do workflow e de cada um dos algorítmos utilizados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 SEGURANÇA DA INFORMAÇÃO

Dois conceitos relacionados, de forma bastante próxima, são os de defesa e segurança cibernética. Neste contexto, insere-se as diversas tentativas de detecção de botnets e as diversas abordagens adotadas.

2.1.1 DEFESA E SEGURANÇA CIBERNÉTICA

Inicialmente tem-se que a "Defesa cibernética diz respeito ao conjunto de ações defensivas, exploratórias e ofensivas, no contexto de um planejamento militar, realizadas no espaço cibernético, com as finalidades de proteger os sistemas de informação, obter dados para a produção de conhecimento de inteligência e causar prejuízos aos sistemas de informação do oponente"(CRUZ JÚNIOR, 2013).

Já a "Segurança cibernética refere-se à proteção e garantia de utilização de ativos de informação estratégicos, principalmente os ligados às infraestruturas críticas da informação (redes de comunicações e de computadores e seus sistemas informatizados) que controlam as infraestruturas críticas nacionais. Também abrange a interação com órgãos públicos e privados envolvidos no funcionamento das infraestruturas críticas nacionais, especialmente os órgãos da administração pública federal"(CRUZ JÚNIOR, 2013).

2.1.2 BOTNETS

Inicialmente, conceitua-se bot como aplicações de software maliciosas que infectam máquinas via internet. Um grupo de ditas máquinas se chama Botnet, e agem sob comando de algum humano, a quem chamamos de Botmaster. Máquinas infectadas se escondem à discrição do Botmaster, que pode então ordená-los a um ataque ou tarefa. Botnets têm crescido em importância como um problema de segurança da internet, tendo sido usados em ataques de DDoS, roubo de informações bancárias, spam e outros. Foram, então, extensamente estudados. Primeiramente, diferem de outros tipos de malware por possuírem uma organização de comando e controle(C&C). Podemos classificá-los segundo diversos critérios (AMINI et al., 2015):

a) Arquitetura:

Centralizada, P2P ou desestruturada.

b) Topologia:

Estrela, múltiplos servidores, hierárquica ou aleatória.

c) Protocolo de comunicação:

IRC, HTTP, P2P e DNS.

d) Mecanismo de infecção:

Download da internet, anexo de email, automático.

e) Proposta:

Roubo de informações, computação distribuída, interrupção de serviço, fraude, add-on e crescimento da rede.

f) Ataques:

Phishing, DDoS, fraude de click, spam, roubo de identidade, vazamento de informações, scareware, rastreamento de tráfego e keylogging.

2.1.3 CICLO DE VIDA DE BOTNETS

O entendimento de o que é uma botnet passa pelo entendimento de seu ciclo de vida e como ele atua em ataques. A operação de um botnet baseia-se no constante recrutamento de novos bots, que são identificados em hosts vulneráveis. Isso é necessário por que frequentemente bots tornam-se comprometidos, uma vez que sistemas de detecção tendem a adicionar tais bots que são detectados em listas negras afim de restringir a atividade deste no alvo. Uma vez comprometido, o bot torna-se inutil para o intuito do Botmaster e deve ser substituído (BARBOSA, 2014). O ciclo de vida da botnet é descrito pela figura 2.1.

A figura mostra que a infecção ocorre em dois momentos: uma infecção primária que atinge o host identificado como vulnerável e posteriormente uma infecção secundária que, através de um malware previamente adquirido, realiza o download de um código malicioso, que será o responsável pela estruturação da botnet. De um modo geral, o sucesso da botnet depende da disponibilidade de seus hosts e da taxa de reposição de bots comprometidos (BARBOSA, 2014). Sucintamente, o ciclo de vida pode ser ilustrado pelos passos de 1 a 4 da figura 2.1. No primeiro momento, um elemento frágil é identificado por algum dos membros da rede, sendo então infectado. Essa infecção inicial possui como objetivo

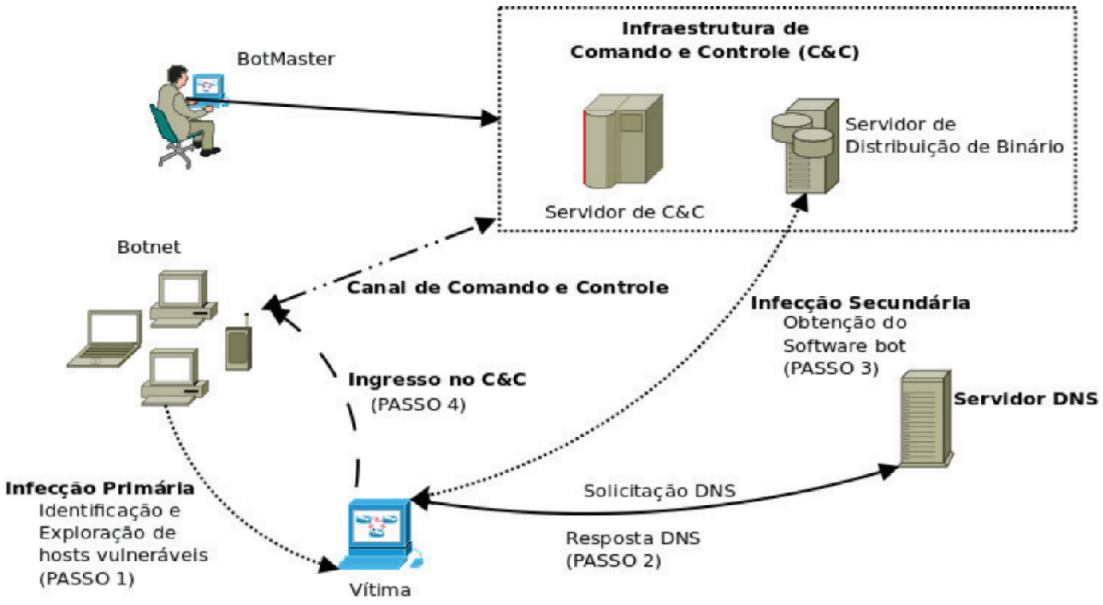

FIG. 2.1: Ciclo de vida da botnet - (BARBOSA, 2014)

a realização de consultas DNS para identificar o distribuidor de software malicioso da rede, o que consiste no passo 2 ilustrado. No próximo passo ocorre a infecção efetiva, com o download do software bot do distribuidor previamente identificado, culminando no ingresso na botnet deste novo host (passo 4).

2.2 ABORDAGENS PARA DETECÇÃO DE BOTNETS

A detecção de botnets segue três abordagens baseadas no ciclo de vida da botnet e a primeira consiste na utilização de honeypots. O nome desta abordagem está relacionado a atratividade que este tipo de aplicação consegue simular, passando-se por um sistema vulnerável para ser alvo de possíveis atacantes e assim permitem certas interações com a botnet. Podem atuar de forma ativa, fornecendo acesso a rede em um ambiente que o simulador controla, ou de forma passiva, apenas para detectar tentativas de varreduras e de acesso. Este tipo de ferramenta de detecção somente possibilita identificar características dos atacantes e da interação com a botnet, de forma que fornece informações sobre aquele ataque em específico. Isso limita a sua usabilidade (BARBOSA, 2014). A próxima abordagem consiste na analise de trafego da rede baseada nas assinaturas dos dados que nela circulam. Ela se baseia na comparação constante de dados no fluxo da rede com uma base de assinaturas descritas por um conjunto de regras. A limitação deste tipo de abordagem consiste na necessidade de um mapeamento prévio de agentes maliciosos e descrição destes através de regras de bloqueio. Por outro lado, apresenta uma taxa de su-

cesso alta no que diz respeito a identificação do que já está mapeado e no não bloqueio de falsos positivos. A última abordagem consiste na análise de fluxo de dados para detecção de anomalias. Neste tipo de abordagem, procura-se identificar padrões no fluxo da rede que possam descrever características de controle da botnet. Neste caso, busca-se comportamentos incomuns na rede, como portas de acesso restrito com grande volume de dados sendo transportados, alta latência, e comportamentos incomuns do sistema (BARBOSA, 2014). São utilizadas ferramentas de classificação e apredizado de máquina para descrever tais padrões, bem como algorítmos de clustering e técnicas de teoria da informação. Pode atuar como forma de prevenir ataques não detectados pela segunda abordagem, e assim aumentar a segurança do sistema. Este trabalho, baseou-se nesta última abordagem, na estruturação e desenvovimento do workfow.

