Correlação Híbrida Baseada em *Stacking* para Detecção de Anomalias em Redes de Computadores

Franklin A. M. Venceslau[1], Rafael R. de Souza[1], Fabiano C. da Silva[1], José A. S. Monteiro[2]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Recife – PE – Brazil

[2]CESAR School
Recife, PE – Brazil.

{famv, rrs4, fcs3, suruagy}@cin.ufpe.br

Abstract. *Anomaly detection in computer networks is a critical challenge in the field of cybersecurity, due to the increasing complexity of threats and the dynamicity of data traffic. This study proposes an ensemble stacking-based approach that combines the Local Outlier Factor (LOF), Isolation Forest (iForest), and One-Class SVM (OCSVM) algorithms for anomaly detection. The scores generated by these models then train a Random Forest classifier, responsible for the final classification of traffic instances. Empirical validation was conducted with the UGR'16 and CIC-IDS2017 datasets and used metrics such as AUC, ROC curves, and F1-score, allowing us to evaluate the performance against traditional and state-of-the-art methods. The proposed solution shows promise in reducing false positives and detecting malicious traffic in realistic and imbalanced scenarios.*

Resumo. *A detecção de anomalias em redes de computadores é um desafio crítico no campo da cibersegurança, devido à crescente complexidade das ameaças e à dinamicidade do tráfego de dados. Este estudo propõe uma abordagem baseada em ensemble stacking, que combina os algoritmos Local Outlier Factor (LOF), Isolation Forest (iForest) e One-Class SVM (OCSVM) para a detecção de anomalias. Em seguida, os scores gerados por esses modelos treinam um classificador Random Forest, responsável pela classificação final das instâncias de tráfego. A validação empírica foi conduzida com o conjuntos de dados UGR'16 e CIC-IDS2017 e utilizaram métricas como AUC, curvas ROC e F1-score, permitindo avaliar o desempenho em relação a métodos tradicionais e do estado da arte. A solução proposta demonstra ser promissora na redução de falsos positivos e na detecção de tráfego malicioso em cenários realistas e desbalanceados.*

1. Introdução

Anomalias em redes de computadores são padrões estatísticos ou comportamentais que se desviam do tráfego legítimo esperado, podendo indicar falhas, intrusões ou atividades maliciosas não autorizadas [Ness 2024]. Identificar esses padrões anômalos em tempo hábil é essencial para mitigar danos financeiros e operacionais, além de garantir a segurança das informações [Lunardi et al. 2022].

Apesar de sua eficácia em cenários específicos, os métodos tradicionais de detecção de anomalias, baseados em assinaturas ou regras, apresentam limitações críticas. Além disso, esses métodos enfrentam dificuldades em acompanhar a crescente heterogeneidade dos padrões de tráfego em redes modernas, especialmente em ambientes complexos e de grande escala. Nesse contexto, as técnicas de aprendizado de máquina surgem como alternativas promissoras, oferecendo a capacidade de aprender e adaptar-se a diferentes padrões de tráfego sem a necessidade de intervenção manual constante.

No entanto, métodos isolados, como o *Local Outlier Factor* (LOF), *Isolation Forest* (IF) e *One-Class Support Vector Machine* (OCSVM), apresentam limitações inerentes, incluindo a sensibilidade à distribuição dos dados e a dificuldade de generalização em ambientes complexos. Estudos prévios demonstraram que abordagens baseadas em empilhamento (*stacking*) podem superar essas limitações. [Wang et al. 2021] ao investigar a detecção de anomalias em redes IoT, demonstrou que um modelo de empilhamento, combinando *Random Forest*, *Gradient Boosting* e *XGBoost*, superou métodos individuais, aumentando o valor do AUC (*Area Under the Curve*) em até 12%. Da mesma forma, [Li et al. 2020] sugerem que a aplicação de empilhamento com *Random Forest* como metamodelo em um ambiente com dados desbalanceados (80% benignos, 20% anômalos) resultou em uma melhoria de 15% no *F1-score* em relação ao melhor modelo individual.

Para abordar essas questões, este trabalho propõe uma abordagem baseada em *ensemble stacking*, combinando os algoritmos mencionados em uma estrutura que utiliza o classificador *Random Forest* como metamodelo para consolidar os resultados [Jeffrey et al. 2024]. O objetivo geral deste estudo é demonstrar como a combinação de algoritmos complementares em um *framework* de empilhamento pode melhorar significativamente a precisão da detecção de anomalias, reduzindo taxas de falsos positivos e aumentando a confiabilidade do sistema.

