Universidade Estadual de Londrina

GABRIEL JESUS VASQUEZ ALTAMIRANO

UTILIZAÇÃO DE CLASSIFICADORES SUPERVISIONADOS PARA DETECÇÃO DE INTRUSÕES EM REDES INDUSTRIAIS

Londrina
2018

GABRIEL JESUS VASQUEZ ALTAMIRANO

UTILIZAÇÃO DE CLASSIFICADORES SUPERVISIONADOS PARA DETECÇÃO DE INTRUSÕES EM REDES INDUSTRIAIS

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Bruno Bogaz Zarpelão.

Londrina
2018

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Vasquez, Gabriel.
Utilização de Classificadores Supervisionados para Detecção de Intrusões em Redes Industriais / Gabriel Vasquez. - Londrina, 2018.
71 f. : il.

Orientador: Bruno Bogaz Zarpelão.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2018.
Inclui bibliografia.

1. Classificação Supervisionada - Tese. 2. Redes Industriais - Tese. I. Zarpelão, Bruno Bogaz. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

GABRIEL JESUS VASQUEZ ALTAMIRANO

UTILIZAÇÃO DE CLASSIFICADORES SUPERVISIONADOS PARA DETECÇÃO DE INTRUSÕES EM REDES INDUSTRIAIS

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Bruno Bogaz Zarpelão
Universidade Estadual de Londrina - UEL

Prof. Dr. Mario Lemes Proença Junior
Universidade Estadual de Londrina - UEL

Prof. Dr. Alexandre de Aguiar Amaral
Instituto Federal Catarinense - IFC

Londrina, 11 de setembro de 2018.

AGRADECIMENTOS

À minha esposa Joana Sanches Justo, que pacientemente me apoiou e compreendeu minha falta em todos os eventos e compromissos em que optei por não ir para que pudesse me dedicar aos estudos.

Ao meu amigo Helder Carlo Belan, amigo que me acompanha desde a graduação, e que me ajudou de inúmeras maneiras desde as primeiras ideias sobre o que é o mestrado.

To my dearest friend Theodorus Ernst Kleinhans, a.k.a "José", who passed away while the obtaining of this degree, you are truly missed here.

Ao Eduardo Alves de Moraes, amigo desde as primeiras aulas como alunos especiais, sempre conseguiu manter o espírito alegre e animado enquanto estivemos nas trincheiras desta luta.

Aos meus pais, aos amigos e familiares pelo apoio durante este período.

Ao meu orientador Professor Dr. Bruno Bogaz Zarpelão, que sempre teve uma paciência imensa e me ajudou a superar muitas dificuldades nesta investida na vida acadêmica.

Aos professores membros da banca Prof. Dr. Mario Lemes Proença Junior e Prof. Dr. Alexandre de Aguiar Amaral, que foram fundamentais para nortear este trabalho.

Ao Departamento de Computação da Universidade Estadual de Londrina por todo o apoio oferecido durante o programa de mestrado.

Obrigado

ALATAMIRANO, Gabriel Jesus Vasquez. **Utilização de classificadores supervisionados para detecção de intrusões em redes industriais.** 2018. 71 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2018.

RESUMO

Ataques recentes a redes industriais têm trazido à tona questões sobre como protegê-las. Essas redes são essenciais para o controle de muitos aspectos de nossa vida cotidiana, como o abastecimento de água, o fornecimento de energia elétrica e gás, etc. Neste trabalho, realizamos um estudo sobre a utilização de algoritmos de classificação supervisionada na detecção de intrusões em redes industriais. Para tanto, foi proposto um modelo de detecção de intrusões que prevê a utilização de fluxos IP gerados a partir de pacotes coletados de uma rede industrial. Estudamos o desempenho de nove algoritmos de classificação supervisionada, pertencentes às famílias dos classificadores de árvore de decisão, dos classificadores discriminativos e dos classificadores estatísticos. Considerando que no tráfego de redes haverá uma quantidade significativamente menor de tráfego malicioso em comparação com tráfego legítimo, teremos o desbalanceamento entre essas duas classes como uma das características chave deste tráfego. A avaliação dos classificadores se deu utilizando métricas apropriadas para lidar com esse desbalanceamento como o f1 score, acurácia média, curva ROC (Receiver Operating Characteristic), curva PR (Precision- Recall) e áreas sob as curvas ROC e PR. Os resultados apontaram que os algoritmos pertencentes à família das árvores de decisão apresentaram um desempenho superior aos demais, sendo o melhor resultado obtido pelo algoritmo Boosted Decision Tree.

Palavras-chave: Detecção de Intrusões. Ataques. Detecção de Anomalias. Desbalanceamento de classe. SCADA. Modbus.

ALATAMIRANO, Gabriel Jesus Vasquez. **Using supervised learning for intrusion detection in industrial networks.** 2018. 71 p. Dissertation (Master's Degree in Computer Science) – Universidade Estadual de Londrina, Londrina, 2018.

ABSTRACT

Recent attacks on industrial networks have raised questions about how to protect them. These networks are essential for the control of many aspects of our daily lives, such as water supply, electricity and gas, etc. In this work, we studied the utilization of supervised classification algorithms for the intrusion detection in industrial networks. Therefore, we propose a model for intrusion detection that makes use of IP flows generated from packets collected from an industrial network. We study the performance of nine supervised classification algorithms, belonging to the families of the decision tree classifiers, the discriminative classifiers and the statistical classifiers. As network traffic usually has much less malicious traffic than normal traffic, the imbalance between these two classes is one of the key features of this study. The evaluation of the classifiers was based on appropriate metrics to deal with this unbalance such as f1 score, medium accuracy, ROC curve (Receiver Operating Characteristic), PR curve (Precision Recall) and areas under the ROC and PR curves. The results showed that the algorithms belonging to the family of decision trees presented a superior performance to the others, being the best result obtained by the algorithm Boosted Decision Tree.

Keywords: Intrusion Detection. Attack. Anomaly Detection. Class Imbalance. SCADA. Modbus.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AUPR	Area under Precision-Recall Curve
AUROC	Area under ROC Curve
ARFF	Attribute-Relation File Format
CI	Critical Infrastructure
CLI	Command Line Interface
CSV	Comma Separated Values
CCL	Closed Control Loop
DNP3	Distributed Network Protocol
FP	False Positive
FPR	False Positive Rate
FN	False Negative
HIDS	Host-based Intrusion Detection System
HMI	Human Machine Interface
ICS	Industrial Control System
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IoT	Internet of Things
IP	Internet Protocol
IPFIX	IP Flow Information Export
KDD	Knowledge Discovery in Databases
k-NN	k-Nearest Neighbor
LD-SVM	Locally Deep Support Vector Machine
MAC	Media Access Control
Modbus	Modicon Communication Bus

MTU	Master Terminal Unit
NIDS	Network-based Intrusion Detection System
OCSVM	One-Class Support Vector Machine
OSI	Open System Interconnection
PCAP	Packet Capture
PHP	PHP: Hypertext Preprocessor
PLC	Programmable Logical Controller
Profibus	Process Field Bus
PR Curve	Precision-Recall Curve
RFC	Request for Comments
ROC Curve	Receiver Operating Characteristic Curve
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SVM	Support Vector Machine
S7Comm	S7 Communication
SWaT	Secure Water Treatment
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TP	True Positive
TPR	True Positive Rate
TN	True Negative
TNR	True Negative Rate
US ICS-CERT	Industrial Control Systems Cyber Emergency Response Team

SUMÁRIO

1 INTRODUÇÃO

Sistemas SCADA (*Supervisory Control and Data Acquisition*) são sistemas que controlam e monitoram vários processos dentro de uma grande variedade de indústrias e órgãos governamentais, e fazem parte dos chamados Sistemas de Controle Industriais (ICS - *Industrial Control Systems*). Nesse contexto, existem aqueles sistemas que, devido à sua grande importância, são chamados de infraestrutura crítica (CI - *Critical Infrastructure*), haja vista que falhas nesse tipo de sistema podem causar graves prejuízos financeiros ou ambientais, afetando uma grande quantidade de usuários. Alguns exemplos de sistemas de infraestrutura crítica, responsáveis pelas infraestruturas que controlam aspectos fundamentais de nossa vida cotidiana, são aqueles utilizados por concessionárias de distribuição de energia elétrica, água e gás. O problema é que os ICSs, quando expostos à Internet, podem não possuir as proteções apropriadas para se protegerem de ataques provenientes de fora da rede, como apontado por Knapp e Langill [1].

Conforme os ICS vão sendo progressivamente conectados à Internet, os riscos enfrentados pelas CI tendem a crescer ainda mais. Há uma série de vantagens na interconexão desses dispositivos com a Internet, como uma maior escalabilidade, interoperabilidade entre sistemas, acesso remoto, etc., contudo, o foco dos ICS sempre foi a disponibilidade e não a questão da segurança [2]. Assim sendo, ataques contra sistemas SCADA podem afetar uma grande parcela da população, interrompendo uma série de serviços essenciais ao cotidiano. Um termo normalmente associado aos ICS e aos sistemas SCADA que os compõe é o de redes industriais, como sendo aquelas que utilizam protocolos industriais de comunicação entre seus dispositivos.

Para que o administrador de redes possa prevenir e proteger esse tipo de sistema contra ações maliciosas, é necessário aprimorar o conjunto de defesas aplicado a essas redes, seguindo o princípio chamado de defesa em profundidade (*defense-in-depth*) [1][3]. No conceito de defesa em profundidade, múltiplos mecanismos de proteção são associados em camadas de modo a aumentar a segurança de uma rede. Dessa maneira, o IDS (*Intrusion Detection System* - Sistema de Detecção de Intrusões) surge como uma das camadas de proteção, sendo responsável por monitorar o tráfego de rede em busca de detectar intrusões, reportando ao administrador os dados encontrados para que ele possa tomar as ações apropriadas. Um dos pontos mais relevantes em um IDS é a confiabilidade dos dados reportados ao administrador [4] [5].

É importante salientar que o tráfego gerado por redes industriais possui características que o difere do tráfego de redes corporativas. No tráfego de sistemas SCADA, há uma repetição de padrões de comunicação relacionado a processos automatizados de supervisão e aquisição de dados denominados *polling*, uma característica presente em re-

des industriais, mas inexistente em redes corporativas. De acordo com Mitchel e Chen [4], o processo de *polling* também pode ser referenciado pelo termo *Closed Control Loop* (CCL), sendo apenas outra designação para a mesma ação. Em redes industriais, o fato de o tráfego restringir-se a poucos protocolos de comunicação industrial e apresentar transações repetitivas devido ao *polling* facilita a diferenciação entre tráfego normal e tráfego anômalo [6]. Tráfego anômalo é aquele que se difere ou se distancia do comportamento observado a partir do tráfego normal.

Na literatura, outros trabalhos estudaram o uso de IDS para redes industriais, como o trabalho de Linda *et al.* [7], que propõe o uso de redes neurais para detecção de anomalias; o trabalho de Yang *et al.* [8], que desenvolveu um IDS híbrido baseado em assinatura e especificação para detectar anomalias de um conjunto real; o trabalho de Junejo e Goh [9], que desenvolveram um IDS para detecção de anomalias em um conjunto de dados próprio; o trabalho de Udd *et al.* [10], que propõe um IDS baseado em assinaturas, dentre outros autores. Porém os estudos mencionados deixam de abordar questões pertinentes à detecção de intrusões em redes industriais, tais como o desbalanceamento normalmente encontrado entre o tráfego malicioso e o tráfego legítimo, e a utilização de conjuntos de dados públicos nos testes.

O objetivo deste trabalho é estudar a utilização de algoritmos de classificação supervisionada na detecção de intrusões em redes industriais. Para tanto, foi proposto um modelo que prevê a utilização de fluxos IP gerados a partir de pacotes coletados desde uma rede industrial. A seguir, são extraídas características desses fluxos IP de maneira que informações sintetizadas sobre o tráfego sejam apresentadas como entradas para o algoritmo de classificação supervisionada. O algoritmo analisa e retorna uma classificação entre tráfego legítimo e tráfego malicioso. Dentro do escopo do trabalho, foram avaliados nove algoritmos de aprendizado supervisionado, pertencentes às famílias dos classificadores discriminativos, classificadores de árvore de decisão e classificadores estatísticos, utilizando um conjunto de dados disponível publicamente. Além disso, foi estudada a questão do desbalanceamento entre classes resultante do agrupamento de pacotes de rede em fluxos IP. A avaliação dos algoritmos se deu utilizando validação cruzada juntamente com um conjunto de métricas com foco no desbalanceamento entre classes. Os resultados obtidos demonstraram que a família dos classificadores de árvore de decisão, composta por algoritmos do tipo *ensemble*, alcançou o melhor desempenho em termos gerais. Particularmente, o algoritmo *Boosted Decision Tree*, pertencente a essa família, apresentou os melhores resultados.

O restante deste trabalho é dividido da seguinte maneira. No capítulo 2 é apresentado o referencial teórico, composto de fundamentos necessários para a compreensão de tópicos pertencentes às redes industrias e à área de aprendizado de máquina. No capítulo 3 é apresentado o modelo proposto para a utilização de aprendizado de máquina

supervisionado na detecção de intrusões em redes industriais. No capítulo 4 são apresentados os resultados obtidos de experimentações sobre um conjunto de dados disponível publicamente. No capítulo 5 são apresentadas as considerações finais.

2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados os sistemas SCADA e uma introdução aos ataques ciber-físicos e sistemas de deteção de intrusões. Logo após são apresentadas as definições sobre fluxos IP e conjuntos de dados, seguido dos algoritmos de aprendizado de máquina e métodos de classificação utilizados para a geração dos resultados obtidos durante este trabalho. O capítulo se encerra com a apresentação dos estudos relacionados à área.

2.1 Sistemas SCADA

Sistemas SCADA surgiram na década de 1960 e são utilizados para monitorar, gerenciar e controlar processos industriais. Além disso, também são utilizados para monitorar e controlar outros sistemas industriais quando estes estão distribuídos geograficamente, possibilitando assim uma coleta em tempo real de dados e, consequentemente, o gerenciamento desses dispositivos a partir de localidades distintas [11] [12].

A primeira geração dos sistemas SCADA, chamada de sistemas SCADA monolíticos, surgiu na década de 1960 e possuía uma arquitetura centralizada, sendo o *mainframe* o responsável por todo o controle e gerenciamento dos processos de coleta e gerenciamento dos dispositivos. Uma característica dessa primeira geração era a falta de padronização nos protocolos de comunicação. Assim sendo cada fabricante desenvolvia e utilizava códigos proprietários em seus dispositivos, de modo que estes ficavam restritos apenas ao conjunto de funções para os quais eles foram idealizados, sem possibilidade de expansão ou suporte a novas padronizações [11].

Na segunda geração, chamada de sistemas SCADA distribuídos, surgiu na década de 1980 e tem como característica a evolução na arquitetura com a adoção de múltiplos dispositivos interconectados através de uma rede local. Nessa geração, o protocolo de comunicação industrial Modbus (*Modicon Communication Bus*) [13] começa a ser utilizado em maior escala pela indústria, visto que se tornara um protocolo aberto e possuía como foco a confiabilidade, a facilidade de implementação e a manutenção [11] [14].

Em sua terceira geração, chamada de sistemas SCADA em rede surgiu na década de 2000, e possui como característica a evolução dos sistemas industriais para uma arquitetura distribuída, de longo alcance, permitindo a comunicação de redes distintas através da Internet, desta forma mantendo sob um mesmo nível de gerenciamento diversos dispositivos separados por grandes extensões geográficas. Além disso, houve o surgimento de outros protocolos de comunicação abertos, como o DNP3 (*Distributed Network Protocol*)

[15] e o IEC 61850 (*International Electrotechnical Commission*) [16], que disponibilizam, opcionalmente, um elevado grau de segurança [11].

No momento da escrita deste trabalho, ainda estamos em transição para a quarta geração, chamada de Sistemas SCADA IoT (*Internet of Things*). Nesta geração os sistemas evoluíram para o uso de tecnologias em nuvem e de uma maior interoperabilidade entre dispositivos diversos. Consequentemente, alguns elementos pertencentes à segunda e terceira geração ganham mobilidade e fisicamente deixam o ambiente industrial, e suas funções passam a estar acessíveis através da Internet, permitindo uma maior flexibilidade e escalabilidade no controle dos processos industriais [11]. Por outro lado, as facilidades apresentadas por esta nova geração trazem uma série de novos riscos de segurança que necessitam ser avaliados, como abordado por Piggin [17].

Na prática, sistemas SCADA serão compostos essencialmente por cinco elementos: HMI (*Human Machine Interface*), que é a interface utilizada pelo operador do sistema para o monitoramento e controle dos processos industriais; MTU (*Master Terminal Unit*), responsável por manter a informação sobre o funcionamento dos processos físicos através da coleta das informações dos diversos RTU (*Remote Terminal Unit*) da rede e também por encaminhar às RTUs comandos provenientes do HMI; o RTU é o responsável por coletar as informações dos PLC (*Programmable Logical Controller*) e encaminhá-las para o MTU, assim como transmitir informações do MTU aos controladores PLC.

PLCs, o quarto elemento dos sistemas SCADA, são sistemas embarcados, dotados de capacidades de comunicação e utilizados para a execução de algum tipo de tarefa ou monitoramento. Nos estudos de Goetz e Shenoi [18], Goose *et al.* [19] e Díaz [20] é possível notar que alguns autores consideram que as funções do PLC podem ser integradas diretamente no RTU, formando assim apenas um dispositivo físico contendo ambas as funções, sendo esta a abordagem utilizada neste trabalho.

O último elemento do sistema são os sensores e atuadores conectados diretamente aos PLCs. Os sensores são aqueles dispositivos que realizam as leituras, como termômetros, sensores de iluminação, gás, água e pressão. Os atuadores são aqueles dispositivos que efetuam alterações físicas no ambiente, por exemplo válvulas, chaves e disjuntores [11] [4] [21].

Um exemplo de sistema SCADA pode ser visto na Figura 1, onde temos um HMI conectado a um MTU e este conectado a diversos PLC. Cada PLC possui um ou mais sensores e atuadores. Na figura também é possível visualizar a interconexão da rede industrial com a rede corporativa, assim como a interconexão com uma rede industrial remota supervisionada pelo mesmo sistema SCADA. No caso da rede remota, temos a presença do PLC/RTU, no qual ambas funcionalidades são realizadas por um único dispositivo.