2.3 WORKFLOW

Workflow, que em tradução livre significa fluxo de trabalho, consiste na sistematização do processo de negócio de forma parcial ou total. Este tipo de modelagem, consiste num fluxo de informações que resume a execução de uma determinada tarefa, software ou processo, mostrando a comunicação entre as partes envolvidas, seja por meio de troca de documentos, arquivos ou parâmetros. Essa sistematização do processo de negócio identifica as várias atividades relacionadas e o controle envolvido por meio de regras de gerenciamento. De forma geral, a modelagem através de workflows permite uma harmonização maior do processo que está sendo modelado, com maior controle sobre todas as etapas envolvidas e melhor visualização dos dados em processo. Desta forma, no escopo deste trabalho, a modelagem através de workflows permite maior controle do processo de detecção e da comunicação entre os módulos envolvidos (PAIM, 2009). De acordo com a Workflow Management Coalition (WfMC) os conceitos relativos a um workflow podem ser descritos pela figura 2.2. Resumidamente, a figura 2.2 descreve que em um processo de negócio no qual são definidas as necessidades do negócio, descritas através de uma lista de atividades e processo inerentes a ele, um WfMS é capaz de automatizar algumas dessas atividades para instâncias definidas do negócio, e assim realizar o controle do processo por meio de workflows definidos, que vão desde a invocação de aplicações até a delegação de trabalhos a componentes do workflow.

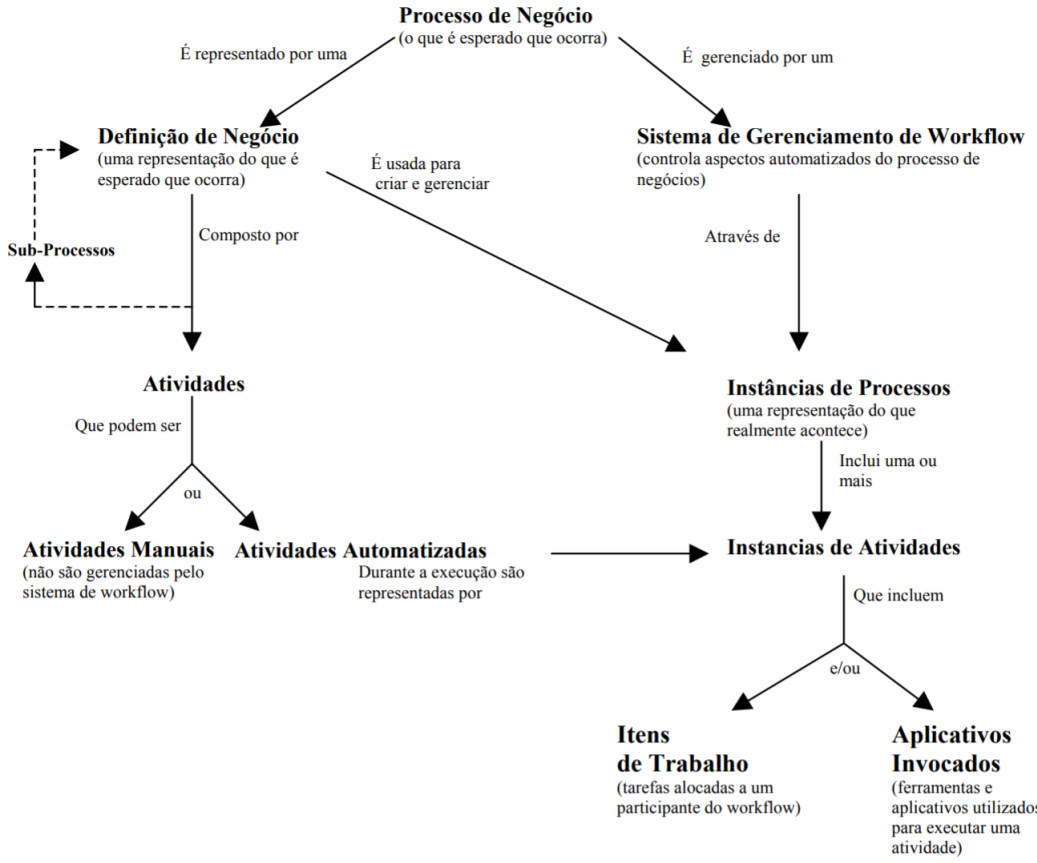

FIG. 2.2: Conceitos relativos a um workflow - (COALITION, 1996)

2.4 SISTEMA DE GERENCIAMENTO DE WORKFLOWS

O termo Gerenciamento de Workflows refere-se a ideias, métodos, técnicas e softwares usados para apoiar processos de negócios estruturados. O fluxo e a distribuição de trabalho devem ser controlados pelos WfMS (PÁDUA). A responsabilidade do WfMS é a de verificar as atividades do workflow e o correto funcionamento da sequência em que são executadas, bem como possibilitar a manipulação das instâncias do workflow. De um modo geral, o Sistema de Gerenciamento de Workflows deve possibilitar a modularização do negócio e a execução desses módulos de forma supervisionada. Inicialmente, isso consiste ~~pela proveniência~~ no uso de ferramentas de modelagem para a construção do workflow em si. Feito isto, traduz-se numa ferramenta de controle de execução de cada módulo do processo do negócio e de fato em uma ferramenta de gerenciamento, como descrito pela figura 2.3 (SALIMFARD; WRIGHT, 2001). Na figura, os componentes do WfMS são descritos em dois momentos distintos: um tempo de construção e um tempo de execução. No tempo de construção, ocorre a definição dos processos envolvidos no projeto em questão, através de uma modelagem e de ferramentas analíticas. Após isso, em um

contexto de execução ocorre a interação mútua entre os serviços de recursos humanos, as aplicações no contexto do WfMS e do projeto em destaque e da representação do negócio no contexto do gerenciamento do workflow, de modo que estejam alinhados e atualizados com as necessidades do negócio.

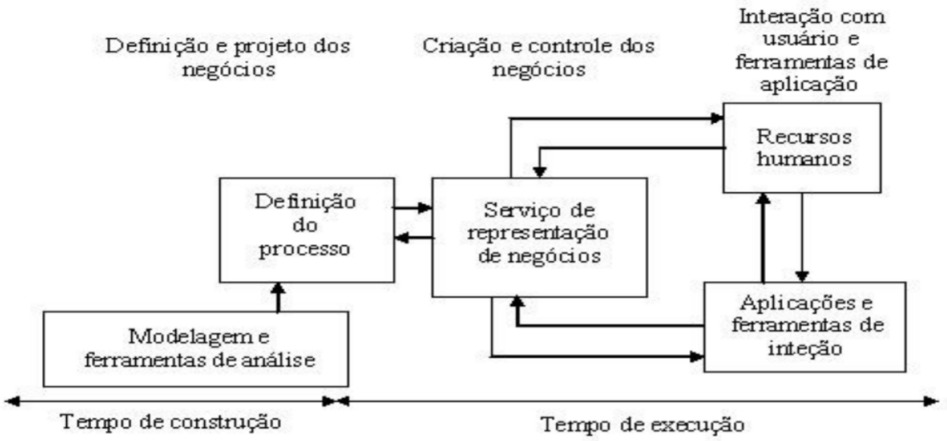

FIG. 2.3: Componentes de um WfMS - (SALIMFARD; WRIGHT, 2001)

2.4.1 VISTRAILS E SPARK

VisTrails é uma ferramenta que foi desenvolvida na Universidade de Utah que prove suporte para exploração e visualização de dados através de um sistema de gerenciamento de workflows. Sua interface permite a construção de workflows de forma gráfica e possibilita a captura de proveniência. Além disso, ele permite uma interação com o Spark, que é um sistema de computação em cluster de propósito geral, utilizado no âmbito deste trabalho como unidade de processamento e classificação dos fluxos da rede, em cada um dos diversos módulos que são chamados pelo gerenciamento do Vistrails.