As demais seções deste artigo estão organizadas da seguinte forma: a Seção 2 apresenta os trabalhos relacionados, que associam mecanismos de detecção de intrusão a técnicas de aprendizagem de máquina; a Seção 3 descreve a proposta do *pipeline* de detecção e de treinamento do modelo; a Seção 4 apresenta os resultados, e a Seção 5 conclui e discute os trabalhos futuros.

2. Trabalhos Relacionados

Em [Tokmak and Nkongolo 2023], foi proposto um modelo baseado em *Stacked Autoencoder* (SAE) combinado com *Long Short-Term Memory* (LSTM) para a detecção e classificação de ameaças *zero-day*. Segundo os autores, o modelo obteve uma precisão de 98% utilizando o conjunto de dados UGRansome, sendo eficaz na identificação de ataques de assinatura, assinaturas sintéticas e anomalias.

Na pesquisa de [Chliah et al. 2023], os autores propuseram uma abordagem híbrida baseada em aprendizado supervisionado e não supervisionado para a detecção de anomalias em tráfego de rede, utilizando o motor de *big data* Apache Spark. Os experimentos indicaram que, com a aplicação do K-*means* para agrupamento de dados e KNN para detecção de anomalias, o modelo alcançou uma precisão de 99,94% ao utilizar validação cruzada com K-*folds* no conjunto completo de 48 *features*. Apesar da alta eficácia, os autores destacam que o desempenho pode variar com a escolha do número de *clusters*, sendo o valor ótimo K=2 para o conjunto avaliado.

Os trabalhos discutidos demonstram que técnicas de aprendizado de máquina, como *Autoencoders* Variacionais, *Stacked Autoencoders* com LSTM e métodos híbridos, têm impacto na detecção de anomalias e ataques cibernéticos, melhorando a precisão e reduzindo falsos positivos. Em geral, os estudos focam em abordagens específicas e por este fato se faz necessária a investigação de novas implementações de *ensemble stacking* objetivando desenvolver *frameworks* mais robustos, capazes de lidar com cenários de maior complexidade.

3. *Pipeline* de Detecção

Diferentemente de estudos anteriores que apenas agregam as saídas de modelos, propomos um mecanismo de empilhamento que explora as correlações cruzadas entre os detectores base para compor um vetor de características mais informativo, aumentando a separabilidade entre instâncias benignas e anômalas.

A Figura 1 ilustra a arquitetura do *pipeline* de aprendizado de máquina para detecção de anomalias. Inicialmente referenciamos os *datasets* utilizados UGR'16 e CIC-IDS2017 como entrada, contendo o tráfego de rede com dados normais e anômalos [Maciá-Fernández et al. 2018, Sharafaldin et al. 2018].