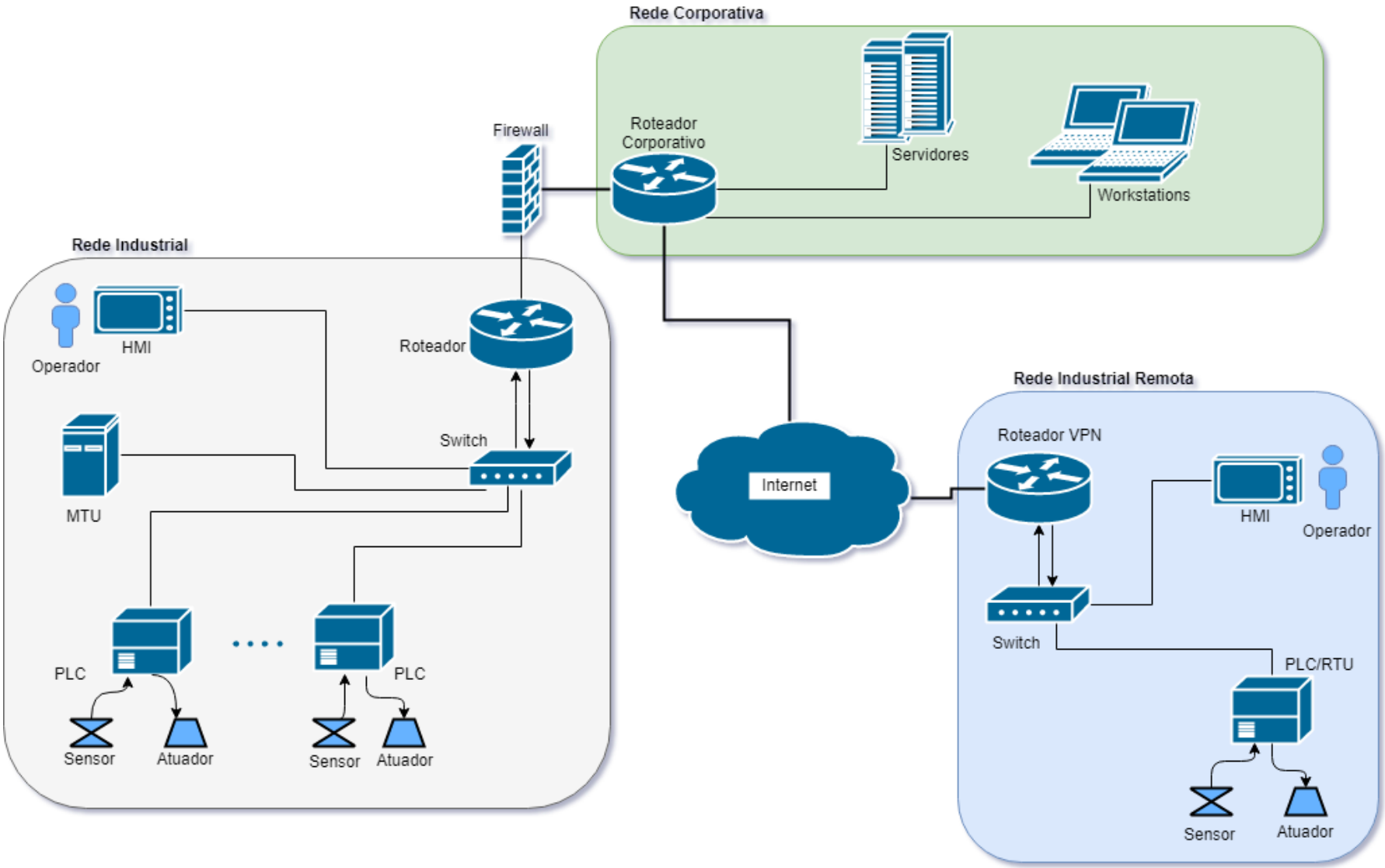

Figura 1 – Exemplo de uma rede SCADA.

Sistemas SCADA utilizam uma série de protocolos de comunicação, como o Modbus [13], o DNP3 [15], o IEC 61850 [16], o Ethernet/IP [22], o Profibus (*Process Field Bus*) [23], o S7Comm (*S7 Communication*) [24], dentre outros. Esses protocolos especificam o meio físico, métodos de comunicação entre os dispositivos e formato do quadro [25]. Dentre os protocolos citados, o Modbus é considerado um dos padrões com maior utilização para comunicações em redes industriais [14], devido principalmente a sua simplicidade e robustez. Foi criado em 1979 pela empresa Modicon, como um protocolo aberto para comunicações seriais nos padrões RS232 ou RS485 [13].

O mecanismo de comunicação no protocolo Modbus [13] é baseado em mensagens de leitura (*read*) e mensagens de escrita (*write*). Na mensagem de leitura, o MTU solicita ao RTU a leitura da informação de um sensor de temperatura, por exemplo. Na mensagem de escrita, o MTU solicita ao RTU que execute uma ação, por exemplo, o fechamento de uma válvula. O processo de *polling*, executado periodicamente pelo MTU para os RTUs, é o responsável pela supervisão e aquisição de dados, onde sucessivos e periódicos processos de leituras e escritas são transmitidos, unidirecionalmente, pela rede. Devido a essa característica, existe uma previsibilidade no comportamento do tráfego em sistemas SCADA [12].

O protocolo Modbus possui duas principais variantes, a Modbus serial [13] e a Modbus TCP/IP [26]. Na variante Modbus serial, a comunicação entre os MTUs e os RTUs é transmitida através de cabos seriais utilizando os protocolos de transmissão RS232 e RS485. Na variante Modbus TCP/IP a comunicação ocorre através de infraestruturas para redes IP, onde o Modbus TCP/IP passa a atuar como um protocolo de camada de aplicação, oferecendo uma conexão na porta TCP 502 [12].

2.2 Ataques Ciber-Físicos

Comparativamente com os sistemas de informação tradicionais, os sistemas SCADA são considerados mais vulneráveis, devido principalmente à dificuldade de aplicar atualizações nos *firmwares* dos equipamentos físicos que compõem essa rede. Essa limitação é consequência de uma das características das redes industriais que é de possuir alta disponibilidade e capacidade de fornecer dados em tempo real, o que dificulta a parada necessária para os processos de atualização e de reinicialização desses equipamentos [5] [8] [25].

Loukas [11] compara diversos aspectos das redes corporativas em relação às redes industriais. Técnicas de criptografia são requeridas nas redes corporativas para a garantia da confidencialidade, integridade e autenticidade. Por outro lado, a criptografia, quando disponibilizada pelo protocolo de comunicação industrial utilizado, não é utilizada em mui-

tas redes industrias por limitações de processamento e consumo em sistemas ciber-físicos. Quanto ao tráfego de dados, é mais difícil definir quais tipos de tráfego são permitidos em uma rede corporativa, enquanto que em uma rede industrial podemos restringir quais tipos de tráfego são esperados desde ou para uma máquina ou, também, quais aplicativos devem ser utilizados.

De acordo com Loukas [11], "Um ataque ciber-físico é uma falha de segurança no ciberespaço que inevitavelmente afeta o espaço físico"[1]. Essa definição apresenta as duas camadas do termo ciber-físico: a primeira remete a sistemas cibernéticos, ou seja, aqueles ligados à Internet, enquanto que a segunda camada remete a equipamentos e processos físicos atuantes no nosso cotidiano. Na prática, essa definição implica em que ações realizadas em um ambiente digital, como é o ciberespaço, podem ter consequências no ambiente físico.

No que tange à questão de ataques voltados a ambientes ciber-físicos, aqui representados pelos sistemas SCADA, temos uma relação direta entre a atratividade do alvo e a criticidade do ambiente controlado. No trabalho de Yussof *et al.* [27], são abordadas as preocupações referentes à privacidade, segurança e impactos financeiros do uso de medidores inteligentes em redes de concessionárias de energia elétrica, um dos muitos possíveis cenários controlados por sistemas SCADA. Uma taxonomia das implicações de privacidade em sistemas cibernéticos é apresentada no trabalho de Toch *et al.* [28], onde os autores levantam uma série de aspectos relevantes sobre riscos e privacidade. Em relação aos ataques, o interesse dos atacantes no alvo estaria na obtenção dos padrões de comportamento dos usuários de uma residência, assim como nas vantagens financeiras que poderiam ser proporcionadas por adulterações nos medidores. No trabalho de Kang *et al.* [25], os autores apresentam as ameaças existentes às redes industriais, assim como preocupações quanto à segurança de sistemas SCADA como possível alvo para ataques terroristas. Esse tipo de preocupação também é recorrente no trabalho de Loukas [11].

De acordo com Huitsing *et al.* [12], a taxonomia dos ataques direcionados, especificamente para o protocolo Modbus [13], pode ser dividida em três categorias: ataques direcionados a falhas nas especificações do protocolo, aqueles nos quais a falta de caracterização do protocolo de comunicação ou suporte a situações não previstas permitirão algum tipo de ataque que afetará todos os dispositivos que utilizam esse protocolo; ataques direcionados à implementação do protocolo pelos fabricantes, aqueles nos quais a implementação errônea ou incompleta do protocolo em algum dispositivo da rede poderá permitir um ataque em dispositivos produzidos por esse fabricante; e, por fim, ataques direcionados à infraestrutura de suporte, nos quais as falhas em outros dispositivos acessórios da rede industrial, como roteadores, *switches* e *access points*, assim como falhas no

[1] Tradução própria de:'A cyber-physical attack is a security breach in cyberspace that adversely affects physical space.'

sistema operacional do HMI, poderão permitir ou facilitar alguma forma de ataque. Nas falhas direcionadas à infraestrutura de suporte podemos também incluir falhas ou falta de proteção nos dispositivos localizados em regiões remotas, que não contam com o mesmo nível de proteção física dos servidores da planta principal, mas que através da rede remota possibilitam acesso à rede industrial principal [2].

Huitsing *et al.* [12] também classificam os ataques em relação às ameaças que podem representar contra a infraestrutura sendo: Interceptação, Interrupção, Modificação e Fabricação de dados. Na Interceptação, há uma alteração na rota que os dados trafegam para que passem por uma rota em que o atacante tenha acesso. Na Interrupção, o fluxo de dados é desviado de seu destino original. Na Modificação, o fluxo de dados é interceptado e modificado antes de chegar a seu destino. Por fim, na Fabricação, um novo fluxo de dados é injetado na rede. Neste contexto, por fluxo se entende como o tráfego passante ao qual o atacante tem acesso.

Como exemplo, um determinado ataque pode estar direcionado a uma falha na implementação do protocolo DNP3 [15], como é o caso do ataque chamado *Rogue Interloper*, um ataque do tipo *man-in-the-middle* em que o atacante se aproveita da falta de obrigatoriedade do uso de protocolos de criptografia entre o MTU e o RTU para modificar os dados enviados entre eles, visando afetar a integridade da rede [12].

A principal consequência desses ataques é que eles possuem a capacidade de perturbar o funcionamento de serviços cotidianos, por exemplo, o de fornecimento de água ou de energia elétrica. Uma análise de incidentes em sistemas industriais, especificamente aqueles voltados a infraestruturas críticas, no período de 1982 a 2014, realizado por Ogie [29], levantou que houve 242 incidentes de segurança reportados. Dentre as informações disponibilizadas no trabalho do autor, pode-se observar que foram registradas 673 mortes como consequência direta de ataques a redes industriais e outras 419 mortes como consequências indiretas desses ataques.

Uma pesquisa envolvendo 599 companhias de 13 países, realizada pelo Instituto Ponemon [30], mostrou que 52% das empresas pesquisadas não estavam cientes de que seus sistemas industriais possuíam algum tipo de vulnerabilidade. De acordo com Piggin [17], a segurança em redes industriais ainda está em seus primórdios, uma vez que os administradores de rede não estão cientes tanto da suscetibilidade a ataques de suas redes quanto dos riscos que estes representam para a continuidade das operações diárias. Desta maneira não estando preparados para a complexidade que os ataques atuais apresentam.

A principal motivação para ataques às redes industriais tem sido a interrupção no fornecimento de serviços, descontinuando momentaneamente o funcionamento de infraestruturas críticas, gerando vantagens econômicas ou políticas aos atacantes. No trabalho de Ogie [29], o autor elenca os principais alvos de ataques sendo sistemas de transporte, concessionárias de energia elétrica e centrais de tratamento de águas.

Apesar de, no ano de 2010, o *malware Stuxnet* ter sido o ataque responsável pela maior visibilidade da área de segurança em redes industriais, também ocorreram outros ataques relevantes, como o Flame [31] no ano de 2012 e mais recentemente ataques como o DragonFly [32] [33] no ano de 2014 e o Industroyer [33] no ano de 2016.

No caso do Stuxnet, que foi o ataque responsável por trazer visibilidade para a questão da segurança em sistemas industriais, temos um *malware*, que utilizando vulnerabilidades na implementação do protocolo S7Comm [24], um protocolo proprietário desenvolvido pelo fabricante Siemens, comprometeu o funcionamento de diversos PLCs, modificando o comportamento de centrífugas utilizadas para enriquecimento de urânio, visando afetar a disponibilidade do sistema alvo.

Para atingir esse objetivo, o Stuxnet se aproveitou de vulnerabilidades não corrigidas do tipo *zero-day* do sistema operacional Microsoft Windows XP utilizado nos HMI. Após sua execução no sistema operacional, o *malware* detectava se a máquina utilizava o *software* Siemens Simatic WinCC, para assim examinar a rede em busca dos PLCs afetados e, em seguida, instalar a modificação nas configurações dos PLCs, os quais passariam a retransmitir informações previamente gravadas sobre o funcionamento da rede em um ataque do tipo *replay*. Enquanto isso, o *malware* modificava os valores de sensores e atuadores, interrompendo o funcionamento da rede. Em suma, o Stuxnet apresenta ataques direcionados a falhas na implementação do protocolo pelos fabricantes e ataques direcionados à infraestrutura de suporte [11] [34] [35].

2.3 Sistemas de Detecção de Intrusões

Os IDS tentam detectar ações, realizadas por um atacante, que possuam como objetivo comprometer os pilares de segurança de uma rede: confidencialidade, integridade e disponibilidade [3] [36].

Dentre esses pilares, a confidencialidade remete a questões de privacidade, ou seja, que os dados trafegados pela rede devem estar restritos apenas àqueles que são seus proprietários. Em uma rede industrial, os dados coletados pelos diversos sensores de campo espalhados pela rede devem ser acessíveis apenas àqueles que controlam essa rede.

O pilar da integridade remete à invulnerabilidade. Desse modo, os dados que trafegam pela rede devem permanecer inalterados desde sua origem até seu destino. Em uma rede industrial, as informações recebidas nas estações de gerenciamento devem representar fidedignamente os dados coletados a partir dos sensores de campo.

Por fim, o pilar da disponibilidade remete à acessibilidade. Neste caso, a rede de dados deverá sempre estar acessível para completar as tarefas necessárias. Em uma rede

industrial, a disponibilidade possibilita que os diversos sensores e atuadores recebam os comandos provenientes de seus controladores.

Em relação aos protocolos de comunicação utilizados em redes industriais, como o Modbus TCP/IP [26], o DNP3 [15], entre outros, é possível observar que eles carecem de mecanismos que auxiliem a garantia da confidencialidade, visto que as mensagens são transmitidas em texto claro, sem o uso de criptografia. Tampouco possuem controles que garantam a integridade, visto que na composição dos cabeçalhos desses protocolos não há campos do tipo *checksum*. Essa função é normalmente delegada ao cabeçalho TCP do pacote. Por fim, não possuem quaisquer mecanismos de autenticação na composição dos pacotes [11].

O papel do IDS na proteção da rede industrial se dá pelo monitoramento do tráfego de rede, a fim de garantir que os três pilares supracitados não sejam comprometidos, notificando o fato ao administrador sobre eventos que possam representar uma tentativa de ataque à rede [1].

A detecção de intrusões em redes industriais pode ser dividida em três tipos: IDS baseados em anomalias, IDS baseados em assinaturas e IDS baseados em especificações [4] [37].

IDS baseados em anomalias, também chamados de IDS baseados em comportamento, tentam detectar intrusões com base em desvios do comportamento normal aprendido previamente a partir da mesma rede. Seu funcionamento depende da criação de um perfil de comportamento normal a partir de amostras do tráfego legítimo. Esse perfil criado servirá de base para a distinção entre o tráfego normal e o tráfego anômalo. Uma das principais vantagens desse tipo de IDS é de não haver a necessidade de especificar todos os tipos de ataques possíveis, mas sim tipificar o que é tráfego normal através de uma etapa de treinamento. A etapa de treinamento possibilita aos algoritmos de detecção a classificação de novos pacotes, ou fluxos, a partir de características extraídas de um subconjunto do tráfego passante. Desta maneira, esse tipo de IDS é capaz de detectar ameaças do tipo *zero-day*[2] a partir da generalização dos padrões aprendidos. Uma consequência disso é a dispensa da atualização da base de definições como no IDS baseado em assinaturas. Por outro lado, uma das desvantagens do IDS baseado em anomalias em relação às demais abordagens é que normalmente apresentam uma grande quantidade de falsos positivos, e, estão sujeitos à possibilidade de manipulação no processo de criação da base de perfil de comportamento da rede, tal como apontam os estudos de Mitchel e Chen [4] e os estudos de Humayed *et al.* [5].

IDS baseados em assinaturas, também chamados de IDS baseados em conhecimento, buscam por padrões previamente classificados como maliciosos através da compa-

[2] Ameaças do tipo *zero-day* são vulnerabilidades desconhecidas no código de dispositivos e equipamentos, para a qual ainda não há correção ou assinatura que permitam o seu reconhecimento [38].

ração do tráfego passante com uma base de assinaturas. Uma das principais vantagens desse tipo de IDS é a dispensa de uma etapa de treinamento como em um IDS baseado em anomalias, desta maneira o IDS passa a detectar intrusões a partir do momento em que descarrega e aplica as últimas definições a partir do site de seu fabricante, atualizando a sua base de conhecimentos sobre ataques. A desvantagem do IDS baseado em assinaturas é que o mecanismo de detecção depende do constante fornecimento de novas definições, permitindo-lhe detectar novos tipos de ataques [4].

A geração das definições utilizadas no IDS baseado em assinaturas depende da captura e análise do tráfego de rede por empresas ou especialistas, que manualmente identificarão tráfegos maliciosos que serão utilizados para a geração da base de conhecimentos. A qualidade das assinaturas é fundamental para assegurar uma alta *acurácia* nos algoritmos de detecção de intrusões [39]. A *acurácia* neste caso representa a capacidade de correta detecção de intrusões. A falta de assinaturas atualizadas impede a identificação correta de novos ataques [4] e de ataques do tipo *zero-day*.

IDS baseados em especificação, também chamados de IDS baseados em especificação de comportamento [4], analisam o estado da rede, detectando diferenças entre o estado designado pelos projetistas e o estado operacional da rede industrial. Isso é feito através da modelagem da rede com base na definição da documentação dos estados normais de operação dos processos que atuam na rede industrial. A partir desses dados, o IDS monitora e reporta eventos que possam afetar negativamente algum dispositivo da rede [40]. A principal vantagem desse tipo de abordagem para a construção de um IDS é a capacidade de detectar ataques do tipo *zero-day* e a baixa quantidade de falsos positivos reportados. Por outro lado, a detecção baseada em especificação possui algumas limitações que prejudicam sua implantação e escalabilidade, como a dificuldade da coleta das especificações de todos os dispositivos pertencentes à rede e a necessidade dessas especificações serem definidas e mantidas por um especialista [41].

Também é possível classificar um IDS de acordo com seu posicionamento na topologia de rede, podendo dividi-lo em dois tipos: *Host-Based IDS* (HIDS) e o *Network-Based IDS* (NIDS). No HIDS, o IDS estará localizado dentro da estação, máquina ou dispositivo, capturando e analisando atividades e processos internos do equipamento, sendo executado como um programa residente. Já no NIDS, o IDS capturará e analisará o tráfego que passa por suas interfaces de rede, verificando os pacotes do tráfego de dados. Assim sendo, no caso de uma rede industrial, o NIDS deverá ser preferencialmente implantado em um ponto na rede na qual possa capturar o tráfego relativo aos protocolos industriais utilizados nos dispositivos.

Em relação ao uso de um HIDS em uma rede industrial, a principal vantagem em relação aos NIDS é monitorar o funcionamento dos processos internos do equipamento onde o HIDS está instalado. A desvantagem dessa abordagem é a possível elevada utili-

zação de processamento e memória em dispositivos que possuem justamente esse tipo de limitação, como é o caso de controladores PLC [4] [5].

Por outro lado, a vantagem dos NIDS é a não utilização de recursos, memória e processamento dos dispositivos industriais. A detecção de intrusões se dá através de análise do tráfego, observando padrões de frequência dos dados ou análise dos protocolos, sendo esse processamento realizado por um dispositivo dedicado para esse propósito. A desvantagem dessa abordagem é a necessidade de interconexão do NIDS com os demais dispositivos de maneira que o NIDS tenha visibilidade sobre todos os pontos da rede [4].

2.4 Fluxos IP

O agrupamento de dados em fluxos, como abordado por Choudhary *et al.* [42], permite sumarizar grandes quantidades de dados que trafegam em uma rede. É dado o nome de fluxo a um conjunto de um ou mais pacotes que compartilhem 5 características básicas, sendo elas: endereço IP de origem, endereço IP de destino, porta de origem, porta de destino e protocolo da camada de transporte.