2.5 ARQUITETURA DE SISTEMAS DE GERENCIAMENTO DE WORKFLOWS PARA DETECÇÃO DE BOTNETS

2.5.1 O PROBLEMA

Apesar de haver bastante pesquisa na área de detecção de botnets, pouco se tem avançado na comparação dos algoritmos já existentes e na reprodução fiel de algoritmos estudados. Uma das dificuldades é a obtenção e compartilhamento de dados de tráfego de rede, que é justo, dado a natureza privada dessa informação. Porém, ainda assim,

impacta negativamente na hora de repetir ou comparar experimentos de terceiros (AVIV; HAEBERLEN, 2011). Há ainda a dificuldade de experimentos mal descritos e a falta de métodos de comparação e métrica de erros (GARCIA et al., 2014). No artigo (GARCÍA et al., 2014), os autores comparam quatorze métodos de detecção de botnets, notando que dez destes não puderam ser reproduzidos por falta de informação e doze por não puderam pela falta de seus datasets.

2.5.2 A SOLUÇÃO

Para resolver parcial ou totalmente esse problemas e ainda otimizar o processo de obtenção de dados, pode se trabalhar numa arquitetura unificada para sistemas de gerenciamento de workflows, permitindo a automatização de diversas fases do processo e a comparação mais fiel de performance de diferentes algoritmos compartilhados por meio da arquitetura (AVIV; HAEBERLEN, 2011).

2.6 MINERAÇÃO DE DADOS

Mineração de dados pode ser vista como a evolução natural da tecnologia da informação, ao passo que o ser humano gera muitos dados em todas as suas atividades e existe uma demanda cada vez maior de transformar estes dados em informação útil (HAN; KAMBER, 2006).

A figura 2.4 mostra como o processo de mineração de dados se insere dentro do contexto de descoberta do conhecimento. Os dados são preprocessados e preparados para serem utilizados por algorítmos que podem ou não interagir com o usuário. A partir disso, estes dados são submetidos a algorítmos afim de se extrair padrões que são apresentados de acordo com a relevância da informação que carregam e que podem gerar conhecimento a ser analisado(HAN; KAMBER, 2006). Neste trabalho, o processo de mineração de dados se traduz na solução de problemas de classificação, no tratamento de dados que trafegam na rede como uma tentativa de identificar padrões que os caracterizem como botnets.

2.7 ÁRVORES DE DECISÃO

O algorítmo de classificação da árvore de decisão utiliza em sua implementação um paradigma chamado de bottom-up, isto é, obtém se o relacionamento entre variáveis dependentes e independentes em bases de dados previamente rotuladas (ZUBEN). Neste

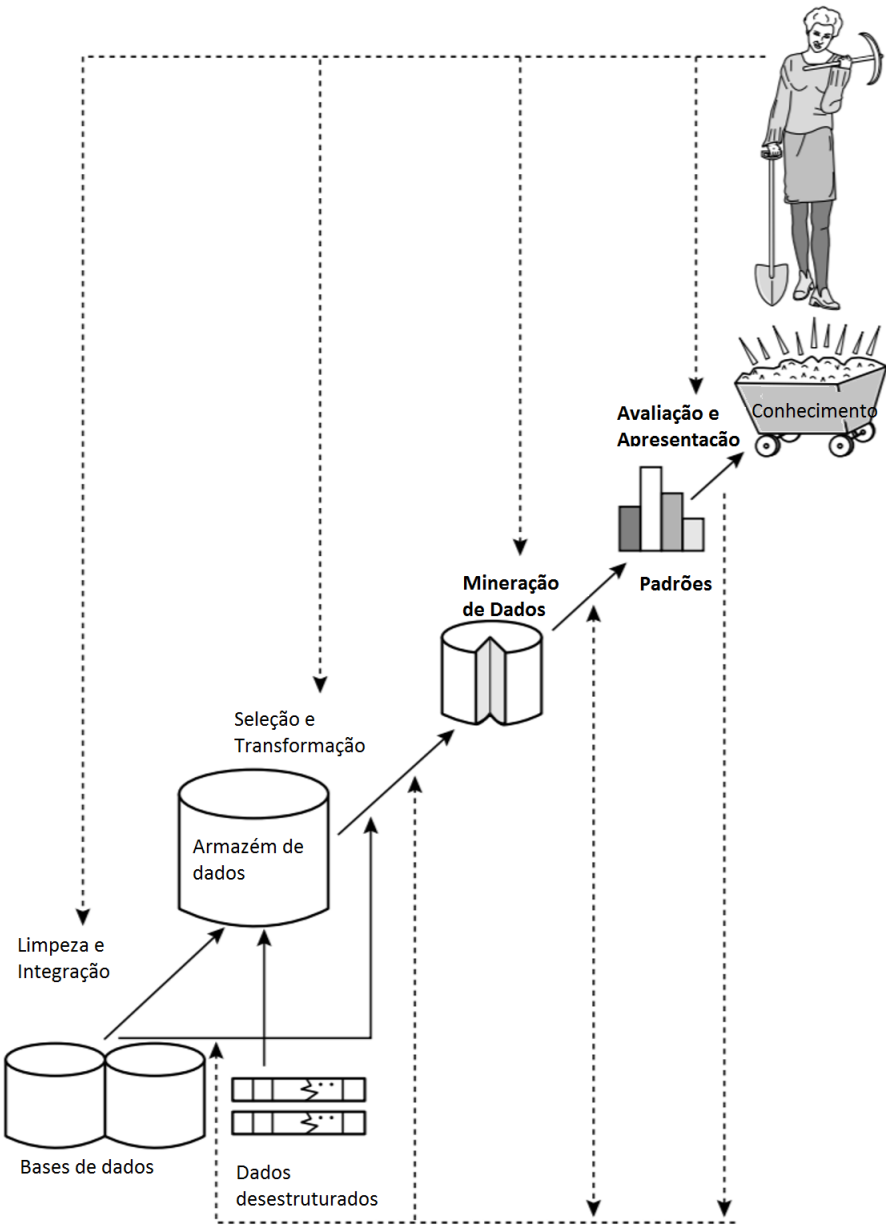

FIG. 2.4: Mineração de dados como um passo no processo de descoberta do conhecimento - (HAN; KAMBER, 2006)

sentido, todas as informações dos objetos, que neste caso são os dados provenientes da rede devem ser expressos em um coleção finita de atributos. Pode-se distinguir dois tipos de treinamento para árvores de decisão: um supervisionado, que requer um número de classes definido, ou não supervisionado que não requer. As classes de classificação podem ser discretas ou contínuas de acordo com a natureza da pertinência dos objetos a elas. Se a pertinência ou não for definida de forma absoluta, fala-se em classes discretas. Se elas forem definidas de forma gradual, fala-se em classes continuas. Neste trabalho, as classes utilizadas para a classificação são definidas como discretas (deseja-se classificar

um dado malicioso ou não) e com quantidade definida de classes, tratando-se de um processo supervisionado. Além disso, distinguem-se duas modelagens de classificação: uma preditiva que busca classificar os atributos cujas classes são desconhecidas de acordo com um modelo previamente aprendido e a uma descritiva que visa distinguir exemplos de classes diferentes. A implementação adotada, por tratar-se de uma ferramenta de detecção, realiza uma modelagem preditiva. O algoritmo usado neste trabalho é chamado de Top-down Induction Decision Tree (TDIDT).

