

Estudo da Influência das Features e do Desempenho dos Classificadores na Detecção de Anomalias em Redes

João Lucas Oliveira Mota¹, João Lucas Rodrigues da Silva², José Alberto Rodrigues Neto²

¹ Departamento de Engenharia de Teleinformática – Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brasil

² Departamento de Computação – Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brasil

{jluucasoliveira2002}@gmail.com, {joaolucas.rodrigues, jose.alberto}@alu.ufc.br

Abstract. *CHANGE ME*

Resumo. *CHANGE ME*

1. Introdução

CHANGE ME

Este artigo investiga a influência da seleção de atributos e do desempenho de diferentes classificadores na detecção de anomalias em tráfego de redes de computadores.

2. Dataset e pré-processamento

2.1. Dataset

O conjunto de dados utilizado foi obtido a partir da plataforma Kaggle, no repositório *Network Intrusion Detection*, disponibilizado publicamente por Sampada Bhosale [Bhosale 2018].

O dataset foi construído a partir da simulação de um ambiente de rede militar típico da Força Aérea dos Estados Unidos, desenvolvido para representar diferentes tipos de tráfego de rede, incluindo tanto comunicações legítimas quanto diversos tipos de ataques. Cada instância do conjunto de dados representa uma conexão de rede, contendo atributos como a quantidade de bytes transmitidos pela origem, a duração da conexão e outros parâmetros, conforme descrito na tabela apresentada no Apêndice.

Por fim, após as simulações realizadas de forma supervisionada, as conexões foram classificadas como tráfego normal ou como intrusão, fazendo com que a variável alvo seja do tipo binária, assumindo os valores *Normal* ou *Anomalia*.

2.2. Pré-processamento

A aplicação da Análise de Componentes Principais (PCA) como etapa de pré-processamento tem se mostrado eficaz para a redução da dimensionalidade dos dados sem perda significativa de informação. Estudos indicam que o PCA é capaz de preservar cerca de 99% da variância original mesmo com reduções superiores a 50% no número de atributos, conforme observado por Santos e Miani [Santos and Miani 2025].

Neste artigo, foram realizadas diferentes análises utilizando modelos de aprendizagem de máquina, com o objetivo de identificar as *features* de maior importância para

cada modelo. Em seguida, buscando simular um cenário mais próximo de um ambiente real, no qual nem sempre todos os atributos estão disponíveis para os algoritmos de análise, foram conduzidos testes com a remoção das *features* mais relevantes. Essa abordagem teve como objetivo analisar a influência de cada atributo no desempenho dos modelos, bem como avaliar a robustez de cada método frente à ausência parcial de informações.

3. Metodologia

3.1. Modelos Utilizados

3.1.1. Random Forest

O Random Forest (RF) é um algoritmo de aprendizado de máquina supervisionado baseado em conjuntos de árvores de decisão, no qual múltiplas árvores são treinadas a partir de subconjuntos aleatórios dos dados e dos atributos, combinando suas previsões por meio de votação majoritária. Sua utilização neste trabalho se deve à robustez frente a dados de alta dimensionalidade, à capacidade de modelar relações não lineares e à possibilidade de analisar a importância das *features*, características relevantes para problemas de detecção de intrusão em redes [Breiman 2001].

3.1.2. K-Nearest Neighbors

Outro modelo utilizado foi o K-Nearest Neighbors (KNN). Esse modelo se diferencia dos demais por não realizar uma etapa explícita de aprendizado a partir dos dados de treinamento. Em vez disso, ele armazena as instâncias conhecidas e classifica uma nova entrada com base na similaridade em relação a um número arbitrário (K) de exemplos mais próximos previamente registrados [Cover and Hart 1967].

3.1.3. Isolation Forest

O Isolation Forest (IF) é um algoritmo de aprendizado de máquina não supervisionado, projetado especificamente para a detecção de anomalias (ou outliers) em conjuntos de dados. A escolha do Isolation Forest foi motivada pelo fato de que, em alguns cenários, não é possível obter todos os dados rotulados necessários para a detecção de intrusões em redes. Dessa forma, buscou-se avaliar se o IF seria capaz de identificar anomalias mesmo na ausência parcial dos rótulos utilizados no treinamento [Liu et al. 2008].

3.1.4. Multilayer Perceptron (MLP)

O Multilayer Perceptron (MLP) é um modelo de rede neural artificial supervisionado, composto por camadas de neurônios interconectados que aplicam transformações não lineares sobre os dados de entrada. A escolha do MLP foi motivada por sua capacidade de aprender padrões complexos no tráfego de rede, sendo adequado para a detecção de intrusões que apresentam comportamentos sutis, embora seu desempenho dependa da escolha adequada de hiperparâmetros e do pré-processamento dos dados [Haykin 2009].

4. Resultados e Discussão

4.1. Resultados com K-Nearest Neighbors

CHANGE ME (COLOCAR TABELA) Após alguns testes, os resultados mais precisos foram encontrados com $K = 1$, resultando em um F1-Score de 0.9946. Provavelmente, esse comportamento ocorre devido à grande quantidade de dados disponíveis para cada entrada de treinamento, o que torna a maioria dos ataques mais evidentes. Valores superiores de K levaram ao sobreajuste e ao decréscimo progressivo do desempenho do modelo.