A principal vantagem da utilização de fluxos é que eles proveem uma organização dos pacotes o que possibilita uma visão geral do comportamento da rede e como o tráfego é entregue entre dois pontos dentro dessa rede. A análise desses fluxos facilita a geração de uma série de estatísticas sobre a interação entre os diversos dispositivos de rede e sobre o comportamento da rede. Outra vantagem é a redução da quantidade de dados após a conversão dos pacotes, que, segundo Hofstede [43], pode chegar a 2000:1.

Uma das desvantagens da utilização do agrupamento em fluxos é referente à perda de algumas informações do pacote original, como o campo *payload*, visto que ele é descartado após a conversão para fluxo, o que impede uma análise mais profunda sobre os dados que compõe cada pacote.

Dentre os protocolos de fluxo mais utilizados temos o Netflow e o IPFIX [44]. O protocolo Netflow foi desenvolvido pela Cisco Systems em 1996, e sua versão mais recente é a versão 9 mencionada na RFC3594[45]. Enquanto que a primeira menção ao IPFIX (*IP Flow Information Export*) surgiu em 2004 através da RFC3917 [46], sendo este baseado no Netflow. Atualmente as últimas especificações do IPFIX constam na RFC7012 [47].

Ambos protocolos propõe o conceito de Janela de Atividade como sendo um delimitador para a duração de um fluxo, assim um fluxo que ultrapassar esse limiar deverá ser dividido em dois ou mais fluxos. O tempo de duração da janela de atividade é um campo que pode ser configurado manualmente pelo administrador de rede entre 120 segundos e 30 minutos. Porém, normalmente utiliza-se o valor de 30 minutos [43].

O encerramento do fluxo se dá de três maneiras: pelo recebimento de um pacote do tipo TCP-FIN ou TCP-RST; por duração ao atingir o tempo definido pela Janela de Atividade, ou, por descarte se o coletor estiver com alguma limitação de recursos [43] [45].

A exportação dos fluxos depende de dois elementos, o exportador de fluxos e o coletor de fluxos. O exportador, também chamado de *flow probe*, é um dispositivo localizado em algum ponto estratégico da rede, sendo normalmente um roteador ou um *switch*, onde poderá capturar o tráfego passante e exportá-lo, continuamente, no formato do fluxo para o coletor. O coletor de fluxos é um *software* que recebe e armazena os fluxos gerados por um ou mais exportadores [43].

2.5 Conjuntos de Dados

Conjuntos de dados (*datasets*) são ferramentas essenciais para pesquisas na área de detecção de intrusão, análise de anomalias e aprendizado de máquina. No levantamento realizado para este trabalho, foram encontrados poucos conjuntos de dados disponíveis para redes industriais que permitam uma pesquisa aprofundada em métodos de detecção de intrusões com foco em protocolos industriais, como o Modbus [13], o IEC 61850 [16] e similares. A disponibilidade de bons conjuntos de dados permite comparações entre métodos aplicados por diversos autores, comparações de resultados e reprodutibilidade de trabalhos.

Um ponto importante a ser considerado em um conjunto de dados é quanto à rotulação dos dados. O processo de rotulação normalmente é feito manualmente pelos responsáveis pela geração do conjunto de dados, assim sendo, são necessários grandes esforços para se obter um conjunto de dados rotulado e com uma quantidade relevante de informações. A dificuldade desse processo tanto limita a quantidade de conjuntos de dados disponíveis como a qualidade dos dados obtidos. Um conjunto de dados pequeno dificilmente conterá uma grande quantidade de casos possíveis de tráfego normal ou anômalo, influenciando negativamente no processo de análise [3] [48].

Dentre os poucos conjuntos de dados encontrados para este trabalho que disponibilizam dados voltados para a área de redes industriais, podemos citar os de Beaver *et al.* [49], Morris *et al.* [50], Lemay e Fernandez [21] e Goh *et al.* [51].

A principal razão que desencoraja as indústrias quanto ao compartilhamento de seus conjuntos de dados com a comunidade científica é a sensibilidade dos dados que trafegam em redes industriais [27]. Assim sendo, administradores de redes industriais tendem a acreditar que seus equipamentos estarão protegidos enquanto detalhes de suas topologias não forem divulgados, aplicando o conceito conhecido por segurança por obscuridade. Porém o uso de ferramentas como o Shodan demonstrou que essas redes podem

ser descobertas e acessíveis a qualquer atacante através da Internet [11] [52] [53].

O conjunto de dados criado em laboratório por Beaver *et al.* [49] contém medições feitas sobre uma rede de transmissão elétrica. Apresenta padrões de comunicação e comportamentos decorrentes de ciberataques. Um problema desse conjunto é que foram removidas informações importantes para a análise de intrusões em redes, como os endereços dos controladores na rede e outros campos dos pacotes de rede.

O conjunto de dados proposto por Morris *et al.* [50], criado em laboratório, apresenta dados referentes a um sistema de transporte de gás e um tanque de armazenamento de água. Os dados disponibilizados estão em arquivos com a extensão ARFF (*Attribute-Relation File Format*), em que não constam informações da arquitetura de rede utilizada.

No conjunto de dados criado por Lemay e Fernandez [21], é simulada, em laboratório, uma pequena rede industrial utilizando o protocolo de comunicação Modbus TCP/IP [26], contendo dois MTUs e alguns RTUs, na qual foram simulados ataques contra os dispositivos. Os autores disponibilizam publicamente os arquivos de captura com extensão PCAP (*Packet Capture*) juntamente com os rótulos correspondentes à classificação dos pacotes.

Por fim, no conjunto de dados disponibilizado por Goh *et al.* [51], criado em laboratório, representa, em menor escala, uma central de tratamento de águas, a qual foi submetida a uma série de ataques pelo período de 11 dias. Os dados foram disponibilizados em arquivos com a extensão CSV (*Comma Separated Values*), possuindo os rótulos correspondentes à classificação do ataque.

Tabela 1 – Sumário dos conjuntos de dados encontrados para redes industriais.

Conjunto de dados	**Ano**	**Tipo**	**Descrição**
Beaver *et al.* [49]	2013	Sintético	Transmissão elétrica
Morris *et al.* [50]	2015	Sintético	Transporte de gás e tanque de armazenamento de água
Lemay e Fernandez [21]	2016	Sintético	Rede industrial genérica
Goh *et al.* [51]	2016	Sintético	Tratamento de água

A Tabela 1 contém um sumário dos conjuntos de dados disponíveis publicamente relativos a redes industriais que foram encontrados durante as pesquisas deste trabalho. A tabela apresenta os dados ordenados em relação ao ano em que o conjunto de dados foi publicado. Pela tabela podemos observar que a quantidade de conjuntos de dados é escassa, e, além disso, não há nenhum conjunto de dados proveniente de um ambiente real, sendo todos gerados em laboratórios.

Dos conjuntos mencionados, apenas os de Lemay e Fernandez [21] e Goh *et al.* [51] disponibilizam as informações do tráfego de rede entre os dispositivos, essenciais para as

etapas de coleta de dados e geração de fluxos IP. Os demais autores se limitam a exibir apenas as comunicações correspondentes à camada de aplicação do modelo OSI. Testes iniciais realizados com o conjunto disponibilizado por Goh *et al.* [51] mostraram que não foi possível utilizar esse conjunto de dados para a proposta deste trabalho, mais detalhes serão fornecidos na seção 4.1. Logo, apenas o conjunto de dados fornecido por Lemay e Fernandez [21] será utilizado para as demais etapas.

2.6 Aprendizado de Máquina e Métodos de Classificação

Métodos de classificação são aqueles que nos permitem agrupar, rotular ou dividir em categorias dados de um conjunto apresentado [54]. Em um IDS, um dos possíveis métodos de classificação usado na busca por intrusões são os algoritmos de aprendizado de máquina. Esses algoritmos classificarão o tráfego passante entre tráfego normal (ou legítimo) e tráfego anômalo, com base em um subconjunto de dados rotulados. É classificado como tráfego anômalo aquele tráfego que se difere ou que se distancia do comportamento observado a partir do tráfego normal. Neste caso, os algoritmos de aprendizado de máquina passarão a analisar o conjunto de dados fornecido em busca de padrões de comportamento distintos ou anomalias.

Algoritmos de aprendizado de máquina buscam uma função de aproximação $f'(X)$ à função $f(X)$ que mapeie, de maneira determinística, o relacionamento entre o conjunto de dados de entrada X e o conjunto de variáveis de saída Y. Outra maneira de representar esse mapeamento é através de $Y = f(X) + \varepsilon$, onde ε representa o erro relacionado à aproximação realizada pelo algoritmo utilizado. Quando esse conjunto de variáveis de saída Y possui apenas dois valores, temos uma classificação binária [54]. No caso do uso de algoritmos de aprendizado de máquina em um IDS, os algoritmos retornarão uma classificação binária, representando tráfego normal ou tráfego anômalo.

Os algoritmos de aprendizado de máquina podem ser classificados em três tipos, sendo eles: Algoritmos Supervisionados, Algoritmos Semi-Supervisionados e Algoritmos Não Supervisionados [3]. A escolha de um algoritmo em detrimento do outro se dá pela quantidade de dados rotulados que serão fornecidos a este algoritmo.

Algoritmos Supervisionados ou algoritmos de aprendizado supervisionado são aqueles que conseguem inferir a classificação de um novo conjunto de dados com base na classificação de um subconjunto fornecido durante sua etapa de treinamento. Essa inferência normalmente gera um modelo de predição, que é um modelo estatístico utilizado para estimar resultados [54].

Nessa classe de algoritmos, a criação do modelo de aprendizado é realizada a partir de um subconjunto de dados rotulados, ao qual é dado o nome de conjunto de

dados de treinamento. Esse subconjunto deve conter uma representação significativa do tráfego a ser analisado, possuindo tráfego normal e anômalo. Dessa maneira os novos dados, também chamados de novas instâncias, não pertencentes ao conjunto de dados de treinamento serão validados contra o modelo de aprendizado gerado, para assim obter uma classificação.

Algoritmos não supervisionados são uma classe de algoritmos que agrupam um conjunto de dados não-rotulados em *clusters*. Esse agrupamento é feito de acordo com similaridades encontradas entre os dados, dispensando a criação de um modelo de aprendizado. Assim, o rótulo deixa de ser necessário previamente, sendo gerado pelas similaridades contidas nos próprios dados.

Algoritmos semi-supervisionados são uma classe de algoritmos parcialmente supervisionados, atuando entre os algoritmos supervisionados e os não supervisionados. Esses algoritmos são normalmente utilizados para classificação quando a quantidade de amostras disponíveis consegue definir bem apenas uma das classes do problema a ser modelado. Nos problemas de detecção de intrusões, é esperado que o número de pacotes relacionados a ataques seja muitas vezes menor que o número de pacotes representantes de um tráfego normal. Dessa forma, esse tipo de algoritmo utilizaria apenas dados da classe majoritária (comportamento padrão) para criar o modelo de aprendizado [55] [56]. Dentre os algoritmos semi-supervisionados podemos citar o *One-Class Support Vector Machine* (OCSVM).

As três principais famílias de classificadores de aprendizado supervisionado são: classificadores discriminativos, classificadores de árvore de decisão e classificadores estatísticos [9].

Os classificadores discriminativos são aqueles que criam o modelo de aprendizado e melhoram sua capacidade de classificação conforme adquirem mais informações sobre o conjunto de dados utilizado. Fazem parte dessa família, algoritmos como o *Support Vector Machine* (SVM), o *Locally Deep Support Vector Machine* (LD-SVM), o *Averaged Perceptron* e o *Neural Network*.

Os algoritmos *Support Vector Machine* (SVM) e o *Locally Deep Support Vector Machine* (LD-SVM) são algoritmos de aprendizado de máquina que realizam a predição da classe correta através da construção de um vetor multidimensional denominado hiperplano, no qual cada instância do conjunto de dados é mapeada. O objetivo é obter a maior distância de separação entre as classes. A diferença entre o SVM e o LD-SVM é a utilização de um *kernel* com uma escalabilidade computacional maior, melhorando o suporte a conjuntos de dados que possuem múltiplos atributos. A palavra *kernel*, neste contexto, representa a função de similaridade utilizada para o mapeamento no hiperplano [54] [57].

Os algoritmos *Averaged Perceptron* e o *Neural Network* são algoritmos de aprendizado de máquina pertencentes à família das redes neurais. O algoritmo *Averaged Perceptron* [54] é o tipo mais simples de rede neural e utiliza apenas um neurônio. Classifica linearmente os dados utilizando uma função de ativação onde é construído um vetor de pesos para cada uma das entradas desse neurônio. O termo *Averaged* se deve à utilização da média entre os vetores de pesos obtidos. O algoritmo *Neural Network* [58], base das redes neurais, classifica linearmente um conjunto de dados através da utilização de múltiplos neurônios, interconectados formando um grafo cíclico, cujos pesos atribuídos para as interconexões são ajustados conforme o processo de aprendizado da rede [59].

Os classificadores de árvore de decisão são aqueles que constroem o modelo de aprendizado com base no ganho de informação que um determinado valor de um atributo representa para o conjunto da classificação. Fazem parte dessa família, algoritmos como o *Boosted Decision Tree*, o *Decision Forest* e o *Decision Jungle*.

O algoritmo *Boosted Decision Tree*, também conhecido como *Adaboost*, é um algoritmo do tipo *ensemble*, no qual cada árvore de decisão construída tenta corrigir os erros de classificação obtidos por árvores anteriores. Algoritmos do tipo *ensemble* são aqueles que combinam o poder de classificação de diversos classificadores a fim de obterem a melhor classificação sobre um conjunto de dados fornecido. No *Boosted Decision Tree*, o final do processo de predição é feito avaliando todo o conjunto de modelos obtidos em busca da combinação que apresentou os melhores resultados. O algoritmo *Decision Forest* [60], também conhecido como *Random Forest*, é um algoritmo do tipo *ensemble*, no qual temos a construção de múltiplos conjuntos de árvores de decisão, e a classificação se dá através da votação naquele conjunto em que as predições mais se aproximam dos rótulos fornecidos pelo conjunto de treinamento [54] [59]. O algoritmo *Decision Jungle* [61] é uma extensão do *Decision Forest*, no qual, em vez de existir apenas uma rota para cada nó pertencente a cada árvore de decisão, são utilizados grafos acíclicos dirigidos (DAG) para permitir múltiplos caminhos da raiz até cada folha.

Dentro da classificação supervisionada, os classificadores estatísticos são aqueles que criam seus modelos de aprendizado baseados em funções de predição. Fazem parte dessa família algoritmos como o *Logistic Regression* e o *Bayes Point Machine*.

O algoritmo *Logistic Regression* recebe esse nome pela utilização da função logística para efetuar a classificação. A função logística é uma função que mapeia o conjunto de dados de entrada entre 0 e 1. Nesse algoritmo, a classificação se dá através da soma do mapeamento de cada uma das características do conjunto de entrada em um polinômio, o qual retornará uma classificação binária [54] [62]. O algoritmo *Bayes Point Machine*, também chamado de *Naive Bayes*, é um dos classificadores mais simples que existem, porque ele assume que não existem dependências entre as características que compõem o conjunto de entrada, assim, apenas calcula a soma das probabilidades de uso de cada uma

das características apresentadas ao conjunto de treinamento em relação a todo o conjunto de características fornecidas [54] [63].

Cada algoritmo de aprendizado de máquina de cada família mencionada utiliza uma abordagem diferente para tratar o problema do mapeamento entre dados de entrada e a classificação de saída. No contexto deste trabalho, a utilização de diferentes abordagens nos permite avaliar o comportamento desses algoritmos sobre um conjunto de dados provenientes de uma rede industrial. Neste tipo de conjunto de dados, a existência do *polling* pode representar um importante recurso na tipificação do tráfego em um IDS baseado em anomalias, como apontado no trabalho Yang *et al.* [8]. Outro aspecto que será avaliado pelos diversos tipos de algoritmos é em relação ao desbalanceamento entre tráfego normal e tráfego malicioso, um aspecto comum a todas as redes quando expostas a um ambiente não confiável, no qual teremos uma quantidade de tráfego legítimo significativamente maior que a de tráfego malicioso.

2.7 Trabalhos Relacionados

Recentemente muitos métodos foram propostos para auxiliar a detecção de intrusões em diferentes tipos de redes, dentre eles estão as redes industriais apresentadas neste trabalho.

No trabalho de Linda *et al.* [7], os autores utilizaram uma combinação de dois algoritmos de redes neurais, sendo o *Error Back-Propagation* e o *Levenberg-Macquardt* para analisar o tráfego de sistemas SCADA, utilizando dados provenientes de uma rede real. Em sua abordagem construíram uma janela de observação do tráfego passante da qual extraíram características obtidas a partir dos pacotes, como número de pacotes passantes na janela de observação, *timestamp* de início e fim, intervalo médio de pacote, quantidade de protocolos, tamanho médio dos pacotes, IP de origem, IP de destino, dentre outros campos do tráfego apresentado no conjunto de dados. A seguir utilizaram séries temporais na composição do vetor de instruções de maneira a auxiliar os algoritmos de redes neurais com a classificação dos dados. Nos testes realizados pelos autores, não foram detectados falsos positivos sob o conjunto de dados de teste, o que pode indicar algum tipo de *overfitting*. Neste contexto, o termo *overfitting* representa um modelo de aprendizado que é capaz de detectar com grande acurácia apenas os dados utilizados no conjunto de treinamento, apresentando um desempenho pobre quando submetido a novas instâncias de um conjunto de dados.

No trabalho de Barbosa *et al.* [64], os autores desenvolvem um sistema de detecção de intrusões baseado em assinatura através do uso de *white-lists* de fluxos IP. Na abordagem os autores desenvolveram um módulo que aprende o comportamento normal de

fluxos bidirecionais de rede observando as características do tráfego, sendo IP de Origem e Destino, Portas de Origem e Destino e Protocolo utilizado. Portanto, todo tráfego de um fluxo fora da *white-list* gerará um alerta ao administrador. Por outro lado, o trabalho apresentado não aborda o fato de que o uso de *white-list* pode ser contornado por uma ação que forjasse os endereços IP de origem, além disso, os autores não apresentaram métricas para validar a eficácia do modelo proposto.

No trabalho de Yang *et al.* [8], é apresentada uma abordagem híbrida entre um IDS baseado em assinaturas e um IDS baseado em especificação para o padrão de comunicação IEC 60870-5-104 [65]. A primeira parte do IDS proposto é baseado em assinatura e utiliza um conjunto de 24 assinaturas desenvolvidas pelos próprios autores para uso no Snort[3]. A segunda parte do IDS aumenta a capacidade de detecção de ataques desconhecidos através do desenvolvimento de modelos que representem o tráfego padrão da rede, reportando tráfegos que não se encaixem no modelo padrão. A criação desses modelos se dá utilizando características do tráfego em sistemas SCADA, como regularidade do tráfego e padrões previsíveis de comportamento. Apesar de essa combinação parecer promissora, os autores não apresentaram métricas que validem os resultados obtidos pela solução apresentada, tampouco disponibilizaram informações sobre o conjunto de dados utilizado ou levantaram as questões de escalabilidade relativas ao uso da parte do IDS que é baseada em especificações.

No trabalho de Hink *et al.* [67] os autores replicam o modelo de uma subestação de distribuição de energia elétrica. Utilizaram o Weka para testar 7 algoritmos de aprendizado supervisionado. Utilizaram o conjunto de dados público fornecido por Morris *et al.* [50] para avaliar o desempenho dos algoritmos. Concluíram que a combinação do algoritmo Adaboost com outro chamado JRipper apresentou a melhor acurácia dentre os algoritmos testados, mesmo quando avaliado com métricas mais complexas, como o *F-Measure*, outro nome usado para a métrica *f1 score*.