2.7.1 TDIDT

Este algoritmo, que caracteriza-se como um algoritmo de busca gulosa, produz regras de decisão de forma implícita na árvore, a qual é construída por particionamento recursivo (BRAMER, 2007). O algoritmo é descrito segundo o conjunto de treinamento com respeito as classes C_1, C_2,...,C_k e sob as possibilidades em relação a este conjunto:

- Se o conjunto contém somente objetos de mesma classe, a árvore de decisao será uma folha que irá identificar essa classe;

- Se o conjunto é vazio, a árvore será uma folha sem determinação interna (necessita de outra informação para ser classificada);

- Se o conjunto contém exemplos de mais de uma classe, dividi-se em T subconjuntos, numa tentativa de se obter subconjuntos com objetos de mesma classe. Nesta etapa é definido um atributo destes objetos como atributo de predição de modo que para o atributo P existem P_1, P_2,...,P_n resultados possíveis. Os objetos são divididos de acordo com este atributo em subconjuntos T_1,T_2,...,T_n de modo que se um objeto pertence a T_j, entao para o atributo P ele possui valor P_j. Assim, o nó de decisao para o atributo P é criado e uma aresta para cada resultado guia a sequencia da árvore e seus proximos nós.

A árvore de decisão se forma pela iteração recursiva da classificação acima, até que somente tenha-se folhas, a partir de um único conjunto inicial que consiste o nó raiz. Este algoritmo pode ser visualizado pela figura 2.5.
O problema agora consiste em definir o atributo de predição. Para tal, pode-se utilizar métricas que avaliem o desempenho de determinado atributo na divisão daquel conjunto. Os critérios utilizados na implementação adotada foram os da entropia através da medida do ganho de informação entre um nó pai e o nó filho e o do índice gini. A entropia é

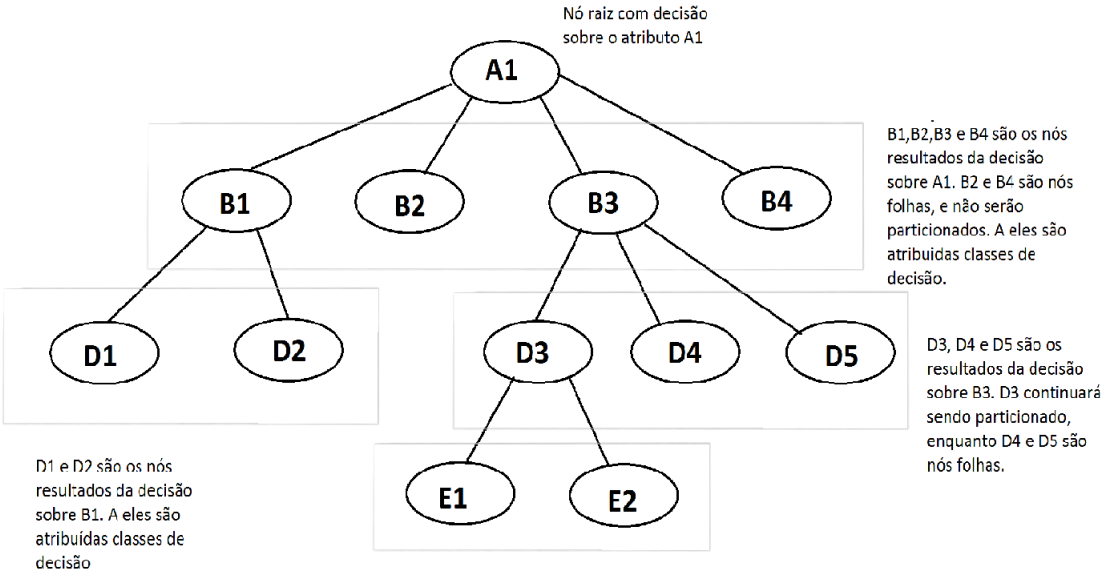

FIG. 2.5: Exemplo de árvore de decisão

calculada segundo a equação:

$$entropia(n) = \sum_{i=1}^{c} p(i/n).\log_2 p(i/n)$$

Onde $p(i/n)$ é a fração dos registros pertencentes a classe i no nó e c é o número de classes. O ganho de informação de um nó é calculado segunda a equação:

$$ganho = entropia(pai) - \sum_{j=1}^{n}[(N(v_j)/N].entropia(v_j)$$

Onde n é o número de valores possíveis do atributo, N é o número total de objetos no nó pai, e $N(v_j)$ o número de objetos associados ao nó filho v_j. Como esta formulação pode induzir a erros relativos a natureza dos atributos (atributos com muitos valores possíveis, ou com valores únicos para cada objeto produzem ganhos maiores de informação uma vez que possuem as menores entropias), utiliza-se a razão de ganho, como ferramenta de avaliação ponderada do ganho sobre a entropia do próprio nó, obtendo-se um melhor resultado para a árvore (QUINLAN, 1988). Assim, é selecionado como atributo aquele que maximiza a razão de ganho na partição do conjunto de testes do nó. A outra métrica utilizada no âmbito deste trabalho foi o índice Gini. Este índice consiste em mais uma medida da impureza de um nó, sendo que valores próximos a zero indicam que o nó é puro. Em contra partida, valores próximos a 1 indicam que os dados estão completamente

divididos em multiclasses naquele nó. O índice Gini é calculado segundo a equação:

$$gini = 1 - \sum_{j=1}^{n} p^2$$

Onde n é o número de classes no nó e p a frequência relativa de cada classe no nó.

3 WORKFLOW PARA DETECÇÃO DE BOTNETS

O workflow consiste no emprego de 5 módulos da ferramenta Vistrails: PySparkShell, LoadNetworkFlows, SelectFeaturesAndLabel, DecisionTree e MulticlassEvaluate. Estes podem ser vistos na figura 3.1.

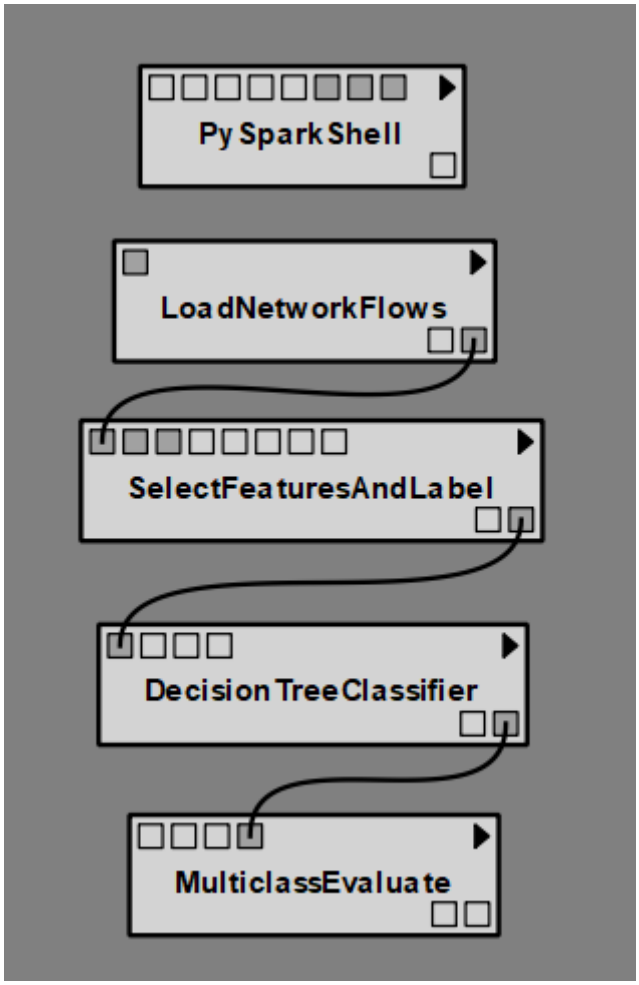

FIG. 3.1: Módulos do Vistrails

Estes módulos interagem então com o Spark para acessar e manipular os dados, segundo ilustrado pelos diagramas das figuras 3.2 e 3.3.