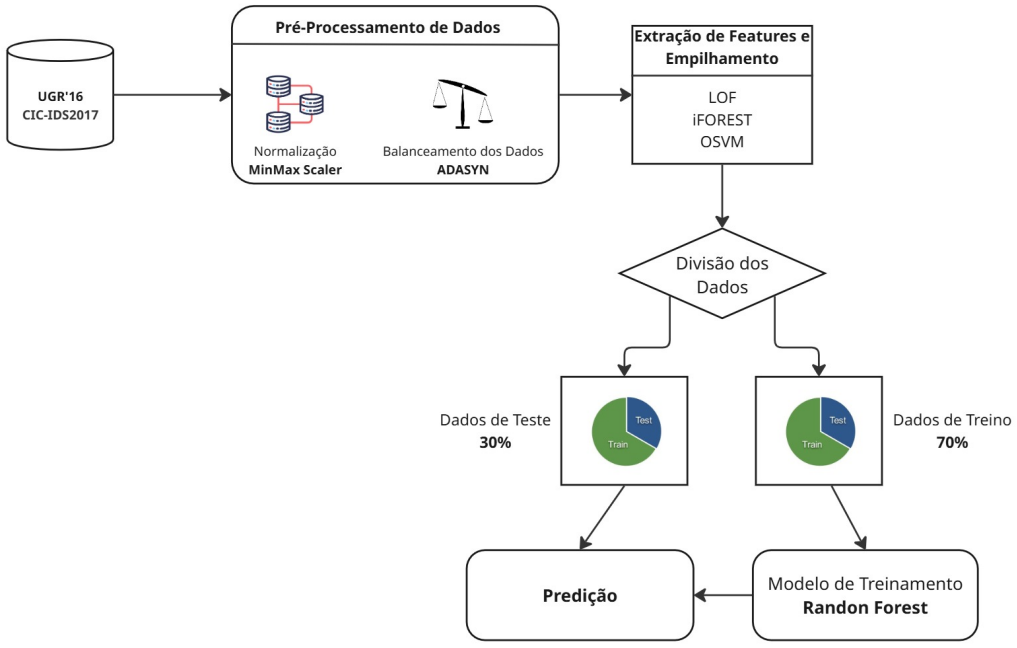

Figura 1. *Pipeline* do modelo proposto

Na etapa de pré-processamento, ajustamos os dados para que todas as variáveis fiquem na mesma escala e equilibramos as classes de dados (normais e anômalos) para evitar viés do modelo. Na etapa de empilhamento, combinamos as saídas dos três detectores de anomalias: LOF, IF e OCSVM. No processo de divisão de dados, processamos o *dataset* distribuindo em 70% para treinamento e 30% para teste, onde, em seguida, o metamodelo *Random Forest* utiliza os dados empilhados para treinar e detectar anomalias no tráfego de rede.

3.1. Extração de *Features*

Antes da extração de *features*, os dados brutos passam por um processo de normalização e balanceamento. Utilizamos a técnica *Min-Max Scaler* para normalizar as variáveis em uma faixa uniforme [0, 1], eliminando impactos de escalas heterogêneas que poderiam distorcer a detecção de *outliers*.

A extração de *features* é realizada a partir de três modelos de detecção de *outliers* que operam de maneira independente e, posteriormente, suas saídas são empilhadas para alimentar um metamodelo baseado em *Random Forest*. Implementamos as classes e métodos LOF obtidas a partir da biblioteca *Scikit-learn* [Pedregosa et al. 2011] e aplicamos ao conjunto de treinamento um número de vizinhos $k = 50$ e um nível de contaminação de 0,5%. O algoritmo avalia a densidade local de cada instância em relação aos seus vizinhos, produzindo um *score* negativo, onde valores mais baixos indicam maior probabilidade de anomalia. Em nossa abordagem, invertemos esse *score* para padronizar a interpretação dos valores e torná-los compatíveis com os demais modelos.

Esses *scores* são então incorporados ao espaço de *features* para a etapa de empilhamento. O IF constrói árvores binárias de maneira iterativa para particionar os dados, gerando um *score* baseado na profundidade média necessária para isolá-la. Em nossa abordagem, o modelo é treinado com um nível de contaminação de 0,5% e sua saída é calculada via função de decisão. Este *score*, ao contrário do LOF, é um valor contínuo, onde valores mais baixos indicam instâncias mais prováveis de serem anômalas. Ele é então empilhado junto às saídas dos outros modelos. O OCSVM mapeia os dados para um espaço de alta dimensionalidade usando um kernel RBF e aprende uma fronteira de decisão que separa os exemplos normais das anomalias. A configuração utilizada no código emprega $\gamma = 0{,}1$ e $\nu = 0{,}05$, ajustando a flexibilidade da fronteira de decisão.

3.2. Treinamento do Modelo

Utilizamos a densidade local do LOF para identificar pontos de dados que se desviam de seus vizinhos. Essa abordagem se baseia na densidade local, onde, para cada ponto p, a densidade é definida com base nas distâncias aos k-vizinhos mais próximos. A distância $d_k(p)$ é utilizada para determinar o raio de alcance $R(p, k)$ necessário para englobar os vizinhos. O LOF calcula a Razão de Densidade Relativa (RDR) como:

$$\text{LOF}(p) = \frac{1}{|N_k(p)|} \sum_{o \in N_k(p)} \frac{\text{densidade}(o)}{\text{densidade}(p)} \tag{1}$$

Um valor $\text{LOF}(p) > 1$ indica que p é menos denso que seus vizinhos, sugerindo uma possível anomalia. Avaliamos diferentes valores para k na faixa de 10 a 100, observando que valores entre 40 e 60 produzem melhor AUC.