No trabalho de Pan *et al.* [37], os autores propõem um IDS baseado em especificações que utiliza redes bayesianas para modelar a interdependência dos atributos presentes na comunicação entre quaisquer dispositivos da rede. Nesse trabalho utilizaram o conjunto de dados disponibilizado por Morris *et al.* [50], no qual analisaram as variações de tempo entre as respostas dos dispositivos após o recebimento de uma solicitação de leitura ou escrita, definindo através dessas interações o estado normal da rede. A modelagem do comportamento normal criada pela rede bayesiana provê as regras que definem o IDS baseado em especificações. A abordagem dos autores é diferente das demais modelagens porque permite que o sistema se auto modele, sem a necessidade da intervenção de um especialista, principal característica do IDS baseado em especificações.

[3] Snort [66] é um NIPS (*Network Intrusion Prevention System*) baseado em assinatura e *open source*. Diferentemente de um IDS, ele é capaz de bloquear o estabelecimento de conexões conforme os dados provenientes de sua base de assinaturas, enquanto que um IDS apenas reporta a intrusão.

No trabalho de Schuster *et al.* [56], os autores utilizaram o algoritmo de aprendizado de máquina semi-supervisionado chamado *One-Class Support Vector Machine* (OCSVM) como método de detecção no desenvolvimento de um IDS baseado em anomalias para redes industriais. Os autores utilizaram um conjunto de dados proveniente de uma rede industrial real, a qual utilizava o protocolo de comunicação PROFINET IO. A seguir, agruparam o tráfego com base em um período de observação, um conceito similar à janela de observação aplicada por Linda *et al.* [7], no qual formaram vetores compostos por 4 pacotes cada, obtendo a partir deles as informações do tráfego. A seguir os autores utilizam o algoritmo OCSVM de modo a criar o modelo de tráfego normal da rede, obtendo assim altos valores nas métricas *precisão*, *sensibilidade* e *f-score*. Outro trabalho que também utiliza OCSVM é proposto por Nader *et al.* [68], no qual utilizam um *kernel* diferente para mapear os 10 atributos obtidos a partir do conjunto de dados utilizado. O *kernel* neste contexto se refere à função de mapeamento das variáveis no hiperplano. No trabalho, os autores geraram quatro tipos de ataques e comparam a abordagem sugerida com outros trabalhos que utilizam o OCSVM obtendo melhores resultados em relação a outras abordagens.

No trabalho de Junejo e Goh [9], os autores desenvolveram um IDS baseado em anomalias utilizando aprendizado supervisionado para ser utilizado no conjunto de dados denominado SWaT (*Secure Water Treatment*) [51], gerado pelo mesmo grupo de pesquisa dos autores. Nesse conjunto de dados foram injetados dez tipos de ataques. Os autores utilizaram nove algoritmos de aprendizado de máquina para classificar o tráfego malicioso. No trabalho não há menções às características do conjunto de dados que foram utilizadas nos algoritmos de aprendizado de máquina. Apesar de eles terem obtido um número pequeno de falsos positivos, elevadas *precisão* e *sensibilidade*, a pesquisa não considerou os impactos do desbalanceamento de classes majoritária e minoritária. No conjunto de dados disponível publicamente, é possível observar que a relação entre o número de instâncias de tráfego normal e tráfego anômalo era próxima a 1:1, o que não ocorre em redes que estão em produção. O mesmo conjunto de dados é utilizado no trabalho de Inoue *et al.* [69], no qual os autores utilizam aprendizado não supervisionado através da implementação do algoritmo DNN (Deep Neural Network), que é uma variação do algoritmo de redes neurais. Implementaram também o OCSVM sobre o mesmo conjunto de dados para fins de comparação. Os resultados demonstraram que o OCSVM foi superior ao DNN.

No trabalho de Ponomarev e Atkison [70] os autores propõem um IDS baseado em anomalias o qual usa dados de telemetria da rede industrial para detectar os ataques. A coleta dos dados telemétricos é feita analisando atrasos na entrega dos pacotes, tamanho de cada pacote, quantidade de pacotes perdidos, tempo de resposta, se houve retransmissão, dentre outros. Também captura o endereço MAC e o IP do dispositivo que está originando o tráfego para assim calcular quantos saltos (*hops*) existem entre origem e destino. O conjunto de dados utilizado foi gerado através da ferramenta Conpot que simula uma rede

industrial, e os ataques foram injetados através da ferramenta MetaSploit[4]. Os resultados apontaram que o algoritmo REPTree, um algoritmo do tipo *ensemble* de árvore de decisão, conseguiu uma acurácia de até 92% quando o ataque era originado na mesma rede do PLC, e que vários algoritmos testados apresentaram acurácia superior a 99% quando o ataque provinha de uma rede externa ao PLC. Os autores não utilizaram métricas mais complexas em seu trabalho, assim como não divulgaram informações sobre a topologia que foi utilizada.

O trabalho proposto por Udd *et al.* [10] apresenta um IDS baseado em assinaturas para o padrão IEC 60870-5-104. No artigo, os autores desenvolveram algumas modificações para o IDS comercial chamado Bro [71][5] de maneira que ele possa detectar 4 tipos de ataques, dos quais três são comuns a qualquer sistema SCADA e um é específico ao protocolo IEC 60870-5-104 [65]. A primeira modificação consiste em um algoritmo que automaticamente cria regras do tipo *white-list* com base no endereço ARP, endereço IP, porta e *timestamp* do pacote, criando assim uma base de conhecimento sobre o padrão normal de tráfego da rede. Em seguida desenvolveram um algoritmo que detecta variações sobre o tempo dos pacotes que trafegam na rede, detectando diferenças entre tempo mínimo, máximo, médio, quantidade de pacotes que trafegaram no período, dentre outros. Nos testes executados sobre um conjunto de dados gerado pelos próprios autores, conseguiram detectar 75% dos ataques injetados.

No trabalho de Kreimel *et al.* [72], os autores implementaram um IDS baseado em anomalias que utiliza abordagem híbrida composta por um detector de anomalias, utilizando um algoritmo semi-supervisionado baseado no k-NN (k-Nearest Neighbor), e um sistema de classificação supervisionada utilizando de redes bayesianas. Para isso reproduziram em menor escala um ambiente industrial com o protocolo Modbus [13] que lhes permitisse coletar os dados e efetuar os experimentos. Para tanto, capturaram a partir do PLC o conjunto de dados utilizado, do qual extraíram 35 características. No conjunto de dados foram injetados 4 tipos distintos de ataques. Nos resultados, os autores obtiveram uma acurácia superior a 90%, porém não foram utilizadas métricas mais complexas. O conjunto de dados utilizado não foi divulgado.

No trabalho de Siddavatam *et al.* [73] os autores verificam o uso de dois algoritmos, sendo um do tipo *ensemble*, para a identificação de anomalias em subestações de transmissão de energia elétrica, um dos muitos cenários com sistemas SCADA. Os autores montaram um pequeno ambiente composto por alguns *smart meters*, que se comunicavam através do protocolo Modbus TCP/IP [26]. A captura do tráfego se deu no *switch*, onde capturaram algumas características como ID do dispositivo, endereços MAC, IP de origem e destino, portas utilizadas, dentre outras. Os resultados apontaram que nos testes

[4] https://www.metasploit.com
[5] https://www.bro.org

efetuados com os algoritmos *Decision Tree* e o *Random Forest*, os melhores resultados foram obtidos pelo algoritmo do tipo *ensemble*, que no trabalho é o *Random Forest*. O *Random Forest* apresentou bons valores para todas as métricas utilizadas, com *acurácia* superior a 0,98, *precisão* superior a 0,98, *sensibilidade* superior a 0,99 e *acurácia média* superior a 0,99. Os testes foram realizados injetando anomalias no conjunto de dados, variando de 7% a 40%.

No trabalho de Yusheng *et al.* [74], os autores desenvolvem um IDS baseado em assinaturas para redes industriais que utilizam o protocolo de comunicação Modbus TCP/IP [26]. A abordagem proposta possui dois módulos: extrator de regras e inspeção profunda. O extrator de regras disseca o pacote Modbus TCP/IP [26] em três camadas: camada de rede, camada de transporte e camada de aplicação. A seguir o extrator de regras cria regras de comportamento normal e anormal de acordo com a correlação entre as três camadas extraídas. O módulo de inspeção profunda permanece atuando continuamente, correlacionando novos pacotes com as regras criadas de modo a reduzir a quantidade de falsos positivos. Os autores não forneceram informações sobre o conjunto de dados utilizado. Os resultados parecem promissores visto que a taxa de falsos positivos (FPR) não excedeu o valor de 0,045% em nenhum dos três ataques que foram simulados.

A Tabela 2 contém um resumo das diversas abordagens utilizadas pelos autores apresentados neste trabalho ordenados cronologicamente. Pela tabela, podemos observar que a quantidade de conjuntos de dados distintos utilizados nos trabalhos é escassa, e muitas vezes esses conjuntos de dados são gerados em laboratório e não tornados público por seus autores, o que dificulta e reduz o número de estudos disponíveis para pesquisa.

A contribuição do presente trabalho se dá no uso de classificação supervisionada para a caracterização do tráfego anômalo, no uso de métricas de avaliação adequadas para lidar com um forte desbalanceamento resultante da conversão de pacotes de rede em fluxos IP, e, na avaliação do desempenho de nove algoritmos de classificação supervisionada pertencentes às famílias dos classificadores de árvore de decisão, dos classificadores discriminativos e dos classificadores estatísticos, quando utilizados sobre um conjunto de dados proveniente de uma rede industrial.

Tabela 2 – Resumo dos trabalhos relacionados.

Autores	**Ano**	**Abordagem Utilizada**	**Conjunto de Dados**
Linda *et al.* [7]	2009	IDS Baseado em Anomalias Aprendizado de Máquinas Redes Neurais	Obtido de uma rede real. Não publicado.
Barbosa *et al.* [64]	2013	IDS Baseado em Assinatura	Combinação de um conjunto coletado de uma rede industrial real com dois conjuntos públicos não industriais.
Yang *et al.* [8]	2013	Hibrido entre IDS Baseado em Assinaturas e IDS baseado em Especificação	Obtido de uma rede real. Não publicado.
Hink *et al.* [67]	2014	IDS Baseado em Anomalias Múltiplos algoritmos de aprendizado de máquina	Utilizaram o conjunto público de Morris *et al.* [50]
Pan *et al.* [37]	2015	IDS Baseado em Especificação Redes Bayesianas	Utilizaram o conjunto público de Morris *et al.* [50]
Schuster *et al.* [56]	2015	IDS Baseado em Anomalias One Class SVM	Obtido de uma rede real. Não publicado.
Junejo e Goh[9]	2016	IDS Baseado em Anomalias Múltiplos algoritmos de aprendizado de máquina	SWaT[51]
Ponomarev e Atkison[70]	2016	IDS Baseado em Anomalias Múltiplos algoritmos de aprendizado de máquina	Gerado pelos próprios autores. Não Publicado
Nader *et al.* [68]	2016	IDS Baseado em Anomalias One Class SVM	Obtido de uma rede real. Não Publicado
Udd *et al.* [10]	2016	IDS Baseado em Assinaturas	Gerado pelos próprios autores. Não Publicado.
Inoue *et al.* [69]	2017	IDS Baseado em Anomalias	SWaT[51]
Kreimel *et al.* [72]	2017	IDS Baseado em Anomalias Abordagem híbrida entre classificação semi-supervisionada e classificação supervisionada	Gerado pelos próprios autores. Não Publicado
Sidavatam *et al.* [73]	2017	IDS Baseado em Anomalias Random Forest e Decision Tree	Gerado pelos próprios autores. Não Publicado
Yusheng *et al.* [74]	2017	IDS Baseado em Assinaturas	Não forneceram informações sobre o conjunto de dados utilizado.

3 MODELO PROPOSTO

Nesta seção, é apresentado o modelo proposto para a aplicação dos algoritmos de aprendizado de máquina através de classificação supervisionada na detecção de intrusões em sistemas SCADA. O protocolo escolhido para o trabalho foi o Modbus TCP/IP [26] o qual é amplamente utilizado em redes industriais. Porém, o modelo proposto é desenvolvido para ser compatível, sem alterações, com os protocolos de comunicação DNP3 [15] e Ethernet/IP [22], além de quaisquer outros que utilizem a pilha TCP/IP para suas comunicações. Este modelo parte da hipótese que um ataque contra uma rede industrial irá causar uma perturbação nos padrões de comunicação dessa rede. Essa perturbação pode ser considerada como um desvio no comportamento normal ou anomalia, sendo, portanto, identificável.

Para gerar o padrão de comunicação é necessário coletar informações sobre o tráfego da rede e construir um conjunto de dados rotulados de treinamento. Essas informações permitirão ao algoritmo de aprendizado de máquina a construção de um modelo de aprendizado com base nas características do tráfego normal e do tráfego anômalo, para assim notificar o administrador de redes quando necessário.

O modelo proposto é dividido em cinco partes: Coleta de Dados da Rede, Geração de Fluxos IP, Extração de Características, Treinamento e Classificação Supervisionada. A Figura 2 apresenta uma visão geral do modelo proposto. A apresentação das etapas é realizada nas próximas seções.

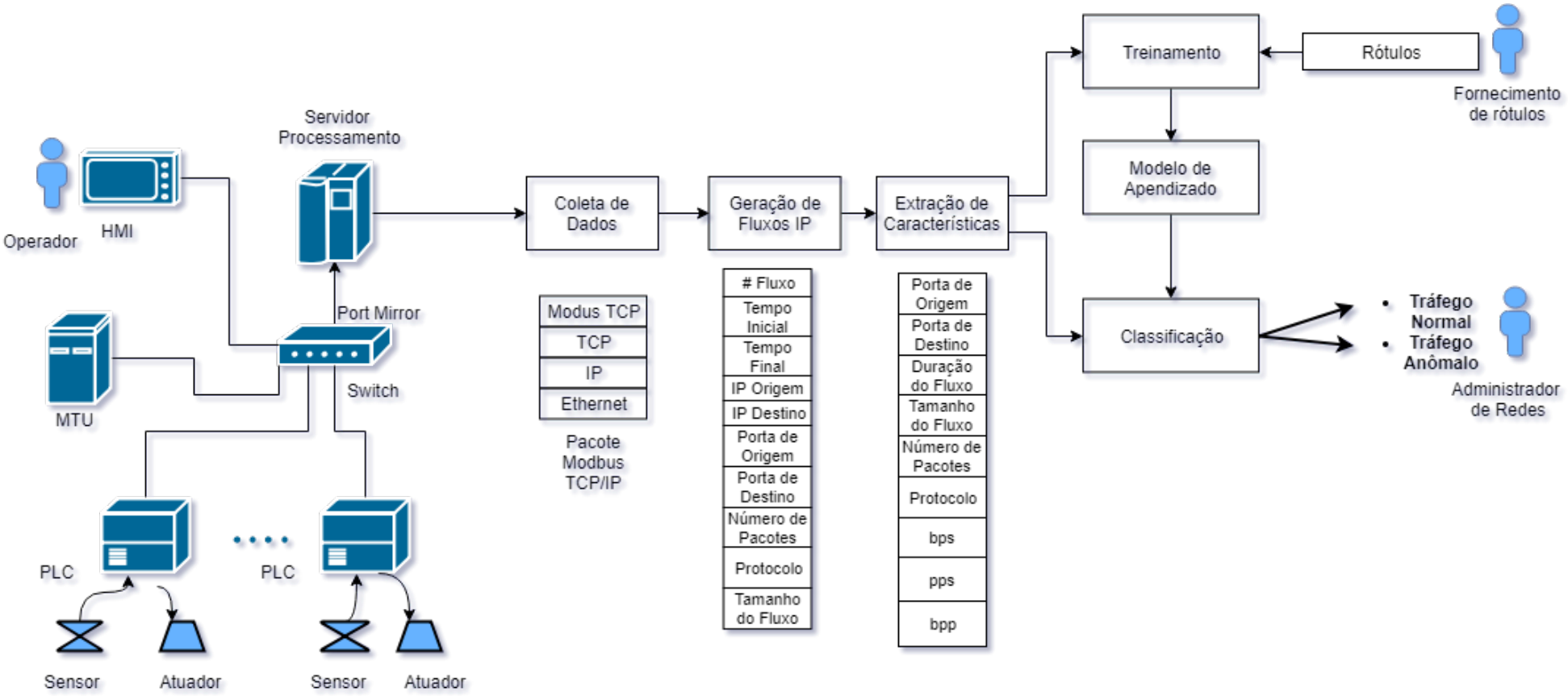

Figura 2 – Visão geral do modelo proposto.

3.1 Coleta de Dados

A primeira etapa do modelo proposto é responsável pela coleta dos pacotes TCP/IP, e é apresentada na Figura 3.

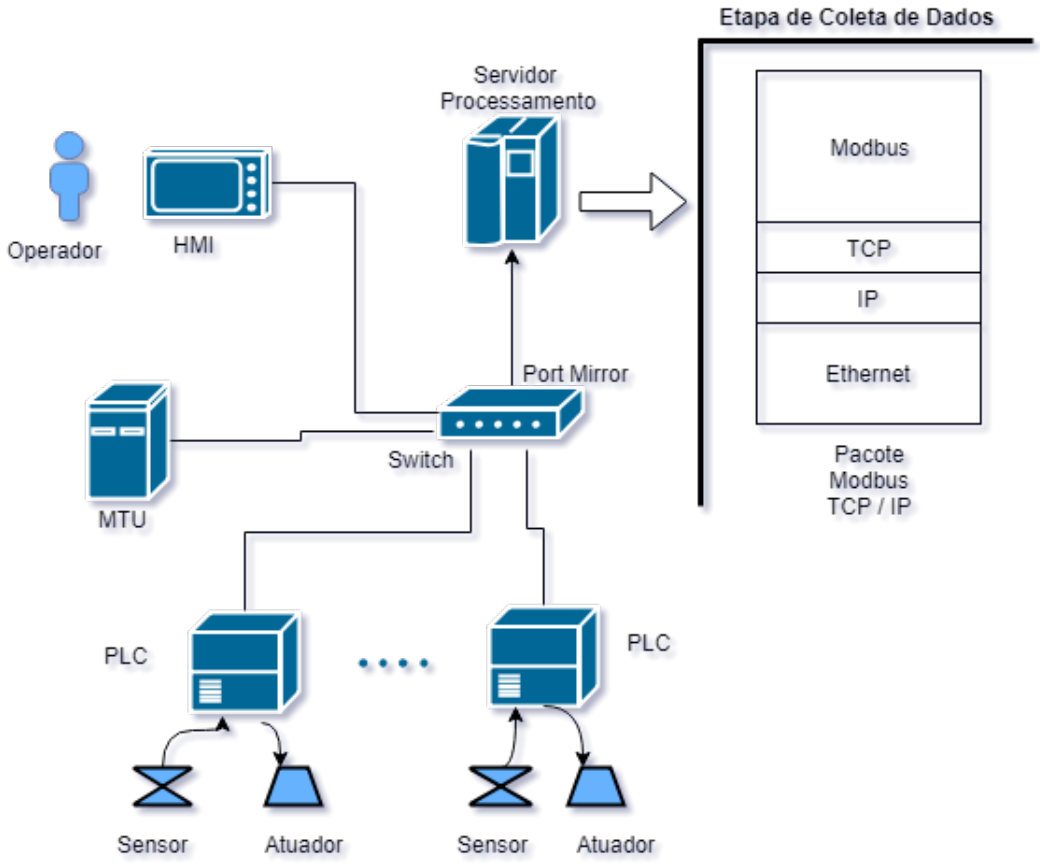

Figura 3 – Proposta de etapa de Coleta de Dados.

Nesta etapa a captura do tráfego é realizada no *switch* principal da rede industrial, aquele que comuta os pacotes de todos os PLCs envolvidos, enviando os dados capturados para o servidor de processamento para geração dos fluxos IP, conforme será apresentado na etapa 3.2.

3.2 Geração de Fluxos IP

Na segunda etapa do modelo proposto, chamada de Geração de Fluxos IP, os dados coletados na etapa anterior serão agrupados pelo coletor em fluxos IP seguindo a 5-tupla estabelecida no protocolo Netflow [44]. A 5-tupla mencionada é a representação de pacotes possuidores de um mesmo endereço IP de origem e destino, mesmas portas de origem e destino, e, que utilizem o mesmo protocolo de transporte para a criação do

fluxo IP. Outra característica utilizada na criação do fluxo IP é referente ao uso da Janela de Atividade, configurável no coletor pelo administrador de rede entre 120 segundos a 30 minutos [43].