3.1 PYSPARKSHELL

Esse módulo cria um canal SSH para o cluster Spark e instancia um shell do PySpark. A comunicação entre os módulos do Vistrails e o Spark é toda realizada através desse

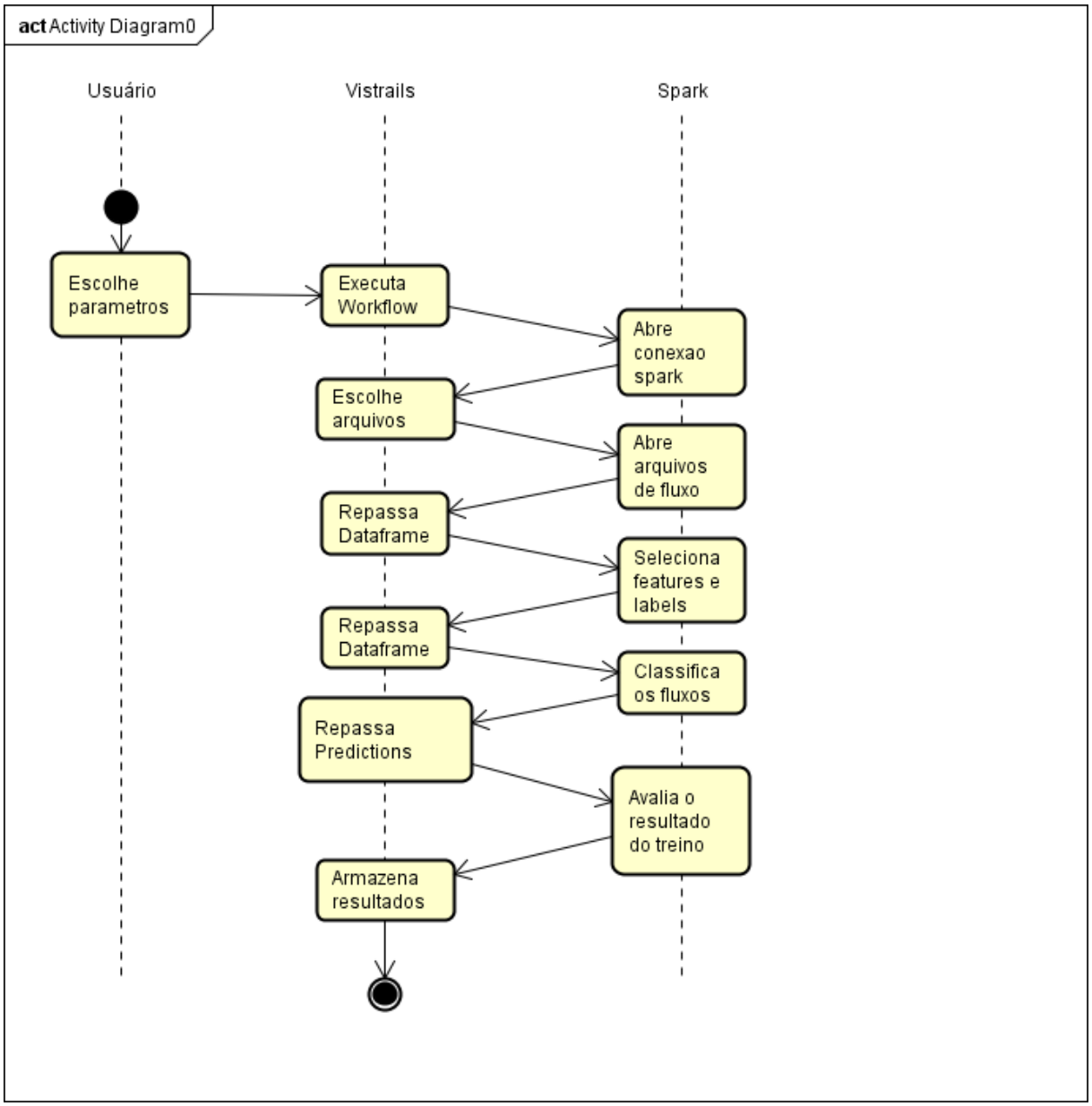

FIG. 3.2: Diagrama de atividade

canal. A conexão com o shell continua funcionando mesmo após o término da execução do workflow, o que causa o Spark a manter dados e recursos de hardware disponíveis para a aplicação, possibilitando o reuso de forma ótima e permitindo a execução parcial do workflow, como nos casos de tratamento de erros ou de troca de parâmetros. Para fechar essa conexão, é necessário fechar o Vistails ou recarregar o pacote Spark.

3.2 LOADNETWORKFLOWS

Esse módulo manda o Spark carregar os arquivos de fluxos de rede, que se encontram em formato CSV. Sua porta de output manda um objeto dataframe para o próximo

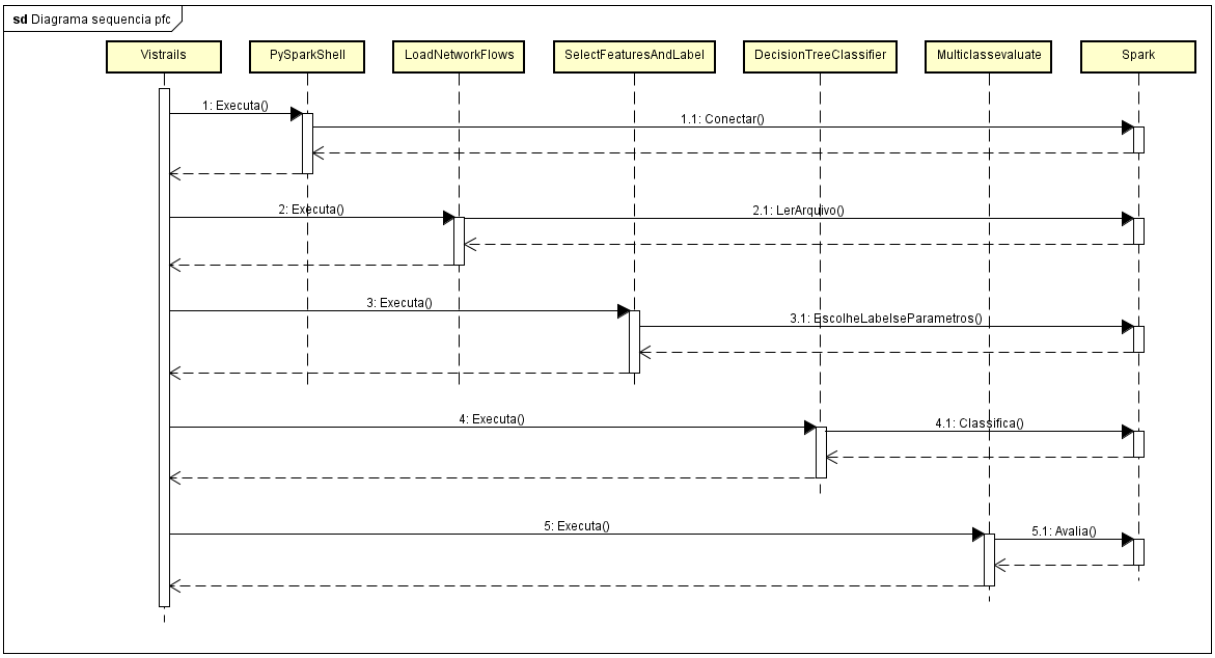

FIG. 3.3: Diagrama de sequência

módulo, para que ele possa acessar o conteúdo dos arquivos abertos pelo Spark.

3.3 SELECTFEATURESANDLABEL

Esse módulo permite ao usuário a escolha dos parâmetros a serem analizados pelo classificador. Permite também a manipulação dos labels a serem classificados. Recebe como entrada um dataframe que permite o acesso aos arquivos abertos no Spark e então trata esses dados para conter apenas os parâmetros e labels escolhidos. Envia na porta de output um objeto dataframe para o próximo módulo poder acessar esses dados no Spark.

3.4 DECISIONTREECLASSIFIER

Esse módulo serve de wrapper para a implementação do algoritmo de árvore de decisão contida no spark. Possui diversos parâmetros para configuração do algoritmo a ser rodado no Spark, como tamanho máximo de profundidade na árvore, tamanho máximo de memória a ser utilizada, mínimo de ganho necessário para expandir um nó na árvore de decisão. Utiliza o dataframe recebido na porta de input para acessar e classificar os dados encontrados no Spark, utilizando-se dos parâmetros escolhidos pelo usuário para criar a árvore de decisão e então predizer as labels dos dados de validação. Envia pela porta de output um dataframe para acesso às predições do algoritmo.