Após uma análise baseada na remoção individual de cada atributo, observou-se que o parâmetro *hot* foi o mais decisivo para o modelo, reduzindo o F1-Score para 0.9931 quando removido. Outra informação relevante é que oito atributos, quando removidos, aumentaram a precisão do modelo. Após essa remoção, o F1-Score atingiu 0.9960, sendo o atributo *diff_srv_rate* o mais prejudicial, cuja exclusão resultou em um aumento de 0.0005 pontos no F1-Score.

4.2. Resultados com Isolation Forest

No entanto, o Isolation Forest se mostrou ineficiente quando comparado aos outros modelos utilizados durante a pesquisa, apresentando a menor acurácia (0.748). Entretanto, a acurácia não é a métrica mais confiável em conjuntos de dados desbalanceados, o que é comum em problemas de detecção de anomalias, pois pode ser inflacionada pela capacidade do modelo em classificar corretamente a classe majoritária (tráfego normal). Dessa forma, uma análise mais aprofundada torna-se necessária.

Table 1. Métricas de desempenho do Isolation Forest

| Métrica | Classe Normal | Classe Anomalia |
|-----------|---------------|-----------------|
| Precision | 1.00 | 0.65 |
| Recall | 0.53 | 1.00 |
| F1-Score | 0.69 | 0.79 |
| Support | 2690 | 2349 |

Como pode ser observado na Tabela 1, todos os eventos classificados como normais eram, de fato, normais. A principal dificuldade do modelo esteve na identificação correta das anomalias, apresentando precisão de 65% (0.65) ao rotular eventos como intrusão. Além disso, considerando todos os eventos analisados, o modelo foi capaz de identificar corretamente apenas 53% (0.53) dos eventos normais, o que indica a geração de um elevado número de falsos positivos, isto é, eventos normais classificados como ataques.

Por outro lado, o Isolation Forest não apresentou o problema de gerar falsos negativos, uma vez que ataques não foram classificados como eventos normais. Em síntese, o modelo apresenta dificuldades em verificar se um evento é realmente normal, gerando falsos positivos que podem ocasionar retrabalho para equipes de segurança, as quais precisam analisar manualmente eventos normais identificados como anômalos.

4.3. Resultados com Random Forest

CHANGE ME

4.4. Resultados com MLP

CHANGE ME

5. Conclusão

CHANGE ME

Este trabalho analisou a influência da seleção de atributos e do desempenho de diferentes classificadores na detecção de anomalias em redes de computadores.

References

- Bhosale, S. (2018). Network intrusion detection. Kaggle. Acesso em 2025.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. Prentice Hall, 3 edition.
- Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 413–422. IEEE.
- Santos, K. C. and Miani, R. S. (2025). Impacto da redução de dimensão e seleção de atributos na generalização de modelos de detecção de intrusão. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, Uberlândia, MG, Brazil. SBC.

Apêndice A — Descrição Completa do Dataset

Table 2. Descrição completa dos atributos do dataset de intrusão em redes

| Atributo | Descrição |
|-----------------------------|---|
| duration | Duração da conexão de rede |
| protocol_type | Protocolo utilizado na conexão (TCP, UDP ou ICMP) |
| service | Serviço de rede acessado |
| flag | Estado da conexão conforme o protocolo |
| src_bytes | Quantidade de bytes enviados pela origem |
| dst_bytes | Quantidade de bytes enviados pelo destino |
| land | Indica se origem e destino possuem mesmo IP e porta |
| wrong_fragment | Número de fragmentos incorretos |
| urgent | Número de pacotes marcados como urgentes |
| hot | Indicadores de comportamentos suspeitos |
| num_failed_logins | Número de tentativas de login malsucedidas |
| logged_in | Indica se o login foi realizado com sucesso |
| num_compromised | Número de condições comprometidas |
| root_shell | Indica obtenção de shell com privilégio root |
| su_attempted | Tentativas de uso do comando <i>su</i> |
| num_root | Número de acessos root |
| num_file_creations | Número de arquivos criados |
| num_shells | Número de shells abertos |
| num_access_files | Número de acessos a arquivos sensíveis |
| num_outbound_cmds | Número de comandos externos enviados |
| is_host_login | Indica login como host |
| is_guest_login | Indica login como convidado |
| count | Conexões com o mesmo host em janela de tempo |
| srv_count | Conexões com o mesmo serviço em janela de tempo |
| serror_rate | Taxa de erros SYN |
| srv_serror_rate | Taxa de erros SYN para o serviço |
| rerror_rate | Taxa de erros de resposta |
| srv_rerror_rate | Taxa de erros de resposta para o serviço |
| same_srv_rate | Taxa de conexões para o mesmo serviço |
| diff_srv_rate | Taxa de conexões para serviços diferentes |
| srv_diff_host_rate | Taxa de serviços acessando hosts distintos |
| dst_host_count | Número de conexões para o mesmo host destino |
| dst_host_srv_count | Número de conexões para o mesmo serviço no host |
| dst_host_same_srv_rate | Taxa de serviços iguais para o host destino |
| dst_host_diff_srv_rate | Taxa de serviços diferentes para o host destino |
| dst_host_same_src_port_rate | Taxa de conexões com mesma porta de origem |
| dst_host_srv_diff_host_rate | Taxa de serviços com hosts distintos |
| dst_host_serror_rate | Taxa de erros SYN no host destino |
| dst_host_srv_serror_rate | Taxa de erros SYN por serviço no host destino |
| dst_host_rerror_rate | Taxa de erros de resposta no host destino |
| dst_host_srv_rerror_rate | Taxa de erros de resposta por serviço no host destino |