Neste contexto, a principal vantagem proveniente do agrupamento dos pacotes em fluxos IP é que estes fornecem uma visão global do comportamento da rede, permitindo a obtenção de dados estatísticos referentes à interação entre os vários dispositivos da rede industrial. Essa interação entre dispositivos pode revelar desvios do comportamento normal causados por ataques.

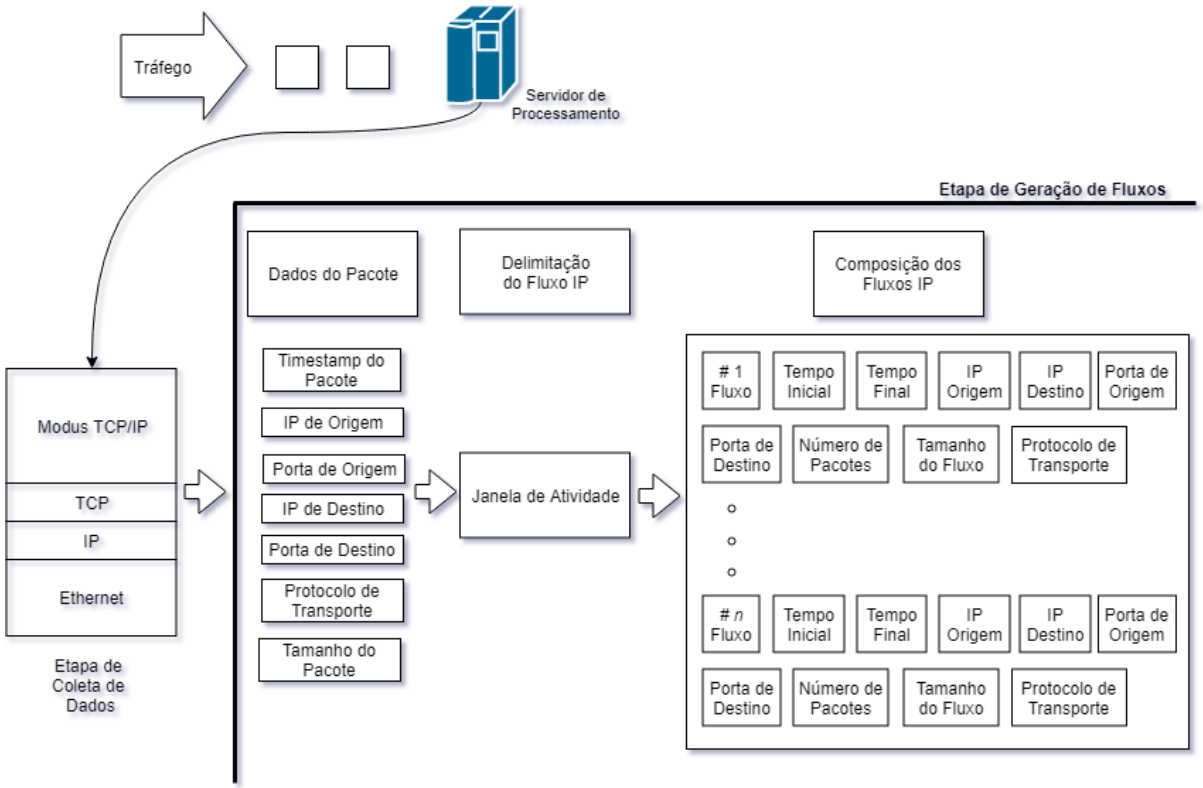

Figura 4 – Proposta de etapa de Geração de Fluxos IP.

A Figura 4 apresenta os componentes desta etapa. Dos pacotes TCP/IP coletados e armazenados no servidor de processamento pela etapa de coleta de dados, serão obtidas as informações de *timestamp* do pacote, endereço IP de origem, endereço IP de destino, porta de origem, porta de destino, tamanho do pacote e protocolo de transporte. A seguir, esses dados são enviados para as subetapas de delimitação e de composição do fluxo.

Na subetapa de delimitação do fluxo IP, temos a implementação da Janela de Atividade, na qual o *timestamp* do pacote é comparado com os tempos inicial e final do fluxo que está sendo composto, para assim verificar se este atingiu o tempo limite estabelecido na Janela de Atividade. Em caso afirmativo, um ou mais novos fluxos serão criados.

Tabela 3 – Composição do fluxo IP.

Propriedade	**Descrição**
Tempo Inicial	*Timestamp* do primeiro pacote que compõe o fluxo.
Tempo Final	*Timestamp* do último pacote que compõe o fluxo.
IP de Origem	Endereço IP de origem do tráfego.
Porta de Origem	Porta de origem do tráfego.
IP de Destino	Endereço IP de destino do tráfego.
Porta de Destino	Porta de destino do tráfego.
Protocolo de Transporte	Protocolo de transporte utilizado na conexão.
Número de Pacotes	Quantidade de pacotes agrupados no fluxo.
Tamanho do Fluxo	Soma do tamanho em bits de todos os pacotes agrupados no fluxo.

Por fim, na subetapa de composição dos fluxos, são apresentados os fluxos coletados, onde temos presentes as seguintes propriedades: Tempo Inicial e Final do fluxo gerado, IP de Origem, Porta de Origem, IP de Destino, Porta de Destino, Protocolo de Transporte, Número de Pacotes do fluxo e Tamanho do fluxo. Uma descrição de cada uma das propriedades do fluxo gerado nesta etapa é apresentada na Tabela 3.

3.3 Extração de Características

A etapa seguinte do modelo proposto é chamada de Extração de Características. Nela, dados estatísticos, aqui denominados de características, são obtidos a partir dos fluxos gerados na etapa anterior. A apresentação desta etapa é feita na Figura 5, e a descrição de cada uma das características é apresentada na Tabela 4.

As características extraídas irão compor um conjunto de dados que será utilizado como entrada para ambas as etapas de Treinamento e Classificação Supervisionada.

O lado esquerdo da Figura 5 apresenta o fluxo IP gerado na etapa de Geração de Fluxos IP sendo utilizado como entrada para a etapa de extração de característica, apresentada do lado direito da mesma figura. Nesta etapa, as características denominadas porta de origem, porta de destino, duração, tamanho do fluxo, número de pacotes, protocolo, *bps - bits per second*, *pps - packets per second*, e *bpp - bytes per packet* são extraídas a partir de informações originárias do fluxo IP. A descrição detalhada de cada uma dessas características é apresentada na Tabela 4.

Uma das principais características do tráfego em sistemas SCADA é o *polling*, que é o nome dado aos contínuos e periódicos processos automatizados de supervisão e aquisição de dados, nos quais, operações de leitura e gravação são realizadas periodicamente a partir dos MTUs em direção aos RTUs. De acordo com Yang *et al.* [8], o processo de *polling* resulta em padrões regulares de tráfego. Desta maneira, os autores afirmam que o uso

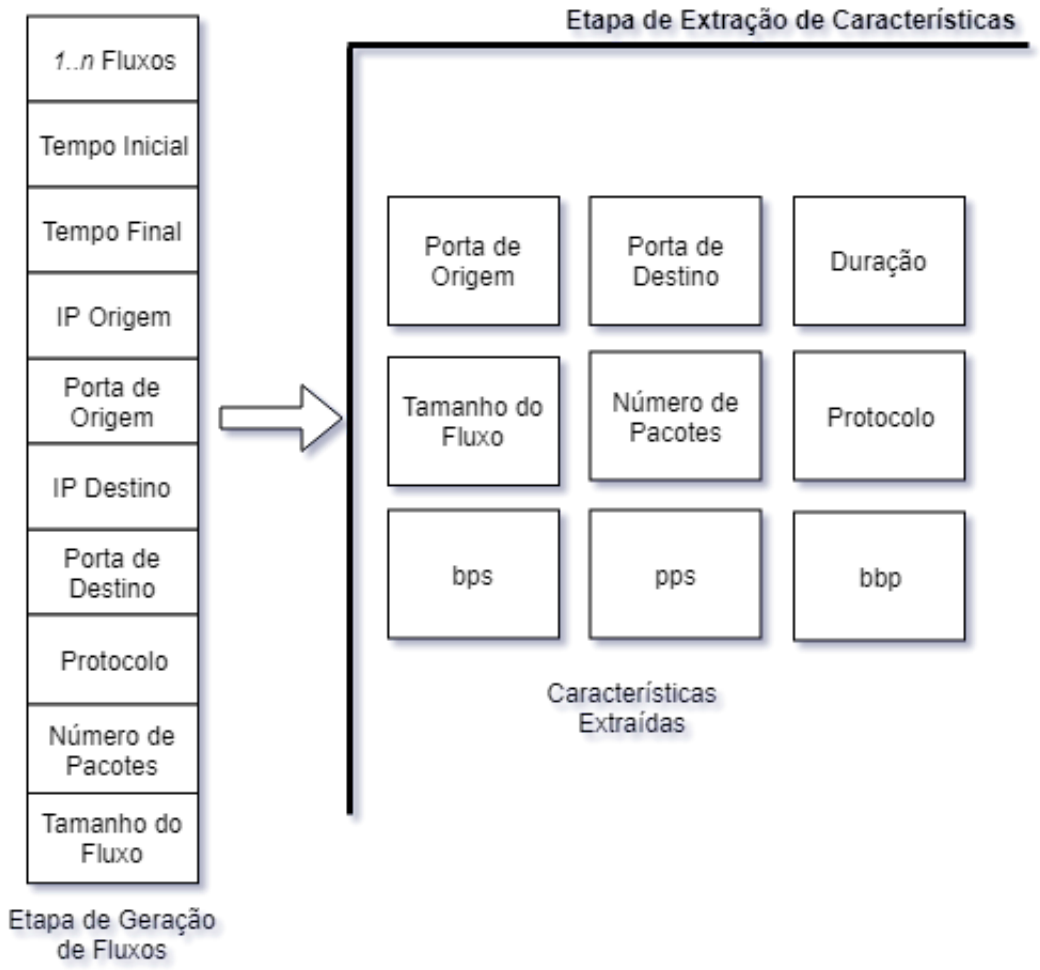

Figura 5 – Proposta de etapa de Extração de Características.

dessa característica poderá contribuir na criação de um perfil de comportamento normal da rede, o que é útil para mecanismos de detecção baseados em anomalias.

Desta maneira, a escolha das características apresentadas na Tabela 4 está diretamente relacionada à sua utilidade para os algoritmos de aprendizado de máquina, e foram escolhidas tanto com base em trabalhos que utilizam abordagens similares, com destaque para o de Linda *et al.* [7] e Siddavatam *et al.* [73], assim como, de trabalhos de outras abordagens como a de Udd *et al.* [10] e Yusheng *et al.* [74]. Características como *pps - packets per second* e *bps - bits per second* apresentam dados estatísticos sobre a velocidade com a qual a comunicação está ocorrendo, enquanto características como *Número de pacotes*, *Tamanho do Fluxo*, *bpp - bytes per packet* apresentam ao classificador informações sobre o volume de dados do fluxo entre dois dispositivos da rede.

Nesta etapa, não são utilizados os campos IP de Origem e IP de Destino provenientes da etapa de geração de fluxos IP, desta maneira, os algoritmos deverão utilizar apenas as demais características mencionadas na Tabela 4 para a criação de seus modelos de aprendizado. O objetivo desta ação é não tendenciar a classificação do modelo

Tabela 4 – Características extraídas.

Característica	**Descrição**
Porta de Origem	Porta de origem do fluxo.
Porta de Destino	Porta de destino do fluxo.
Duração	Diferença entre os *timestamps* do primeiro pacote e do último pacote que compõem o fluxo. A duração é expressa em segundos.
Tamanho do Fluxo	Soma do tamanho em bits de todos os pacotes agrupados no fluxo.
Número de Pacotes	Quantidade de pacotes agrupados no fluxo.
Protocolo	Protocolo de transporte utilizado no fluxo.
bps - bits por segundo	Divisão entre o tamanho do fluxo (em bits) e a duração do fluxo (em segundos). ou seja, $\frac{Tamanho\ do\ Fluxo}{Duração}$. O *bps* é expresso em bits por segundo.
pps - pacotes por segundo	Divisão entre o número de pacotes do fluxo e a sua duração (em segundos), ou seja, $\frac{Número\ de\ Pacotes}{Duração}$. O *pps* é expresso em pacotes por segundo.
bpp - bits por pacote	Divisão entre o tamanho do fluxo (em bits) e seu número de pacotes. Representa o tamanho médio de um pacote no fluxo, ou seja, $\frac{Tamanho\ do\ Fluxo}{Número\ de\ Pacotes}$. O *bpp* é expresso em bits.

de aprendizado ao tráfego proveniente de um dispositivo ou segmento de rede que esteja sendo analisado.

3.4 Treinamento

Na etapa de Treinamento, um subconjunto dos dados gerados pela Etapa de Extração de Características, que a partir deste ponto passa a ser chamado de conjunto de dados de treinamento, juntamente com um conjunto de rótulos fornecidos por um administrador da rede, permitirão ao algoritmo de aprendizado de máquina a criação de seu modelo de aprendizado.

Na Figura 6, é possível visualizar os componentes desta etapa. Nela, o conjunto de dados de treinamento juntamente com os rótulos fornecidos pelo administrador são utilizados como entrada para o algoritmo de aprendizado de máquina e este gerará o modelo de aprendizado.

É importante salientar que a qualidade do modelo de aprendizado gerado depende tanto do conjunto de dados de treinamento, que devem conter uma quantidade significativa de amostras, quanto do conjunto de rótulos fornecidos pelo administrador de redes. Portanto uma pequena quantidade de amostras rotuladas incorretamente influenciará ne-

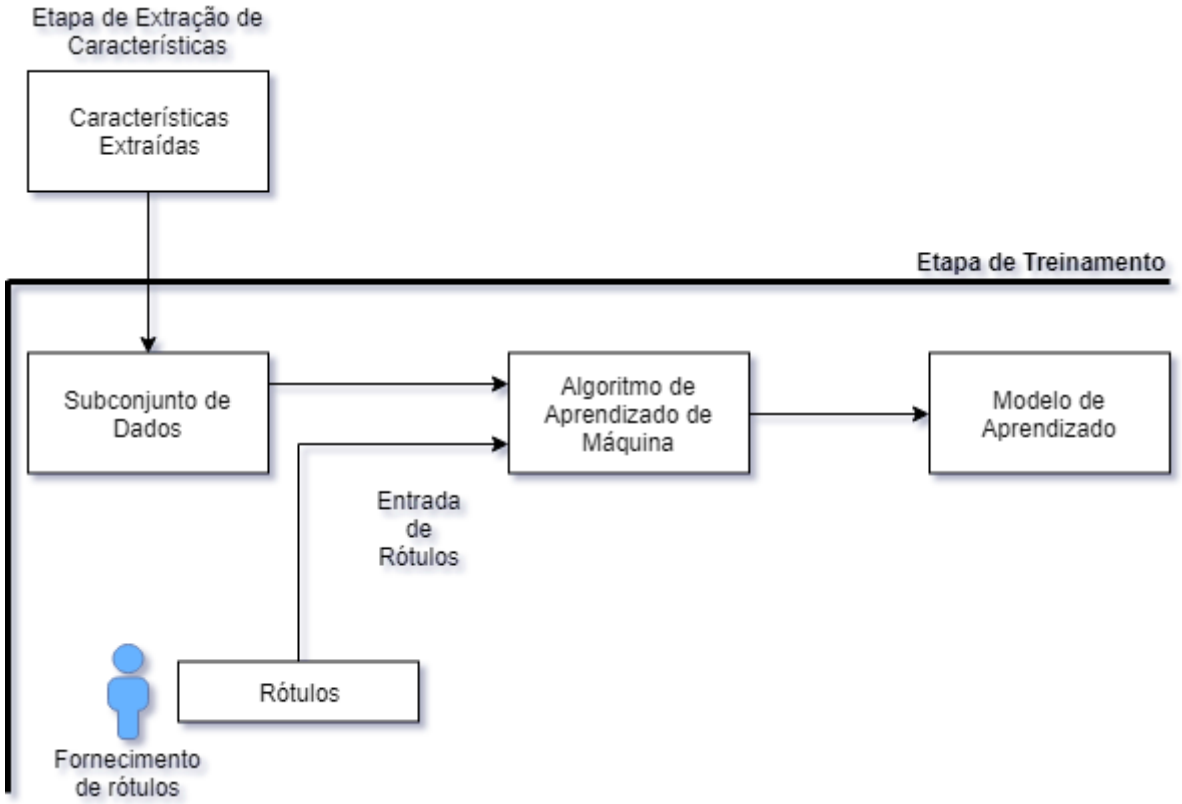

Figura 6 – Proposta de etapa de Treinamento dos Algoritmos de Aprendizado de Máquina.

gativamente na construção do modelo de aprendizado [3]. A qualidade do modelo de aprendizado é avaliada com o uso de métricas, que serão apresentadas na seção 4.3.

3.5 Classificação Supervisionada

Na última etapa, chamada de Classificação Supervisionada, novas instâncias serão apresentadas ao classificador, que, com base no modelo de aprendizado gerado na Etapa de Treinamento, realizará a classificação entre tráfego normal e tráfego anômalo, retornando um valor de probabilidade de acerto ao administrador.

Nesta etapa o ajuste da sensibilidade do classificador se dá através de um limiar de decisão chamado de *threshold* [48]. Através do *threshold* o administrador de rede poderá associar a probabilidade retornada pelo classificador com o conjunto de resultados esperados, de maneira a que valores acima de um determinado algarismo serão considerados Tráfego Normal, enquanto que valores abaixo serão considerados Tráfego Anômalo. Na Figura 7, é possível visualizar todos os componentes dessa etapa.

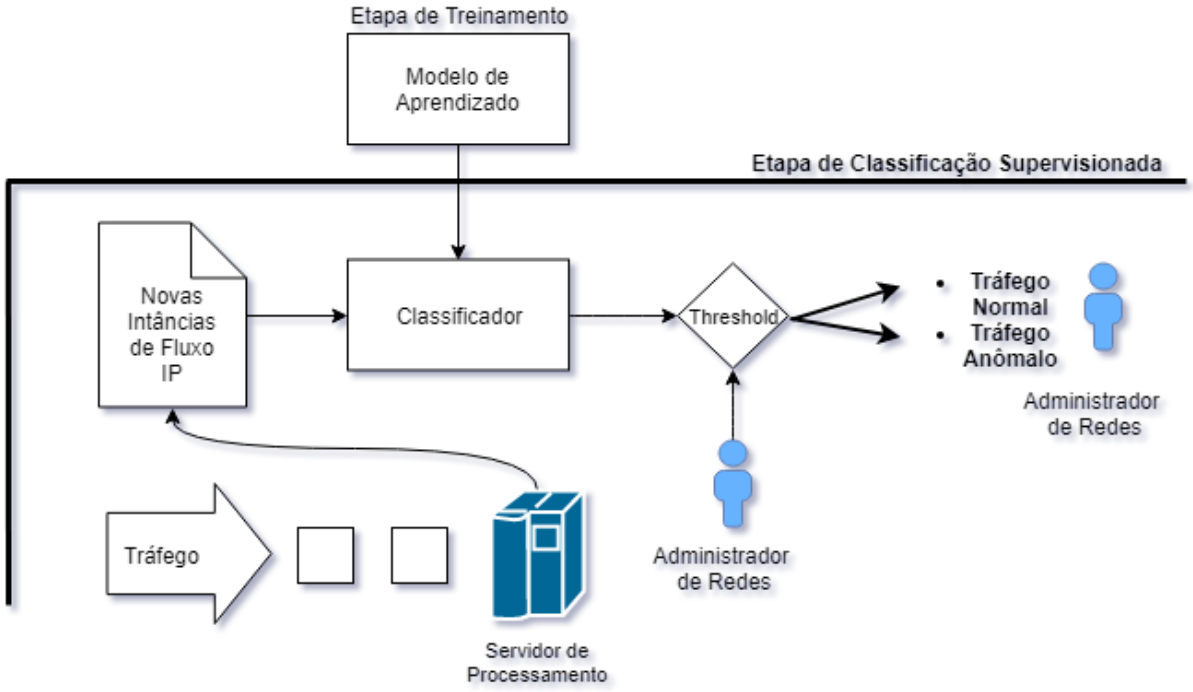

Figura 7 – Proposta de etapa de Classificação Supervisionada.