3.5 MULTICLASSEVALUATE

Esse módulo manda o Spark avaliar as predições do classificador usando diversas métricas como precisão, falsos e verdadeiros positivos e outros. Cria um arquivo com os resultados escritos em formato JSON. Ganha acesso ao spark por meio do dataframe recebido na porta de entrada.

4 EXPERIMENTAÇÃO E RESULTADOS

Para determinar os valores de parâmetros que encontram melhores resultados a experimentação foi dividida em duas etapas. Na primeira, foi feita uma varredura nos parâmetros de forma esparsa, fazendo grandes incrementos dos parâmetros para encontrar as faixas de valores problemáticos. A segunda etapa foi então direcionada para as faixas de valores mais promissoras para a obtenção de resultados favoráveis.

4.1 PARÂMETROS E MÉTRICAS

Os parâmetros variados foram a profundidade máxima da árvore de decisão, o ganho mínimo de informação necessário para expansão de um nó da árvore e a impureza utilizada, conforme explicados na seção 2.7. Foram utilizadas as métricas de precisão, revocação, falsos positivos e verdadeiros positivos. Falsos positivos são elementos não relevantes que foram selecionados, verdadeiros positivos são elementos relevantes que foram selecionados, precisão é a razão entre elementos relevantes selecionados sobre o total de elementos selecionados e revocação é a razão entre elementos relevantes selecionados sobre o total de elementos relevantes. Essas métricas foram aplicadas sobre os fluxos de rede normal, de fundo e de bots, onde fluxo normal é de máquinas confiáveis, o de fundo é de máquinas não confiáveis e o de bot é de máquinas infectadas.

4.2 METODOLOGIA

Para a primeira etapa da experimentação, foram feitos diversos testes da seguinte maneira: Primeiro, foram variados conjuntamente os parâmetros de profundidade máxima e ganho mínimo para a impureza Gini, onde logo foi constatado que valores maiores que 0.012 para o ganho mínimo zeram a classificação do fluxo de bots, independente da profundidade máxima da árvore, tornando-se esse então o valor limite para as iterações seguintes. Em seguida foram variados esses mesmos parâmetros para a impureza do tipo entropia, onde se encontrou outro valor limite para o ganho mínimo, dessa vez de 0.039 .

4.3 RESULTADOS

Os resultados encontrados em todos os processos estão num arquivo complementar anexo ao trabalho. Esta opção foi feita pelo volume de dados e o tamanho das tabelas relativas a eles ser bastante grande para ser incorporado no escopo do relatório. Os dados mais relevantes e de amostragem são referenciados no âmbito do texto através de imagens de mapas de calor utilizadas na seção de estatísticas.

5 ESTATÍSTICAS

5.1 ANÁLISE DOS DADOS REFINADOS

Os dados que melhor descrevem as informações de desempenho do workflow são os relativos a classe bot e os dados gerais. Especificamente, para os bots, a melhor métrica a ser analisada é a revocação, que diz quantos dados relevantes foram selecionados. Para os dados gerais, que medem o desempenho do workflow como um todo em identifcar as 3 classes adotadas, normal, fundo e bot, a melhor métrica a ser avaliada é a da precisão, que mede a relevância dos dados selecionados nas diversas classes. Porém, este fator está bastante orientado aos dados da classe com maior presença. No escopo deste trabalho foram levantadas as métricas descritas na seção resultados para todas as classes, porém a análise de resultados foi feita somente nas classes acima citadas com as métricas descritas acima para cada uma delas. A análise de dados foi efetuada sobre os dados refinados resultados da seleção de parâmetros para cada uma das medidas de impureza adotadas. Para os dados relativo à entropia, os limites adotados foram de 15 a 30 com passo de 3 para a profundidade máxima da árvore e de ganho mínimo de informação de 0 até 0.018 com passo de 0.002. Os melhores resultados são mostrados no mapa de calor das figuras 5.1 e 5.2.

A análise dos mapas revela uma consistência entre eles, o que mostra que a classe geral descreve de uma forma bastante similar o comportamento relativo do que acontece na classe Bot. Além disso, para mesmos valores de profundidade, quando menos restrito é o ganho mínimo de informação, melhor são os resultados, o que também era esperado, uma vez que se restringe menos os critério de seleção de parâmetros de cada nó. Outra consistência observada é a inversão do mapa de falso positivos quando comparado com o mapa das outras métricas. Isso é relevante no sentido que deseja-se que parametrizações boas capturem poucos falsos positivos. Outra observação a ser feita é a de que as métricas possuem comportamentos similares para as mesmas parametrizações. Na utilização da função entropia têm-se novamente o comportamento esperado de melhores resultados para menores restrições em ganho mínimo de informação para mesma profundidade. As figuras 5.7 e 5.8 mostram os valores de desvio padrão e média para as classes. Como supramencionado, os valores mais importantes para as classes geral e bot são respectivamente os de precisao e revocação. Os valores de precisão ficaram melhores para a classe

	A	B	C	D	E	F	G	H	I
1	Iteração	Profundidade Máxima	Ganho Mínimo	Impuridade	Geral/F1	Geral/Revocação	Geral/VerdadeiroPositivo	Geral/Precisão	Geral/FalsoPositivo
2	1	15	0	entropy	0.942591083	0.945641733	0.945641733	0.942084869	0.280542516
3	2	15	0.002	entropy	0.941713006	0.945118931	0.945118931	0.941275029	0.29019706
4	3	15	0.004	entropy	0.941111073	0.944520176	0.944520176	0.940362097	0.29976536
5	4	15	0.006	entropy	0.941313188	0.9441819	0.9441819	0.9407242	0.285562932
6	5	15	0.008	entropy	0.942339459	0.945102482	0.945102482	0.942193911	0.273897502
7	6	15	0.01	entropy	0.939431301	0.942832292	0.942832292	0.938578725	0.310264625
8	7	15	0.012	entropy	0.936522003	0.939531722	0.939531722	0.935934499	0.305952545
9	8	15	0.014	entropy	0.938778471	0.942233876	0.942233876	0.937879672	0.313424956
10	9	15	0.016	entropy	0.936721126	0.941120858	0.941120858	0.93555141	0.330982426
11	10	15	0.018	entropy	0.935097226	0.938751998	0.938751998	0.934789773	0.307147419
12	11	18	0.02	entropy	0.930234968	0.937821441	0.937821441	0.922999347	0.357608314
13	12	18	0.022	entropy	0.92939864	0.937277617	0.937277617	0.934175993	0.372037118
14	13	18	0.024	entropy	0.929755152	0.936376592	0.936376592	0.931237388	0.330635774
15	14	18	0.026	entropy	0.930590009	0.937548523	0.937548523	0.933213905	0.340446198
16	15	18	0.028	entropy	0.927436445	0.934735909	0.934735909	0.926628768	0.362089918
17	16	18	0.03	entropy	0.90650188	0.924232917	0.924232917	0.906777742	0.608128372
18	17	18	0.032	entropy	0.905955571	0.92372581	0.92372581	0.905692054	0.608819991
19	18	18	0.034	entropy	0.905315232	0.922854427	0.922854427	0.904137288	0.60537358
20	19	18	0.036	entropy	0.90563345	0.92286286	0.92286286	0.904042192	0.602009839
21	20	18	0.038	entropy	0.905141202	0.922928076	0.922928076	0.904460145	0.607881899
22	21	21	0	entropy	0.94548621	0.947615702	0.947615702	0.94535828	0.251574923
23	22	21	0.002	entropy	0.944832332	0.9471676	0.9471676	0.944268724	0.264017374
24	23	21	0.004	entropy	0.945047724	0.947486991	0.947486991	0.945247207	0.255305345
25	24	21	0.006	entropy	0.944361288	0.946436093	0.946436093	0.944228625	0.256427461
26	25	21	0.008	entropy	0.939290671	0.94263671	0.94263671	0.93816535	0.309076559
27	26	21	0.01	entropy	0.938553661	0.942210555	0.942210555	0.937738142	0.310746607
28	27	21	0.012	entropy	0.935923119	0.939365461	0.939365461	0.935049716	0.320980117
29	28	21	0.014	entropy	0.937823178	0.941711729	0.941711729	0.936656635	0.332747239
30	29	21	0.016	entropy	0.935527878	0.937926048	0.937926048	0.935232374	0.287778413
31	30	21	0.018	entropy	0.928534309	0.936353747	0.936353747	0.928261072	0.371217991
32	31	24	0.02	entropy	0.93011221	0.937014209	0.937014209	0.930566865	0.337188132
33	32	24	0.022	entropy	0.930934889	0.938033077	0.938033077	0.931997204	0.342795959