4 RESULTADOS

Neste capítulo são apresentados os resultados da avaliação dos algoritmos de aprendizado supervisionado utilizando o modelo proposto no capítulo 3. A avaliação foi realizada com um conjunto de dados disponibilizado publicamente. A seguir serão apresentados o conjunto de dados na seção 4.1, o ambiente de testes e a metodologia utilizada na seção 4.2, as métricas de avaliação e resultados obtidos na seção 4.3, e, por fim, a discussão sobre os resultados na seção 4.4.

4.1 Conjuntos de Dados

Há uma certa dificuldade em localizar conjuntos de dados que possam ser utilizados para pesquisas na área de detecção de intrusões em redes industriais. Os motivos para isso estão relacionados a preocupações decorrentes da sensibilidade dos dados fornecidos quando coletados de redes industriais pertencentes a órgãos governamentais e indústrias privadas. Os poucos conjuntos de dados encontrados para a área foram gerados em ambientes controlados, sendo reproduções em escala de ambientes reais, contendo limitações na quantidade ou na variedade de dispositivos e padrões de comunicação.

Durante o desenvolvimento deste trabalho, alguns problemas impediram o uso de alguns dos conjuntos de dados encontrados. No caso de Beaver *et al.* [49] e Morris *et al.* [50], os conjuntos de dados disponibilizados por ambos os autores estavam em formatos de arquivo nos quais foram omitidas informações necessárias para a composição dos fluxos IP, como o IP dos dispositivos de rede, protocolos utilizado, entre outras. No caso do conjunto de dados disponibilizado por Goh *et al.* [51], foram encontradas inconsistências no conjunto de dados que invalidaram os testes realizados. Os problemas encontrados nesse conjunto de dados se apresentaram quando utilizados na etapa de Classificação Supervisionada. Investigando-se a questão, verificou-se que ao coletar duas instâncias quaisquer das classes majoritárias e minoritárias, não havia atributos suficientes que permitissem ao classificador predizer a qual classe aquela instância pertencia. De acordo com Hastie *et al.* [54], a dificuldade encontrada é uma das apresentações do problema denominado *curse of dimensionality*[1].

Por outro lado, o conjunto de dados disponibilizado por Lemay e Fernandez [21] não apresentou nenhuma das limitações mencionadas. Sendo assim, este foi o conjunto de dados utilizado como entrada para a etapa de Coleta de Dados. O conjunto de dados disponibilizado por Lemay e Fernandez [21] é composto principalmente por transações

[1] Neste caso o modelo de aprendizado gerado é ruim porque havia uma baixa dimensionalidade do conjunto de dados, ou seja, o número de características recebidas eram insuficiente para representar corretamente o conjunto de dados, resultando em um modelo com baixa precisão.

automatizadas entre dispositivos mestres (MTU) e escravos (RTU). Esse tipo de comunicação normalmente possui uma periodicidade no tráfego entre os dois pontos que se comunicam. Desta forma, pequenas discrepâncias nessa periodicidade podem representar tanto uma interação de um humano com o sistema através do HMI, quanto um tráfego malicioso representando um ataque. No conjunto de dados são providas mais de seis horas de tráfego, que incluem ataques à rede, como a execução de *exploits* remotos, *upload* de arquivos para a rede infectada, ataques do tipo *fingerprint* e o envio de comandos não autorizados a um controlador.

Os arquivos *pcap* apresentados na Tabela 5 foram utilizados como entrada para a etapa de Coleta de Dados. Na Tabela 5, é apresentado o nome do arquivo, uma breve descrição de seu conteúdo, data de início e fim da captura, o número de pacotes representando tráfego normal, o número de pacotes representando tráfego malicioso e uma coluna representando a soma de pacotes normais com pacotes maliciosos. É importante salientar que na alimentação da etapa de Coleta de Dados os arquivos mencionados foram importados em ordem cronológica conforme a data de início e data de fim.

A implementação da etapa de Coleta de Dados se deu juntamente com as etapas de Geração de Fluxos IP e de Extração de Características. Para tanto, foi utilizada a suíte de aplicativos NFDUMP [75]. A suíte NFDUMP é desenvolvida desde 2005 por Peter Haag e consiste em uma série de ferramentas capazes de capturar e processar fluxos IP no padrão Netflow v9.

Assim sendo, foi preparado um servidor utilizando o *software* Oracle VirtualBox como plataforma de virtualização. Esse servidor físico executou o Linux CentOS versão 7 na arquitetura 64 bits como sistema operacional virtualizado, no qual dois processos estiveram em execução, o *nfpcapd* e o *nfdump*. A coleta de fluxos normalmente é realizada por roteadores compatíveis com o protocolo Netflow, porém neste ambiente optamos por utilizarmos o *software nfpcapd* para a geração dos fluxos. Assim o *nfpcapd* recebeu como entrada os arquivos *pcap* da Tabela 5, e, configuramos um parâmetro correspondente a uma Janela de Atividade de trinta minutos, gerando como resultado os arquivos correspondentes aos fluxos IP. Esses arquivos foram utilizados como entrada para o *software nfdump*, responsável por extrair as características apresentadas na Tabela 4. O resultado da extração de características é uma coleção de arquivos CSV, que representam o tráfego agrupado em fluxos e com as características extraídas, os quais passaremos a referenciar com o termo de conjunto de dados com características.

Em seguida, sobre o conjunto de dados com características foi executada a associação com os rótulos fornecidos pelos autores do conjunto de dados original. Os rótulos consistem em marcações que identificam dos pacotes originais quais continham tráfego malicioso e quais continham tráfego legítimo.

Tabela 5 – Sumário do conjunto de dados utilizado.

Arquivo	**Descrição**	**Data Inicio**	**Data Fim**	**Pacotes Normais**	**Pacotes Maliciosos**	**Total**
moving_two_files _modbus_6RTU.pcap	Tráfego entre 1 MTU e 6 RTU. Ataque: Upload de um arquivo para o MTU	24/02/15 13:48:50	24/02/15 13:52:00	3244	75	3319
exploit_ms08_net api_modbus_6RTU _with_operate.pcap	Tráfego entre 1 MTU e 6 RTU Ataque: Uso de exploit para comprometer o RTU	24/02/15 14:09:57	24/02/15 14:10:30	657	1199	1856
CnC_uploading_e xe_modbus_6RTU _with_operate.pcap	Tráfego entre 1 MTU e 6 RTU Ataque: Upload de um executável entre dois RTUs.	24/02/15 14:18:20	24/02/15 14:19:30	1305	121	1426
characterization_mod bus_6RTU_with_opera te.pcap	Tráfego entre 1 MTU e 6 RTU Ataque: Sem ataques	24/02/15 14:27:45	24/02/15 14:33:24	12296	0	12296
send_a_fake_command _modbus_6RTU _with_operate.pcap	Tráfego entre 1 MTU e 6 RTU. Ataque: Envio de comandos falsos.	24/02/15 14:35:20	24/02/15 14:46:30	11156	10	11166
Total de Pacotes				28658	1405	30063

A associação dos rótulos com o conjunto de dados com características foi implementada em linguagem PHP (*PHP: Hypertext Preprocessor*)[2,3] [76] e sua execução se deu utilizando o PHP CLI[4] [77] e, durante a associação, consideramos que se um dos pacotes que compõe o fluxo fosse malicioso, todo o fluxo seria considerado malicioso. Os resultados obtidos foram exportados em formato CSV para utilização nas etapas de Treinamento e Classificação Supervisionada.

Um sumário dos resultados da conversão dos pacotes provenientes da etapa de Coleta de Dados no conjunto de dados com características após o processo de associação com os rótulos é apresentado na Tabela 6.

Tabela 6 – Conjunto de dados e conversão em fluxos IP.

	Número de Pacotes do Conjunto de Dados	**Número de Fluxos após conversão**
Classe Majoritária (Tráfego Normal)	28658	4867
Classe Minoritária (Tráfego Malicioso)	1405	27
Relação entre classes (Tráfego Normal vs. Tráfego Malicioso)	20:1	180:1
Total	**30063**	**4896**

Observando os resultados apresentados na Tabela 6 podemos notar que os 30063 pacotes do conjunto de dados resultaram em 4896 fluxos IP após o processo de conversão, o que alterou a relação entre classes de 20:1 para 180:1.

4.2 Ambiente de Testes e Metodologia

Nas etapas de Treinamento e Classificação Supervisionada, o conjunto de dados com características e rotulado, gerado na seção 4.1, foi utilizado para alimentar nove algoritmos de aprendizado de máquina. A escolha dos algoritmos foi motivada por três fatores: o primeiro relacionado a abordar representantes das principais famílias de classificadores;

[2] O PHP é uma linguagem de *script*, de código aberto (*open source*), com sintaxe similar ao C++, é amplamente utilizada para desenvolvimento *Web*, mas não limitada apenas para esse fim, sendo utilizada em diversos tipos de aplicações. Possui uma série de vantagens perante outras linguagens, como ser multiplataforma, com capacidades de interação tanto com o sistema operacional através de aplicações locais quanto através de aplicações *Web*.

[3] http://php.net

[4] O PHP CLI (*Command Line Interface*) permite ao PHP ser executado como se fosse um aplicativo independente do sistema operacional, recebendo parâmetros e exibindo respostas na tela do usuário.

o segundo para verificar o funcionamento de cada algoritmo com conjuntos de dados provenientes de redes industriais; e o terceiro para verificar o funcionamento dos algoritmos quando o conjunto de dados de entrada possui um desbalanceamento entre as classes.

A ferramenta utilizada para a implementação das etapas de Treinamento e de Classificação Supervisionada foi o Microsoft Azure Machine Learning Studio[5]. O Microsoft Azure Machine Learning Studio [78] é uma ferramenta de análise preditiva baseada em computação em nuvem, na qual é possível a execução, em paralelo, de múltiplos algoritmos de aprendizado de máquina, aproveitando-se de uma estrutura escalável e elástica [79]. A ferramenta mencionada é compatível com as linguagens de programação R e Python, o que permite que programas e algoritmos desenvolvidos nessas linguagens sejam executados na nuvem e os resultados do processamento sejam acessíveis em locais geograficamente distantes.

Durante o processo de detecção de tráfego malicioso, é esperado que haja um desbalanceamento entre a classe majoritária, representante do tráfego normal, e a classe minoritária, representante do tráfego anômalo ou ataques, já que este é o comportamento normal de uma rede de dados, onde temos uma quantidade maior de tráfego normal do que anômalo. Esta é uma das características não seguidas pelo conjunto de dados de Goh *et al.* [51] mencionado anteriormente, visto que, no conjunto de dados disponibilizado pelos autores, a relação entre as duas classes contabilizada a partir do número de pacotes disponibilizado era próxima a 1:1. O desbalanceamento, esperado no tráfego de qualquer rede, pôde ser observado no conjunto de dados fornecido por Lemay e Fernandez [21], onde a relação entre as duas classes contabilizada pelo número de pacotes era próxima a 20:1.

Ao observar o conjunto de dados, constatamos que, na etapa de treinamento, os dados apresentados aos algoritmos continham um desbalanceamento total de 180:1. Durante o treinamento do modelo de aprendizado esse desbalanceamento era de 182:1, enquanto que correspondeu a 162:1 durante os testes do mesmo modelo. Estes desbalanceamentos correspondem a cada uma das iterações ao qual os algoritmos foram submetidos durante a construção do modelo de aprendizado, e verificada através da metodologia da validação cruzada, explicada em mais detalhes na seção 4.3. A Tabela 7 apresenta o conjunto de dados completo, assim como as relações entre classes nos conjuntos de treinamento e testes.

[5] https://studio.azureml.net

Tabela 7 – Nível de desbalanceamento e tamanho das classes.

	Conjunto de Treinamento *(9 partições)*	**Conjunto de Testes *(1 partição)***	**Total do Conjunto de Dados**
Fluxos IP Legítimos	4379	488	**4867**
Fluxos IP Maliciosos	24	3	**27**
Relação	182:1	162:1	**180:1**

De acordo com He e Ma [48], alguns algoritmos de aprendizado de máquina podem apresentar melhores resultados quando a relação entre as classes é próxima a 1:1. Como não é este o caso e, de acordo com os dados apresentados na Tabela 7, o forte desbalanceamento encontrado poderá influenciar negativamente na performance de alguns dos algoritmos de aprendizado de máquina mais simples, portanto esperamos ver essa característica nos resultados finais.

4.3 Métricas de Avaliação e Resultados

As métricas de avaliação permitem analisar a performance de algoritmos de aprendizado de máquina. Uma das formas mais simples de avaliação é através da matriz de confusão (*confusion matrix*), que é uma tabela na qual os dados de erros e acertos entre as classes são contabilizados como apresentado na Tabela 8.

Tabela 8 – Matriz de confusão.

	Valor Verdadeiro: Positivo	Valor Verdadeiro: Negativo
Valor Previsto: Positivo	**TP Verdadeiro Positivo**	**FP Falso Positivo**
Valor Previsto: Negativo	**FN Falso Negativo**	**TN Verdadeiro Negativo**

Na matriz de confusão, como apresentada na Tabela 8, um falso negativo (FN - *False Negative*) ocorre quando o classificador reporta incorretamente um tráfego malicioso como sendo tráfego legítimo. Um verdadeiro positivo (TP - *True Positive*) ocorre quando o classificador reporta corretamente um tráfego normal como assim sendo. Um falso positivo (FP - *False Positive*) ocorre quando o classificador reporta incorretamente um tráfego legítimo como sendo um tráfego malicioso. Por fim, um verdadeiro negativo (TN - *True*

Negative) ocorre quando o classificador corretamente reporta um tráfego malicioso como assim sendo [4]. Outras métricas mais complexas são derivadas a partir dos resultados apresentados na matriz de confusão e são apresentadas a seguir:

- $precisão = \frac{TP}{TP+FP}$
  A *precisão* (*precision*) representa a razão entre o número de predições corretas (TP) e a soma das predições corretas (TP) com o número de predições incorretas (FP). É apresentada como um número contínuo compreendido entre 0 e 1, sendo 1 o melhor valor e 0 o pior valor.

- $sensibilidade = \frac{TP}{TP+FN}$
  A *sensibilidade* (*recall ou sensitivity*), também chamada de TPR (True Positive Rate), representa a razão entre as predições corretas (TP) e a soma de todas as predições corretas possíveis (TP + FN). É apresentada como um número contínuo compreendido entre 0 e 1, sendo 1 o melhor valor e 0 o pior valor.

- $FPR = \frac{FP}{FP+TN} = 1 - especificidade$
  A *FPR* (*False Positive Rate*), ou Taxa de Falsos Positivos, representa a razão entre as predições incorretas (FP) em relação à soma das predições incorretas com as corretas da classe majoritária (FP + TN). Demonstra a quantidade de falsos positivos que o algoritmo está retornando. É apresentada como um número contínuo compreendido entre 0 e 1, sendo 0 o melhor valor e 1 o pior valor.

- $especifidade = \frac{TN}{TN+FP}$
  A *especificidade* (*specificity*), também chamada de *TNR* (*True Negative Rate*), demonstra o número de predições corretas na classe majoritária (TN) em relação ao total de predições possíveis da classe majoritária (TN + FP). É apresentada como um número contínuo compreendido entre 0 e 1, sendo 1 o melhor valor e 0 o pior valor.

- $acurácia = \frac{TP+TN}{TP+FN+FP+TN}$
  A acurácia mede a razão entre o número de casos corretamente previstos em relação a todos os casos apresentados. É apresentada como um número contínuo compreendido entre 0 e 1, sendo 1 o melhor valor e 0 o pior valor.

- $acurácia\_média = \frac{sensibilidade+especificidade}{2} = \frac{TP}{2TP+2FN} + \frac{TN}{2TN+2FP}$
  A *acurácia média* representa a média simples entre a *sensibilidade* e a *especificidade*. A métrica da *acurácia média* é apresentada como um número contínuo compreendido entre 0 e 1, sendo 1 o melhor valor e 0 o pior valor.

- $f1\ score = 2\frac{precisão*sensibilidade}{precisão+sensibilidade} = \frac{2TP}{2TP+FP+FN}$
  Também chamada de *F-Mesure*, esta métrica mensura a média harmônica entre a

precisão e a *sensibilidade*. A métrica do *f1 score* é apresentado como um número contínuo compreendido entre 0 e 1, sendo 1 o melhor valor e 0 o pior valor.

Em relação ao uso de métricas em conjuntos de dados que apresentem um desbalanceamento entre classes superior a 10:1, os autores He e Ma [48] recomendam o uso das métricas *f1 score* e a *acurácia média* no lugar da *acurácia* para a avaliação do desempenho do classificadores. Apesar da *acurácia* ser uma das principais métricas utilizadas, ela não é uma métrica eficiente quando há um desbalanceamento entre as classes estudadas. De acordo com os autores, o principal problema encontrado por classificadores, quando há um desbalanceamento forte entre classes, é que o erro de detecção por classe aumenta significativamente na classe minoritária, o que pode prejudicar a capacidade de detecção dos algoritmos avaliados. Mesmo assim, neste trabalho, a *acurácia* também foi calculada e é utilizada para fins de comparação com as demais métricas.

Os parâmetros utilizados durante o treinamento de cada um dos algoritmos testados são apresentados na Tabela 11 localizada no capítulo de Apêndice A deste trabalho. Cada algoritmo possui uma quantidade grande de parâmetros possíveis, assim sendo, neste trabalho experimentamos vários valores a fim de obtermos os maiores resultados na métrica *f1 score*.

A avaliação dos resultados dos algoritmos se deu utilizando o método da validação cruzada. A validação cruzada (*cross-validation*) é uma técnica que consiste em particionar o conjunto de dados de entrada em n partes denominadas partições. Neste trabalho utilizamos $n = 10$, ou seja, o conjunto de dados de entrada foi dividido em dez partes iguais e foram realizadas dez iterações com ele. Além disso, foi utilizada a parametrização *Stratified Split* na divisão das partições, a qual garante que a proporção de desbalanceamento entre as duas classes permanecerá inalterada em cada uma das partes. As métricas apresentadas são o resultado médio da avaliação de $n - 1$ partes utilizadas para o treinamento do modelo computacional e a parte restante utilizada para testes, repetido por n vezes, assim, garantindo que cada uma das partes seja utilizada pelo menos uma vez para os testes [80].

Em números, isso significa que a cada iteração da validação cruzada 4379 fluxos IP legítimos e 24 fluxos IP maliciosos foram utilizados para treinar o algoritmo, e logo a seguir o modelo de aprendizado é testado contra 488 fluxos IP legítimos e outros 3 fluxos IP maliciosos, sendo este processo repetido por 10 vezes.

Analisando os resultados apresentados na Tabela 9 e considerando como critério de avaliação primeiramente os valores obtidos pela métrica *f1 score*, seguida da métrica *acurácia média* e, por fim, da métrica *acurácia*, podemos afirmar os melhores algoritmos foram, em ordem , o *Boosted Decision Tree*, o *Decision Forest*, o *Decision Jungle*, o *Bayes Point Machine*, o *Neural Network*, o *Averaged Perceptron*, o *Locally Deep Support Vector*

Tabela 9 – Resultados da avaliação dos algoritmos.

Algoritmo	***f1 score***	***acurácia média***	***acurácia***
Boosted Decision Tree	0,86667	0,99845	0,99836
Decision Forest	0,84745	0,99804	0,99816
Decision Jungle	0,80701	0,96142	0,99775
Bayes Point Machine	0,76667	0,92489	0,99713
Neural Network	0,75862	0,96121	0,99713
Averaged Perceptron	0,73333	0,94816	0,99673
Locally Deep Support Vector Machine	0,61224	0,89297	0,99611
Support Vector Machine	0,56250	0,94272	0,99550
Logistic Regression	0,44067	0,73878	0,99468

Machine, o *Support Vector Machine* e, por fim, o *Logistic Regression*.

Em cenários no qual há um forte desbalanceamento entre classes, os autores He e Ma [48] recomendam a combinação do uso das métricas citadas anteriormente sendo *f1 score* e *acurácia média* juntamente com o uso de outros avaliadores, como a Curva ROC (*Receiver Operating Characteristic Curve - ROC Curve*) e a Curva PR (*Precision-Recall Curve - PR Curve*), pois elas podem prover uma visão mais completa sobre a análise dos resultados obtidos.

A Curva ROC provê um método visual para verificar o desempenho de um classificador [48]. Nela é possível visualizar o *trade-off* de valores entre a métrica *sensibilidade* e a métrica *taxa de falsos positivos* (FPR) para cada um dos algoritmos utilizados. Em uma curva ROC, o valor ideal estará localizado no quadrante superior esquerdo, o que, no conjunto de resultados observado, implicaria em uma sensibilidade próxima a 1 e uma taxa de falsos positivos (FPR) próxima a 0.

A curva PR, assim como a curva ROC mencionada anteriormente, provê um método visual para avaliar o desempenho de um algoritmo. Nela é possível visualizar o *trade-off* de valores entre a métrica *precisão* e a métrica *sensibilidade* para cada um dos algoritmos testados. Em uma curva PR, o valor ideal estará localizado no quadrante superior direito, o que, no conjunto de dados observado implica em uma *precisão* e uma *sensibilidade* próximas ao valor 1. Portanto, quando um determinado algoritmo apresentar uma alta *sensibilidade* com uma baixa *precisão*, ocupando o quadrante inferior direito, reportará um elevado números de falsos positivos em sua matriz de confusão como apresentada na Tabela 8. Enquanto que um algoritmo que apresentar uma alta *precisão*, porém com uma baixa *sensibilidade*, ocupando o quadrante inferior esquerdo da curva PR, observaremos um elevado número de falsos negativos em sua matriz de confusão, não acusando a maioria dos ataques ao administrador.

Considerando que a curva ROC é composta pelas métricas *sensibilidade* e *taxa de*

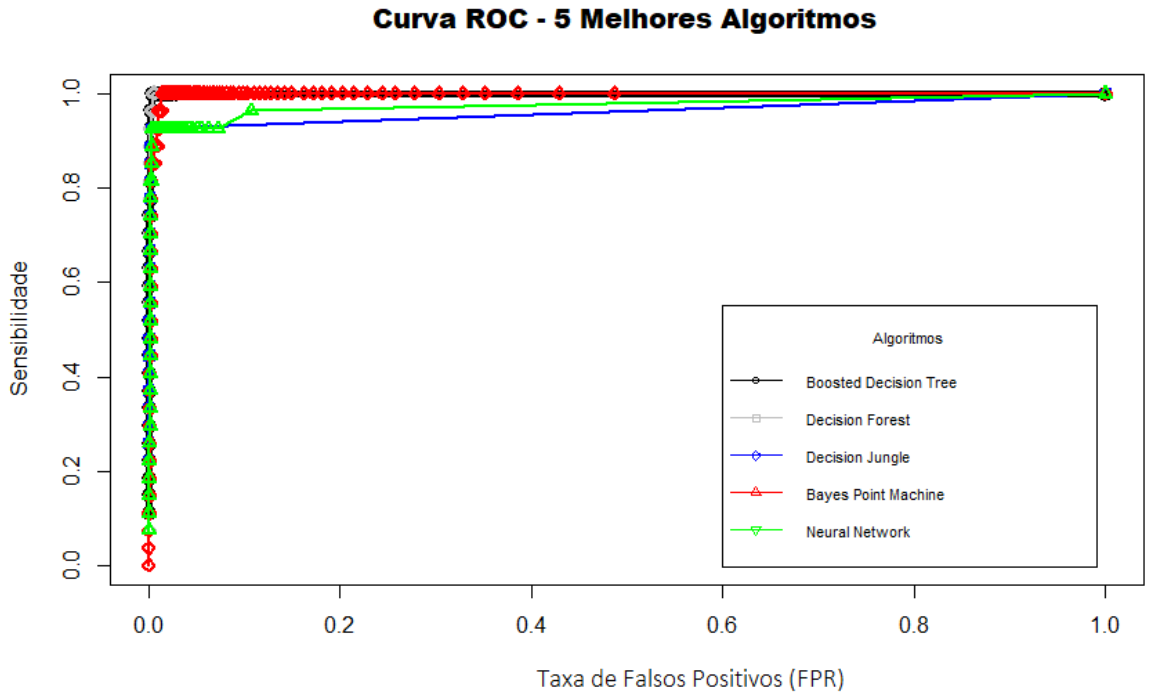

Figura 8 – Curva ROC para os 5 melhores algoritmos.

falsos positivos, e, que a curva PR é composta pelas métricas *precisão* e *sensibilidade*, podemos inferir que algoritmos que retornaram um baixo valor para a métrica *f1 score* na Tabela 9, métrica que é composta pela média harmônica entre a *precisão* e *sensibilidade*, não retornarão bons resultados nas suas curvas ROC e PR. Levando isto em consideração, limitamos a quantidade de algoritmos para os 5 melhores colocados para a elaboração e análise das referidas curvas.

Sob a ótica das famílias de classificadores aos quais esses 5 algoritmos pertencem, podemos observar que os três melhores resultados pertencem à família dos classificadores de árvore de decisão, seguido de um resultado pertencente à família dos classificadores estatísticos e um resultado pertencente à família dos classificadores discriminativos.

Na Figura 8 é apresentada a curva ROC para os cinco melhores algoritmos. Nos gráficos apresentados, podemos visualmente observar que os algoritmos *Boosted Decision Tree* e *Decision Forest* apresentaram um desempenho praticamente idêntico, seguido pelo desempenho dos algoritmos *Bayes Point Machine* e *Neural Network*. O algoritmo *Decision Jungle* apresentou o pior resultado entre os cinco algoritmos analisados. Analisando o conjunto dos resultados, os algoritmos da família de classificadores de árvore de decisão obtiveram resultados ligeiramente melhores que os algoritmos da família dos classificadores estatísticos.

Na Figura 9 é apresentada a curva PR para os mesmos cinco algoritmos selecionados através da Tabela 9. Nos gráficos apresentados, podemos visualmente observar que o

algoritmo *Boosted Decision Tree* apresentou um desempenho superior aos demais. Também podemos notar que os algoritmos *Decision Jungle* e *Decision Forest* apresentaram um desempenho similar, novamente mostrando uma vantagem dos algoritmos pertencentes à família dos classificadores de árvore de decisão. Os algoritmos *Neural Network* e *Bayes Point Machine* apresentaram os piores resultados entre os algoritmos testados.

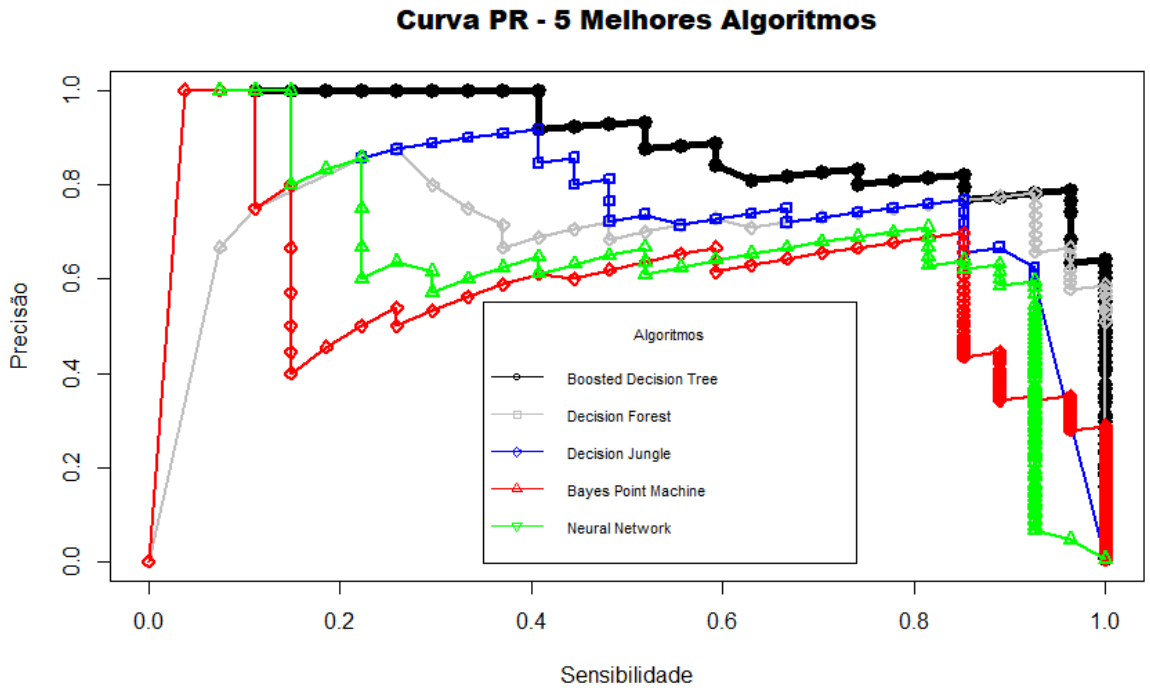

Figura 9 – Curva PR para os 5 melhores algoritmos.

Com base nos resultados visualmente apresentados pela curva ROC e curva PR, podemos calcular a área sob ambas as curvas apresentadas, sendo estas áreas denominadas AUROC (*Area under ROC Curve*) para a área sob a curva ROC e AUPR (*Area under Precision Recall Curve*) para a área sob a curva PR. O cálculo da área das curvas permite medir de maneira objetiva o desempenho de um ou mais algoritmos quando não há uma distinção visual clara dos resultados. O cálculo das AUROC e AUPR foi efetuado no *software* RStudio[6] através do uso da função *trapz*, parte da biblioteca *pracma*[7]. Os resultados dos cálculos de área dos cinco melhores algoritmos são apresentados na Tabela 10 e ordenados de maneira decrescente com base nos valores obtidos pela AUROC.

[6] https://www.rstudio.com
[7] https://cran.r-project.org/web/packages/pracma/pracma.pdf

Tabela 10 – Resultados dos cálculos da AUROC e AUPR.

Algoritmo	AUROC	AUPR
Boosted Decision Tree	0,99944	0,89214
Decision Forest	0,99892	0,70383
Bayes Point Machine	0,99735	0,59142
Neural Network	0,97490	0,65772
Decision Jungle	0,96208	0,75545

4.4 Discussão dos Resultados

Os resultados apresentados na Tabela 9, que apresenta os valores das métricas calculadas para os diferentes algoritmos, na Tabela 10, que demonstra o cálculo da AUROC e AUPR, e nas Figuras 8 e 9, que exibem visualmente a performance dos algoritmos, permitem-nos obter as seguintes conclusões.

Primeiramente, baseando-se nos resultados apresentados pela Tabela 9 e seguindo os critérios definidos previamente, podemos concluir que os três algoritmos pertencentes à família dos classificadores de árvores de decisão superaram os demais. Em particular, o algoritmo *Boosted Decision Tree* apresentou um *f1 score* de 0,86667 e uma *acurácia média* de 0,99845. Dos algoritmos pertencentes às demais famílias, apenas o *Bayes Point Machine*, pertencente à família dos classificadores estatísticos, se aproximou dos resultados obtidos com um *f1 score* de 0,76667.

Com base nos resultados da Tabela 9, também é possível verificar que o uso da métrica *acurácia* não trouxe resultados relevantes em cenários em que há um grande desbalanceamento entre as classes, visto que todos os algoritmos apresentaram desempenho superior a 0,99.

A Figura 8 mostra a curva ROC dos cinco melhores algoritmos selecionados a partir das métricas *f1 score*, *acurácia média* e *acurácia* apresentadas na Tabela 9. A análise da curva ROC nos permite observar que a performance dos algoritmos *Boosted Decision Tree* e *Decision Forest*, ambos pertencentes à família dos classificadores de árvore de decisão, apresentaram resultados próximos ao quadrante que representa o valor ideal. Pelo gráfico, também é possível observar que o desempenho dos algoritmos supracitados foi seguido pelo algoritmo *Bayes Point Machine*. Por fim, os piores desempenhos observados foram dos algoritmos *Decision Jungle* e *Neural Network*.

A Figura 9 apresenta a Curva PR dos mesmos cinco algoritmos selecionados através dos critérios de métricas definidos anteriormente e apresentados na Tabela 9. A análise da curva PR nos permite observar que o desempenho do algoritmo *Boosted Decision Tree* foi visualmente superior aos demais. O desempenho deste foi seguido pelos algoritmos

Decision Jungle e *Decision Forest*, também membros da família dos classificadores de árvore de decisão, que apresentaram resultados visualmente similares. Também podemos observar que os piores desempenhos foram apresentados pelos algoritmos *Bayes Point Machine* e *Neural Network*.

Por fim, os resultados apresentados na Tabela 10 possibilitam mensurar o desempenho dos algoritmos anteriormente mencionados através do cálculo das áreas sob cada uma das curvas de desempenho dos gráficos, quando analisados individualmente. A AUROC e a AUPR auxiliam na escolha dos melhores algoritmos em casos no qual a distinção visual não está clara, como na Figura 8 e na Figura 9, respectivamente, gerando um único valor para cada classificador, independentemente do valor de *threshold* utilizado [48]. Analisando os dados computados sobre a área das curvas, apresentados pela Tabela 10, podemos concluir que o desempenho do algoritmo *Boosted Decision Tree* foi superior aos demais algoritmos testados. O desempenho deste foi seguido pelos algoritmos *Decision Jungle* e *Decision Forest* que apresentaram resultados similares entre si, demonstrando que os algoritmos da família dos classificadores de árvore de decisão se mostraram superiores aos demais para este conjunto de dados. Ao final, os algoritmos *Neural Network* e *Bayes Point Machine* apresentaram os piores resultados.

Desta maneira, considerando o conjunto dos resultados apresentados na Tabela 9, que apresenta os valores das métricas, com as análises apresentadas nas curva ROC (Figura 8) e na curva PR (Figura 9), e, pela computação das áreas sob as curvas apresentada na Tabela 10 podemos afirmar que o algoritmo *Boosted Decision Tree* foi o algoritmo que apresentou os melhores resultados, sendo a melhor opção para a geração do modelo de aprendizado a ser utilizado na Etapa de Classificação quando utilizado com este conjunto de dados.

O principal motivo que permitiu ao algoritmo *Boosted Decision Tree* ser considerado o melhor dentre todos os algoritmos testados é referente ao aprendizado do tipo *ensemble*, uma técnica que combina a capacidade de predição de múltiplos algoritmos de modo a fornecer um único modelo com maior poder de predição [48]. No caso deste algoritmo, múltiplas árvores de decisão foram combinadas para melhorar a qualidade da predição. O prefixo *boost* no nome do algoritmo se deve ao fato que o seu processo de *ensemble* se baseia no uso da técnica chamada *boosting*. Nessa técnica, o conjunto de dados é aplicado sequencialmente a uma série de classificadores simples, como são as árvores de decisão, e é atribuído um peso a cada classificador utilizado conforme sua capacidade de acerto. Desta forma, a somatória ponderada de cada um dos classificadores individuais irá compor um modelo de aprendizado com uma capacidade superior de classificação [54].

A técnica de *boosting* teve outro papel importante para esse classificador, já que o auxiliou a lidar com o forte desbalanceamento entre as classes, penalizando as combinações de árvores de decisão que apresentaram resultados ruins. Durante a etapa de

treinamento, o conjunto de dados de treinamento possuía um desbalanceamento de 182:1, enquanto que o conjunto de dados de testes possuía um desbalanceamento de 162:1. Dessa maneira, os algoritmos de classificação supervisionada necessitaram lidar com esse pesado desbalanceamento para a geração e avaação do modelo de aprendizagem gerado. Os desbalanceamentos são decorrentes da conversão dos pacotes Modbus TCP/IP [26] em fluxos IP. Conforme pode ser observado na Tabela 9, esse forte desbalanceamento prejudicou algoritmos mais simples como o *Support Vector Machine* e o *Averaged Perceptron* dentre outros. Por outro lado, algoritmos mais complexos como o *Decision Forest* e *Decision Jungle*, que também utilizam a técnica *ensemble*, apresentaram bons resultados sobre o mesmo conjunto de dados.

5 CONCLUSÃO

Sistemas SCADA possuem uma grande importância no nosso cotidiano, já que monitoram e controlam diversos processos industriais. Conforme esses sistemas forem gradativamente sendo conectados à Internet, eles estarão sujeitos aos mesmos riscos que qualquer outro dispositivo conectado à rede mundial de computadores, porém, com uma diferença importante: falhas no funcionamento desses sistemas podem causar graves prejuízos financeiros ou ambientais ao mesmo tempo em que afetam a vida de uma grande quantidade de usuários.

Em vista disso, neste trabalho estudamos a utilização de algoritmos de aprendizado supervisionado na detecção de intrusões em redes industriais. Para realização do estudo, foi proposto um modelo, o qual se inicia pelo agrupamento de pacotes coletados na rede industrial em fluxos IP. Na sequência, extraímos algumas características baseadas em atributos e dados estatísticos dos fluxos IP. Por fim, comparamos os resultados de diversas famílias de classificadores de aprendizado de máquina em busca daqueles que melhor se adéquem às peculiaridades do tráfego. No presente trabalho, também abordamos as questões relacionadas ao desbalanceamento de classes no tráfego analisado.

Em um sistema SCADA, o tráfego de rede entre seus dispositivos é distinto do tráfego de uma rede corporativa. A principal característica desse tráfego é consequência da ação de processos periódicos de aquisição de dados e de controle de processos, característico da comunicação entre MTUs e RTUs. A essa ação é dado o nome de *polling*. Durante a avaliação do modelo proposto utilizamos um conjunto de dados publicamente disponível gerado pelos autores Lemay e Fernandez [21] como entrada da etapa de coleta de dados. Em seguida, na etapa de geração de fluxos agrupamos o conjunto de dados utilizado em fluxos IP unidirecionais, agregando pacotes que possuem em comum o mesmo IP de origem e destino, mesmas portas de origem e destino assim como o mesmo protocolo de comunicação. Esse agrupamento provê uma visão geral sobre a interação entre os dispositivos da rede industrial além de ter como vantagem a redução, significativa, da quantidade de dados a serem analisados. Por outro lado, o agrupamento aumentou o desbalanceamento entre as classes estudadas, sendo elas a de tráfego normal e a de tráfego malicioso. O desbalanceamento resultante da conversão dos pacotes em fluxos IP exigiu o uso de métricas como o *f1 score*, a *acurácia média*, curva ROC, curva PR e AUROC e AUPR de modo que pudéssemos avaliar os resultados dos classificadores após a criação dos modelos de aprendizagem.

Os resultados demonstraram que algumas famílias de classificadores apresentaram uma melhor resposta ao conjunto de dados utilizado do que outras. A família de classificadores de árvore de decisão apresentou os melhores resultados, e, em particular o

algoritmo *Boosted Decision Tree* apresentou os melhores resultados dentre os algoritmos avaliados. Acreditamos que seu bom desempenho esteve diretamente relacionado ao uso de *ensemble* na construção do algoritmo, uma técnica na qual diversos classificadores são combinados a fim de obterem, em conjunto, a melhor classificação de um conjunto de dados. Além disso, o uso de *ensemble* proveu suporte ao tráfego desbalanceado. Esse suporte é uma consequência da utilização do *ensemble*, na qual ocorrem penalizações por erros de classificação durante a criação do modelo de aprendizado. Em relação às demais famílias de classificadores, os algoritmos estatísticos apresentaram um resultado apenas razoável para o algoritmo *Bayes Point Machine*, e, a família dos classificadores discriminativos não apresentou bons resultados em nenhuma das métricas utilizadas, indicando que esses tipos de algoritmos não são adequados para este tráfego.