FIG. 5.1: Mapa de calor para classe Geral com impureza do tipo Entropia

	A	B	C	D	E	F	G	H	I
1	Iteração	Profundidade Máxima	Ganho Mínimo	Impuridade	bot/Revocação	bot/F1	bot/VerdadeiroPositivo	bot/Precisão	bot/FalsoPositivo
2	1	15	0	entropy	0.756931658	0.762756	0.756931658	0.76867026	0.022108115
3	2	15	0.002	entropy	0.746967628	0.75904	0.746967628	0.771509168	0.021713421
4	3	15	0.004	entropy	0.731283566	0.752657	0.731283566	0.775316726	0.020799401
5	4	15	0.006	entropy	0.750777287	0.754419	0.750777287	0.758096497	0.023078257
6	5	15	0.008	entropy	0.765443103	0.761965	0.765443103	0.758518241	0.02371315
7	6	15	0.01	entropy	0.716228615	0.739239	0.716228615	0.763776582	0.021585988
8	7	15	0.012	entropy	0.726263504	0.729084	0.726263504	0.731925486	0.025823996
9	8	15	0.014	entropy	0.714853573	0.73756	0.714853573	0.76175657	0.021721654
10	9	15	0.016	entropy	0.701302217	0.731917	0.701302217	0.765326321	0.020912813
11	10	15	0.018	entropy	0.735475088	0.730305	0.735475088	0.725208054	0.026943649
12	11	18	0.02	entropy	0.695128198	0.718239	0.695128198	0.742940295	0.023429826
13	12	18	0.022	entropy	0.676533514	0.710162	0.676533514	0.747307646	0.022269619
14	13	18	0.024	entropy	0.730449612	0.722712	0.730449612	0.715137542	0.028296597
15	14	18	0.026	entropy	0.718866843	0.721851	0.718866843	0.724861027	0.02626277
16	15	18	0.028	entropy	0.689500175	0.705537	0.689500175	0.722338146	0.025885093
17	16	18	0.03	entropy	0.372668958	0.516486	0.372668958	0.841060412	0.006836412
18	17	18	0.032	entropy	0.37254902	0.514764	0.37254902	0.832595915	0.007256067
19	18	18	0.034	entropy	0.375764928	0.514487	0.375764928	0.815575449	0.008282879
20	19	18	0.036	entropy	0.380011293	0.517086	0.380011293	0.808847937	0.008764334
21	20	18	0.038	entropy	0.373421138	0.514111	0.373421138	0.824897156	0.007728947
22	21	21	0	entropy	0.786716684	0.775612	0.786716684	0.764817001	0.023503377
23	22	21	0.002	entropy	0.769433962	0.77118	0.769433962	0.772934041	0.022015121
24	23	21	0.004	entropy	0.786348862	0.775024	0.786348862	0.764020106	0.023604139
25	24	21	0.006	entropy	0.778808304	0.768889	0.778808304	0.759218861	0.024059396
26	25	21	0.008	entropy	0.718720501	0.741345	0.718720501	0.765439191	0.021458867
27	26	21	0.01	entropy	0.725049751	0.741493	0.725049751	0.758700365	0.022234079
28	27	21	0.012	entropy	0.708404663	0.724106	0.708404663	0.740519813	0.024019184
29	28	21	0.014	entropy	0.688947127	0.728301	0.688947127	0.772422985	0.019722939
30	29	21	0.016	entropy	0.7482525	0.731574	0.7482525	0.71562275	0.029070761
31	30	21	0.018	entropy	0.679536156	0.708263	0.679536156	0.739525284	0.023221153
32	31	24	0.02	entropy	0.72138054	0.722415	0.72138054	0.723451507	0.026829467
33	32	24	0.022	entropy	0.712980977	0.723886	0.712980977	0.735129412	0.025001118

FIG. 5.2: Mapa de calor para a classe Bot com impureza do tipo Entropia

Geral, o que se deve ao grande volume de dados de fundo que são detectados e acabam por influenciar o valor desta medida.

	Revocação	F1	VerdadeiroPositivos	Precisão	FalsoPositivos
media	0.9289	0.9368	0.9368	0.9285	0.3870
desvio	0.0125	0.0044	0.0044	0.0127	1.0455

FIG. 5.3: Média e desvio padrão das métricas da classe Geral com impureza do tipo Entropia.

	Revocação	F1	VerdadeiroPositivos	Precisão	FalsoPositivos
media	0.6391	0.6822	0.6391	0.7697	0.0196
desvio	0.1582	0.5935	0.1582	0.0919	0.0032

FIG. 5.4: Média e desvio padrão das métricas da classe Bot com impureza do tipo Entropia.

Para os dados relativos ao índice gini foram realizadas rodadas de teste dentro dos limites de 5 a 17 para a profundidade máxima com passo 3 e de 0 até a.011 para o ganho mínimo de informação com passo 0.001. Novamente, os melhores resultados são mostrados pelos mapas de calor das figuras 5.5 e 5.6.

	A	B	C	D	E	F	G	H	I
1	Iteração	Profundidade Máxima	Ganho Mínimo	Impuridade	Geral/F1	Geral/Revocação	Geral/VerdadeiroPositivo	Geral/Precisão	Geral/FalsoPositivo
37	36	11	0.011	gini	0.9002	0.9204	0.9204	0.9025	0.6473
38	37	11	0.012	gini	0.9009	0.9213	0.9213	0.9048	0.6477
39	38	14	0	gini	0.9440	0.9467	0.9467	0.9436	0.2700
40	39	14	0.001	gini	0.9384	0.9424	0.9424	0.9378	0.3285
41	40	14	0.002	gini	0.9373	0.9422	0.9422	0.9366	0.3548
42	41	14	0.003	gini	0.9217	0.9320	0.9320	0.9260	0.4844
43	42	14	0.004	gini	0.9227	0.9329	0.9329	0.9257	0.4802
44	43	14	0.005	gini	0.9003	0.9210	0.9210	0.9046	0.6503
45	44	14	0.006	gini	0.9017	0.9220	0.9220	0.9060	0.6457
46	45	14	0.007	gini	0.8996	0.9208	0.9208	0.9053	0.6542
47	46	14	0.008	gini	0.9011	0.9217	0.9217	0.9062	0.6483
48	47	14	0.009	gini	0.9006	0.9212	0.9212	0.9051	0.6469
49	48	14	0.01	gini	0.9005	0.9211	0.9211	0.9047	0.6481
50	49	14	0.011	gini	0.9007	0.9214	0.9214	0.9054	0.6500
51	50	14	0.012	gini	0.9013	0.9217	0.9217	0.9055	0.6477
52	51	17	0	gini	0.9475	0.9497	0.9497	0.9472	0.2522
53	52	17	0.001	gini	0.9390	0.9428	0.9428	0.9383	0.3266
54	53	17	0.002	gini	0.9378	0.9426	0.9426	0.9372	0.3485
55	54	17	0.003	gini	0.9246	0.9344	0.9344	0.9282	0.4661

FIG. 5.5: Mapa de calor para classe Geral com impureza do tipo Gini

Neste contexto, valores maiores de profundidade obtiveram melhores resultados para mesmos valores de ganho de informação e dentro da mesma profundidade o comportamente esperado do ganho mínimo de informação se repetiu, isto é, para valores mais relaxados os resultados foram melhores. Verificam-se ainda as consistências esperadas entre as métricas e entre as classes o que traduz se finalmente pela consistência existente

	A	B	C	D	E	F	G	H	I
1	Iteração	Profundidade Máxima	Ganho Mínimo	Impuridade	bot/Revocação	bot/F1	bot/VerdadeiroPositivo	bot/Precisão	bot/FalsoPositivo
36	35	11	0.01	gini	0.3192	0.4632	0.3192	0.8443	0.0057
37	36	11	0.011	gini	0.3233	0.4665	0.3233	0.8375	0.0061
38	37	11	0.012	gini	0.3220	0.4692	0.3220	0.8640	0.0049
39	38	14	0	gini	0.7681	0.7674	0.7681	0.7667	0.0226
40	39	14	0.001	gini	0.6961	0.7322	0.6961	0.7724	0.0200
41	40	14	0.002	gini	0.6659	0.7233	0.6659	0.7916	0.0170
42	41	14	0.003	gini	0.5225	0.6270	0.5225	0.7838	0.0139
43	42	14	0.004	gini	0.5260	0.6341	0.5260	0.7981	0.0130
44	43	14	0.005	gini	0.3197	0.4678	0.3197	0.8716	0.0046
45	44	14	0.006	gini	0.3245	0.4732	0.3245	0.8735	0.0046
46	45	14	0.007	gini	0.3137	0.4635	0.3137	0.8872	0.0039
47	46	14	0.008	gini	0.3208	0.4702	0.3208	0.8802	0.0043
48	47	14	0.009	gini	0.3223	0.4710	0.3223	0.8742	0.0045
49	48	14	0.01	gini	0.3214	0.4692	0.3214	0.8687	0.0047
50	49	14	0.011	gini	0.3193	0.4678	0.3193	0.8746	0.0045
51	50	14	0.012	gini	0.3222	0.4702	0.3222	0.8692	0.0047
52	51	17	0	gini	0.7845	0.7837	0.7845	0.7830	0.0211
53	52	17	0.001	gini	0.6989	0.7311	0.6989	0.7665	0.0204
54	53	17	0.002	gini	0.6748	0.7284	0.6748	0.7912	0.0173
55	54	17	0.003	gini	0.5445	0.6479	0.5445	0.7996	0.0133

FIG. 5.6: Mapa de calor para a classe Bot com impureza do tipo Gini

entre os valores médios e de desvio para as diversas métricas desta classe apresentadas nas figuras 5.3 e 5.4.

	Revocação	F1	VerdadeiroPositivos	Precisão	FalsoPositivos
media	0.9119	0.9272	0.9272	0.9143	0.5510
desvio	0.0133	0.0045	0.0045	0.0115	0.9177

FIG. 5.7: Média e desvio padrão das métricas da classe Geral com impureza do tipo Gini.

	Revocação	F1	VerdadeiroPositivos	Precisão	FalsoPositivos
media	0.4389	0.5529	0.4389	0.8217	0.0102
desvio	0.1572	0.6770	0.1572	0.1047	0.0021

FIG. 5.8: Média e desvio padrão das métricas da classe Bot com impureza do tipo Gini.

5.2 CONCLUSÃO

Os resultados obtidos durante a experimentação foram coerentes com o esperado, desta forma o trabalho cumpriu seu papel no que diz respeito à experimentação envolvida. Além disso, mostrou a força da integração entra as ferramentas utilizadas, Spark e Vistrails, e a viabilidade de um framework de gerenciamento de workflows para a utilização como ferramenta de descobrimento de conhecimento na área de segurança cibernética, seja pela possibilidade de captura de dados neste ambiente, ou pela facil visualiação e comparação entre os experimentos.

Como trabalhos futuros, cita-se a realização de refinamentos estatísticos nos experimentos, bem como a utilização de métodos não empíricos para a escolha de rodadas e conjunto de valores para os parâmetros, como por exemplo, a utilização de algorítmos que decidam sobre qual é o melhor conjunto de valores para variar cada parâmetro de decisão. Também cita-se a utilização de outros algorítmos de aprendizado de máquina para comparação de eficiência entre eles.

Por fim, no contexto de segurança cibernética, a pesquisa mostra o potêncial de atuação deste tipo de metodologia, e da utilização de sistemas de gerenciamento de workflow neste contexto, como importante ferramenta para a detecção de ataques de botnets.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AMINI, P.; ARAGHIZADEH, M. A. ; AZMI, R. A survey on botnet: Classification, detection and defense. In: ELECTRONICS SYMPOSIUM (IES), 2015 INTERNATIONAL, none., 2015. **Anais**... [S.l.: s.n.], 2015, p. 233–238. Disponível em: <https://www.researchgate.net/publication/291698628_A_survey_on_Botnet_Classification_de Acesso em: 18 de Maio de 2017.

AVIV, A. J.; HAEBERLEN, A. Challenges in experimenting with botnet detection systems.. In: CSET, none.., 2011. **Anais**... [S.l.: s.n.], 2011. Disponível em: <none>. Acesso em: 18 de Maio de 2017.

BARBOSA, KAIO R. S., ET AL. Botnets: Características e Métodos de Detecção Atraves do Tráfego de Rede. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/sbseg/2014/0036.pdf>. Acesso em: 15 de Julho de 2017.

BRAMER, M. **Principles of data mining**. London: Springer, 2007.

COALITION, W. M. **The workflow reference model**. [S.l.: s.n.], 1996. (Relatório Técnico, WFMC-TC00-1003).

CRUZ JÚNIOR, S. C. D. A segurança e defesa cibernética no brasil e uma revisão das estratégias dos estados unidos, rússia e índia para o espaço virtual. **Ipea**, v. none, 2013. Disponível em: <http://hdl.handle.net/11058/1590>. Acesso em: 18 de Maio de 2017.

GAO. Cybersecurity Actions Needed to Address Challenges Facing Federal Systems. Disponível em: <http://www.gao.gov/assets/670/669810.pdf>. Acesso em: 10 de Maio de 2017.

GARCIA, S.; GRILL, M.; STIBOREK, J. ; ZUNINO, A. An empirical comparison of botnet detection methods. **computers & security**, v. 45, p. 100–123, 2014.

GARCÍA, S.; ZUNINO, A. ; CAMPO, M. Survey on network-based botnet detection methods. **Security and Communication Networks**, v. 7, n. 5, p. 878–903, 2014.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. 2. ed. San Francisco: Morgan Kaufmann Publishers, 2006. 772 p.

PAIM, RAFAEL, E. A. **Gestão de processos - pensar, agir e aprender**. [S.l.]: Bookman, 2009. 328 p.

PÁDUA, SILVIA INÊS DALLAVALLE DE, ET AL. Sistema de Gerenciamento de Workflow: um overview e um estudo de caso. Disponível em: <http://www.abepro.org.br/biblioteca/enegep2003_tr0905_0435.pdf>. Acesso em: 1 de Agosto de 2017.

QUINLAN, J. R. **Decision trees and multivalued attributes**. New York: Oxford University Press, 1988. 318 p.

SALIMFARD, K.; WRIGHT, M. Petri net based modelling of workflow systems: an overview. **European Journal of Operational Research**, v. 134, 2001. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221700002927?viaAcesso em: 1 de Agosto de 2017.

KEEPER SECURITY. The 2016 State of SMB Cybersecurity. Disponível em: <https://signup.keepersecurity.com/state-of-smb-cybersecurity-report/>. Acesso em: 10 de Maio de 2017.

CYBERSECURITY VENTURES. Cybersecurity Ventures predicts cybercrime will cost the world in excess of $6 trillion annually by 2021. Disponível em: <http://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/>. Acesso em: 10 de Maio de 2017.

ZUBEN, FERNANDO J. VON, ET AL. Árvores de decisão. Disponível em: <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia004_1s10/notas_de_aula/topico7_IA004_1
Acesso em: 22 de Junho de 2017.