Como trabalhos futuros, sugere-se o foco em três áreas: o conjunto de dados, as características extraídas e ataques. Há uma carência muito grande de conjuntos de dados representantes do tráfego de redes industriais, e ainda maior para conjuntos de dados rotulados, assim uma área de interesse estaria relacionada à obtenção de novos conjuntos de dados, gerado ou capturado de alguma rede industrial existente. A segunda área de foco seria em relação às características extraídas, na qual a extração de outras características pode melhorar os resultados obtidos. A terceira área de foco seria a verificação da classificação sobre um novo conjunto de ataques, no qual seria interessante analisar o comportamento dos algoritmos quando um ou mais ataques poderiam ocorrer simultaneamente.

REFERÊNCIAS

[1] KNAPP, E. D.; LANGILL, J. T. *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. [S.l.]: Syngress, 2014.

[2] NAZIR, S.; PATEL, S.; PATEL, D. Assessing and augmenting SCADA cyber security: A survey of techniques. *Computers & Security*, Elsevier, v. 70, p. 436–454, 2017.

[3] BHUYAN, M. H.; BHATTACHARYYA, D. K.; KALITA, J. K. Network anomaly detection: methods, systems and tools. *IEEE communications surveys & tutorials*, IEEE, v. 16, n. 1, p. 303–336, 2014.

[4] MITCHELL, R.; CHEN, I.-R. A survey of intrusion detection techniques for Cyber-physical Systems. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 46, n. 4, p. 55:1–55:29, mar. 2014. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/2542049>.

[5] HUMAYED, A. et al. Cyber-Physical Systems Security—A Survey. *IEEE Internet of Things Journal*, v. 4, n. 6, p. 1802–1831, Dec 2017. ISSN 2327-4662.

[6] STOUFFER, K. et al. *Guide to Industrial Control Systems Security, SP 800-82*. [S.l.]: Gaithersburg: NIST, 2015.

[7] LINDA, O.; VOLLMER, T.; MANIC, M. Neural network based intrusion detection system for critical infrastructures. In: IEEE. *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. [S.l.], 2009. p. 1827–1834.

[8] YANG, Y. et al. Intrusion detection system for IEC 60870-5-104 based SCADA networks. In: IEEE. *Power and Energy Society General Meeting (PES), 2013 IEEE*. [S.l.], 2013. p. 1–5.

[9] JUNEJO, K. N.; GOH, J. Behaviour-based attack detection and classification in cyber physical systems using machine learning. In: ACM. *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*. [S.l.], 2016. p. 34–43.

[10] UDD, R. et al. Exploiting Bro for intrusion detection in a SCADA system. In: *Proceedings of the 2Nd ACM International Workshop on Cyber-Physical System Security*. New York, NY, USA: ACM, 2016. (CPSS '16), p. 44–51. ISBN 978-1-4503-4288-9. Disponível em: <http://doi.acm.org/10.1145/2899015.2899028>.

[11] LOUKAS, G. *Cyber-physical attacks: A growing invisible threat*. [S.l.]: Butterworth-Heinemann, 2015.

[12] HUITSING, P. et al. Attack taxonomies for the modbus protocols. *International Journal of Critical Infrastructure Protection*, Elsevier, v. 1, p. 37–44, 2008.

[13] SWALES, A. et al. Open ModBus/TCP specification. *Schneider Electric*, v. 29, 1999.

[14] ZURAWSKI, R. *Industrial communication technology handbook*. [S.l.]: CRC Press, 2014.

[15] IEEE. IEEE standard for electric power systems communications-distributed network protocol (DNP3). *IEEE Std 1815-2012 (Revision of IEEE Std 1815-2010)*, p. 1–821, Oct 2012.

[16] MACKIEWICZ, R. E. Overview of IEC 61850 and benefits. In: IEEE. *Power Systems Conference and Exposition, 2006. PSCE'06. 2006 IEEE PES*. [S.l.], 2006. p. 623–630.

[17] PIGGIN, R. Are industrial control systems ready for the cloud? *International Journal of Critical Infrastructure Protection*, Elsevier Science Publishers BV, v. 9, n. C, p. 38–40, 2015.

[18] GOETZ, E.; SHENOI, S. *Critical Infrastructure Protection*. [S.l.]: Springer Heidelberg, 2008.

[19] GOOSE, S.; KIRSCH, J.; WEI, D. SKYDA: cloud-based, secure SCADA-as-a-service. *International Transactions on Electrical Energy Systems*, Wiley Online Library, v. 25, n. 11, p. 3004–3016, 2015.

[20] DíAZ. Using SNORT for intrusion detection in MODBUS TCP/IP communications. *SANS Institute InfoSec Reading Room*, SANS Institute, 2011.

[21] LEMAY, A.; FERNANDEZ, J. M. Providing SCADA network data sets for intrusion detection research. *9th USENIX Workshop on Cyber Security Experimentation and Test (CSET '16)*, Usenix, p. 1–8, 2016.

[22] BROOKS, P. Ethernet/IP-industrial protocol. In: IEEE. *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*. [S.l.], 2001. v. 2, p. 505–514.

[23] BENDER, K. *Profibus: the fieldbus for industrial automation*. [S.l.]: Prentice-Hall, Inc., 1993.

[24] SIEMENS. *What properties, advantages and special features does the S7 protocol offer?* 2007. Disponível em: <https://support.industry.siemens.com/cs/document/26483647/what-properties-advantages-and-special-features-does-the-s7-protocol-offer-?dti=0&lc=en-WW>.

[25] KANG, D.-J. et al. Analysis on cyber threats to SCADA systems. In: IEEE. *Transmission & Distribution Conference & Exposition: Asia and Pacific, 2009*. [S.l.], 2009. p. 1–4.

[26] ORGANIZATION, I. M. *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b*. 2006. Disponível em: <http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf>.

[27] YUSSOF, S. et al. Financial impacts of smart meter security and privacy breach. In: IEEE. *Information Technology and Multimedia (ICIMU), 2014 International Conference on*. [S.l.], 2014. p. 11–14.

[28] TOCH, E. et al. The privacy implications of cyber security systems: A technological survey. *ACM Computing Surveys (CSUR)*, ACM, v. 51, n. 2, p. 36, 2018.

[29] OGIE, R. I. Cyber security incidents on critical infrastructure and industrial networks. In: *Proceedings of the 9th International Conference on Computer and Automation Engineering*. New York, NY, USA: ACM, 2017. (ICCAE '17), p. 254–258. ISBN 978-1-4503-4809-6. Disponível em: <http://doi.acm.org/10.1145/3057039.3057076>.

[30] PONEMON, I. *Critical Infrastructure: Security Preparedness and Maturity*. 2014. Disponível em: <https://www.hunton.com/files/upload/Unisys_Report_Critical_Infrastructure_Cybersecurity.pdf>.

[31] MILLER, B.; ROWE, D. A survey SCADA of and critical infrastructure incidents. In: ACM. *Proceedings of the 1st Annual conference on Research in information technology*. [S.l.], 2012. p. 51–56.

[32] SYMANTEC. *Dragonfly: Cyberespionage attacks against energy suppliers*. 2014. Disponível em: <https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/Dragonfly_Threat_Against_Western_Energy_Suppliers.pdf>.

[33] KSHETRI, N.; VOAS, J. Hacking power grids: A current problem. *Computer*, IEEE, v. 50, n. 12, p. 91–95, 2017.

[34] HASSANZADEH, A.; MODI, S.; MULCHANDANI, S. Towards effective security control assignment in the industrial internet of things. In: IEEE. *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. [S.l.], 2015. p. 795–800.

[35] FALLIERE, N. et al. *W32. Stuxnet Dossier: Version 1.4 (February 2011)*. Symantec Corporation, 2011. Disponível em: <https://books.google.com.br/books?id=Bm2vnQAACAAJ>.

[36] RUBIO, J. E. et al. Analysis of intrusion detection systems in industrial ecosystems. In: *14th International Conference on Security and Cryptography (SECRYPT 2017)*. [S.l.: s.n.], 2017.

[37] PAN, S.; MORRIS, T. H.; ADHIKARI, U. A specification-based intrusion detection framework for cyber-physical environment in electric power system. *IJ Network Security*, v. 17, n. 2, p. 174–188, 2015.

[38] VACCA, J. R. *Computer and information security handbook*. [S.l.]: Newnes, 2009.

[39] VASILOMANOLAKIS, E. et al. Multi-stage attack detection and signature generation with ICS honeypots. In: *NOMS*. [S.l.: s.n.], 2016. p. 1227–1232.

[40] CARCANO, A. et al. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Transactions on Industrial Informatics*, IEEE, v. 7, n. 2, p. 179–186, 2011.

[41] FAURI SANTOS, C. H. E. T. From system specification to anomaly detection (and back). In: ACM. *CPS '17 Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy*. [S.l.], 2017. p. 13–24.

[42] CHOUDHARY, S.; SRINIVASAN, B. Usage of netflow in security and monitoring of computer networks. *International Journal of Computer Applications*, Foundation of Computer Science, v. 68, n. 24, 2013.

[43] HOFSTEDE, R. et al. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 4, p. 2037–2064, 2014.

[44] CLAISE, B.; TRAMMELL, B.; AITKEN, P. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information (2013)*. [S.l.]: RFC 7011, 2013.

[45] CLAISE, B. Cisco systems netflow services export version 9. 2004.

[46] QUITTEK, J. et al. RFC 3917: requirements for IP flow information export: IPFIX. *Published by Internet Engineering Task Force (IETF). Internet Society (ISOC) RFC Editor. USA. out*, 2004.

[47] CLAISE, B.; TRAMMEL, B. *Information Model for IP Flow Information Export (IPFIX)—RFC 7012*. [S.l.], 2013.

[48] HE, H.; MA, Y. *Imbalanced learning: foundations, algorithms, and applications*. [S.l.]: John Wiley & Sons, 2013.

[49] BEAVER, J. M.; BORGES-HINK, R. C.; BUCKNER, M. A. An evaluation of machine learning methods to detect malicious SCADA communications. In: IEEE. *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*. [S.l.], 2013. v. 2, p. 54–59.

[50] MORRIS, T. H.; THORNTON, Z.; TURNIPSEED, I. *Industrial control system simulation and data logging for intrusion detection system research*. 2015.

[51] GOH, J. et al. A dataset to support research in the design of secure water treatment systems. In: SPRINGER. *International Conference on Critical Information Infrastructures Security*. [S.l.], 2016. p. 88–99.

[52] BODENHEIM, R. et al. Evaluation of the ability of the shodan search engine to identify internet-facing industrial control devices. *International Journal of Critical Infrastructure Protection*, Elsevier, v. 7, n. 2, p. 114–123, 2014.

[53] WILLIAMS, P. M. *Distinguishing internet-facing ICS devices using PLC programming information*. [S.l.], 2014.

[54] HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning*. [S.l.: s.n.], 2001.

[55] CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning. *IEEE Transactions on Neural Networks*, IEEE, v. 20, n. 3, p. 542–542, 2009.

[56] SCHUSTER, F. et al. Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks. In: IEEE. *Computational Intelligence, 2015 IEEE Symposium Series on*. [S.l.], 2015. p. 83–90.

[57] JOSE, C. et al. Local Deep Kernel Learning for Efficient Non-linear SVM Prediction. In: *Proceedings of the International Conference on Machine Learning*. [S.l.: s.n.], 2013.

[58] MICROSOFT. *rxNeuralNet: Neural Net*. 2017. Disponível em: <https://docs.microsoft.com/en-us/machine-learning-server/r-reference/microsoftml/rxneuralnet>.

[59] DUA, S.; DU, X. *Data mining and machine learning in cybersecurity*. [S.l.]: CRC press, 2016.

[60] MICROSOFT. *rxFastForest: Fast Forest*. 2017. Disponível em: <https://docs.microsoft.com/en-us/machine-learning-server/r-reference/microsoftml/rxfastforest>.

[61] SHOTTON, J. et al. Decision jungles: Compact and rich models for classification. In: *Proc. NIPS*. [s.n.], 2013. Disponível em: <https://www.microsoft.com/en-us/research/publication/decision-jungles-compact-and-rich-models-for-classification/>.

[62] MICROSOFT. *rxLogit: Logistic Regression*. 2017. Disponível em: <https://docs.microsoft.com/en-us/machine-learning-server/r-reference/revoscaler/rxlogit>.

[63] MICROSOFT. *rxNaiveBayes: Parallel External Memory Algorithm for Naive Bayes Classifiers*. 2017. Disponível em: <https://docs.microsoft.com/en-us/machine-learning-server/r-reference/revoscaler/rxnaivebayes>.

[64] BARBOSA, R. R. R.; SADRE, R.; PRAS, A. Flow whitelisting in SCADA networks. *International journal of critical infrastructure protection*, Elsevier, v. 6, n. 3-4, p. 150–158, 2013.

[65] ZHAO, Y.; SHEN, Z.-j. APPLICATION OF TCP/IP BASED IEC60870-5-104 TELECONTROL PROTOCOL IN POWER SYSTEM [J]. *Power System Technology*, v. 10, p. 016, 2003.

[66] CISCO. *What is Snort?* 2018. Disponível em: <https://www.snort.org/faq/what-is-snort>.

[67] HINK, R. C. B. et al. Machine learning for power system disturbance and cyber-attack discrimination. In: *2014 7th International symposium on resilient control systems (ISRCS)*. [S.l.: s.n.], 2014. p. 1–8.

[68] NADER, P.; HONEINE, P.; BEAUSEROY, P. Detection of cyberattacks in a water distribution system using machine learning techniques. In: IEEE. *Digital Information Processing and Communications (ICDIPC), 2016 Sixth International Conference on*. [S.l.], 2016. p. 25–30.

[69] INOUE, J. et al. Anomaly detection for a water treatment system using unsupervised machine learning. In: IEEE. *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*. [S.l.], 2017. p. 1058–1065.

[70] PONOMAREV, S.; ATKISON, T. Industrial control system network intrusion detection by telemetry analysis. *IEEE Transactions on Dependable and Secure Computing*, IEEE, n. 1, p. 1–1, 2016.

[71] PAXSON, V. Bro: A system for detecting network intruders in real-time. In: *Proceedings of the 7th Conference on USENIX Security Symposium - Volume 7*. Berkeley, CA, USA: USENIX Association, 1998. (SSYM'98), p. 3–3. Disponível em: <http://dl.acm.org/citation.cfm?id=1267549.1267552>.

[72] KREIMEL, P.; EIGNER, O.; TAVOLATO, P. Anomaly-based detection and classification of attacks in cyber-physical systems. In: ACM. *Proceedings of the 12th International Conference on Availability, Reliability and Security*. [S.l.], 2017. p. 40.

[73] SIDDAVATAM, I. A. et al. An ensemble learning for anomaly identification in SCADA system. In: IEEE. *2017 7th International Conference on Power Systems (ICPS)*. [S.l.], 2017. p. 457–462.

[74] YUSHENG, W. et al. Intrusion detection of industrial control system based on Modbus TCP protocol. In: IEEE. *Autonomous Decentralized System (ISADS), 2017 IEEE 13th International Symposium on*. [S.l.], 2017. p. 156–162.

[75] HAAG, P. NFDUMP-NetFlow processing tools. *URL: https://github.com/phaag/nfdump*, 2018.

[76] GROUP, T. P. *What is PHP?* 2017. Disponível em: <https://secure.php.net/manual/en/intro-whatis.php>.

[77] GROUP, T. P. *Using PHP from the command line*. 2017. Disponível em: <http://php.net/manual/en/features.commandline.php>.

[78] BARNES, J. *Azure Machine Learning Microsoft Azure Essentials*. [S.l.]: Microsoft Press: Redmond, Washington, 2015.

[79] MICROSOFT. *How to choose algorithms for Microsoft Azure Machine Learning*. 2017. Disponível em: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice>.

[80] MALOOF, M. A. *Machine learning and data mining for computer security: methods and applications*. [S.l.]: Springer, 2006.

Apêndices

APÊNDICE A – CONFIGURAÇÕES UTILIZADAS NOS ALGORITMOS DE APRENDIZADO DE MÁQUINA

Neste apêndice são apresentadas as configurações ótimas utilizadas pelos algoritmos de classificação supervisionada.

Tabela 11 – Parâmetros utilizados em cada algoritmo de classificação supervisionada.

Algoritmo	**Parâmetros**	**Configuração Ótima**
Support Vector Machine	**Create trainer mode:** Parameter Range **Number of iterations:** 1 - 4 **Lambda:** 1^{-6} - 1.00^{-1} **Normalize features:** True **Project to the unit-sphere:** False **Allow unknown categorical levels:** True	 3 $0,08633273$
Locally-Deep Support Vector Machine	**Create trainer mode:** Parameter Range **Depth of the tree:** 1 - 9 **Lambda W:** 3.00^{-2} - 3.00^{-1} **Lambda Theta:** 5.00^{-3} - 3.00^{-2} **Lambda Theta Prime:** 5.00^{-3} - 3.00^{-2} **Sigmoid sharpness:** 3.00^{-1} - 3.00 **Number of iterations:** 5000 - 45000 **Feature normalizer:** Min-Max normalizer **Allow unknown categorical levels:** True	 1 0.1246033 0.0269226134 0.00744694937 2.02324367 26806
Averaged Perceptron	**Create trainer mode:** Parameter Range **Learning rate:** 3.00^{-1} - 1.00 **Maximum number of iterations:** 1 - 30 **Allow unknown categorical levels:** True	 0.938622832 29

Neural Network	**Create trainer mode:** Parameter Range **Hidden layer specification:** Fully-connected case **Number of hidden nodes:** 100 **Learning rate:** $3.00^{-}2$ - 3.00^{-1} **Number of learning iterations:** 30 - 300 **The initial learning weights:** 0.1 **The momentum:** 0 **The type of normalizer:** Min-Max normalizer **Shuffle examples:** True **Allow unknown categorical levels:** True	 0.132147461 199
Logistic Regression	**Create trainer mode:** Parameter Range **Optimization tolerance:** 10^{-7} **L1 regularization weight:** 3.00^{-1} - 1.00 **L2 regularization weight:** 3.00^{-1} - 1.00 **Memory size for L-BFGS:** 7 - 60 **Allow unknown categorical levels:** True	 0.919831634 0.8993623 38
Bayes Point Machine	**Number of training iterations:** 30 **Include bias:** True **Allow unknown values in categorical features:** True	
Boosted Decision Tree	**Create trainer mode:** Parameter Range **Maximum number of leaves per tree:** 7 - 60 **Minimum number of samples per leaf node:** 3 - 30 **Learning rate:**7.00^{-2} - 6.00^{-1} **Number of trees constructed:** 3 - 300 **Allow unknown categorical levels:** True	 50 26 0.432686 32
Decision Forest	**Resampling method:** Bagging **Create trainer mode:** Parameter Range **Number of decision trees:** 3 - 24 **Maximum depth of the decision trees:** 10 - 100 **Number of random splits per node:** 42 - 384 **Minimum number of samples per leaf node:** 1 - 4 **Allow unknown values for categorical features:** True	 11 12 122 2

Decision Jungle	**Resampling method:** Bagging **Create trainer mode:** Parameter Range **Number of decision DAGs:** 2 - 24 **Maximum depth of the decision DAGs:** 10 - 100 **Maximum width of the decision DAGs:** 42 - 384 **Number of optimization steps per decision DAG layer:** 1024 - 6140 **Allow unknown values for categorical features:** True	 6 37 156 3882

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados pelo autor durante o programa.

1. Gabriel Vasquez, Rodrigo S. Miani, Bruno B. Zarpelão, **Flow-Based Intrusion Detection for SCADA networks using Supervised Learning**, XVII SBSEG, 11/2017, eBook, 168-181, 978-85-7669-422-96, (Qualis CC 2017, B3)