Para o Isolation Forest, aplicamos a métrica de pontuação $s(p)$, baseada no comprimento médio do caminho $h(p)$ até o isolamento do ponto, conforme:

$$s(p) = 2^{-\frac{h(p)}{c(n)}} \tag{2}$$

onde $c(n)$ é uma constante de normalização baseada no tamanho da amostra. O parâmetro de contaminação foi variado de $0{,}001$ a $0{,}05$, e o número de estimadores

(`n_estimators`) entre 100 e 1000. Identificamos que `n_estimators` $= 700$ oferece um bom equilíbrio entre desempenho e custo computacional.

O OCSVM foi configurado com kernel RBF, sendo utilizados os hiperparâmetros γ e ν, que controlam, respectivamente, a influência dos pontos de suporte e a fração de anomalias esperadas. Avaliamos $\gamma \in \{0{,}01; 0{,}05; 0{,}1; 0{,}5\}$ e $\nu \in \{0{,}01; 0{,}05; 0{,}1; 0{,}2\}$. A combinação $\gamma = 0{,}1$ e $\nu = 0{,}05$ apresentou os melhores resultados.

O metamodelo Random Forest foi avaliado com diferentes números de árvores (100, 300, 500, 1000). Os melhores resultados foram obtidos com 1000 árvores, embora com maior tempo de execução. Para sistematizar a seleção dos hiperparâmetros, empregamos um procedimento de *Grid Search* com validação cruzada estratificada, utilizando o conjunto de validação extraído de 20% dos dados.

Essa análise de sensibilidade é relevante para a calibração fina dos modelos. Em especial, observamos que valores extremos de hiperparâmetros tendem a comprometer o desempenho global do *ensemble*.

3.3. Conjuntos de Dados

O conjunto de dados utilizado neste estudo é o UGR'16, composto por aproximadamente 19 bilhões de registros de tráfego de rede. Esse conjunto de dados representa um dos maiores e mais desafiadores conjuntos de dados disponíveis para a detecção de anomalias e ataques em redes. Apresenta características típicas de redes de produção, incluindo a distribuição desbalanceada entre tráfego benigno (cerca de 95%) e anomalias (aproximadamente 5%).

Focamos na análise de três classes de tráfego: *anomaly-spam*, *background* e *blacklist*, selecionadas com base em sua relevância prática em ambientes reais. A classe *anomaly-spam* representa um padrão de tráfego associado à disseminação de *spam*, frequentemente observadas em ataques automatizados. A classe *background* refere-se ao tráfego de plano de fundo não categorizado, mas que pode conter eventos atípicos difíceis de rotular sendo relevante na avaliação de modelos de detecção de anomalias de natureza genérica. Já a classe *blacklist* corresponde a interações com domínios ou IPs listados em listas negras, frequentemente usados para evasão ou exfiltração, tornando-se representativa em estratégias de detecção reativas e proativas.

Adicionalmente, o conjunto de dados CICIDS2017 [Sharafaldin et al. 2018] foi empregado para reforçar a validação do modelo proposto em amostras distintas de tráfego. Este dataset é reconhecido na literatura científica por sua abrangência e padronização, incorporando tráfego benigno e diversas classes de ataques realistas capturados em um ambiente controlado que simula uma rede corporativa. Sua estrutura balanceada entre tráfego normal e malicioso, combinada com características extraídas dos fluxos de rede, torna uma referência consolidada para validação de modelos de detecção de intrusões.

4. Resultados

Para avaliar o desempenho dos modelos, utilizamos curvas ROC por classe. Para cada classe, as taxas de verdadeiros positivos (TPR = TP / (TP + FN)) e falsos positivos (FPR = FP / (FP + TN)) são calculadas usando `roc_curve`. O AUC é computado para medir a capacidade do modelo de distinguir entre classes normais e anômalas. Também geramos

uma curva ROC agregada, considerando todas as classes e utilizando os rótulos binarizados para calcular as TPR e FPR em todo o conjunto de dados. A Figura 2 apresenta as curvas ROC para cada classe, com linhas que indicam o desempenho do modelo em termos de TPR e FPR. As curvas incluem os valores de AUC para cada classe.

O Eixo X mede a proporção de falsos positivos em relação ao total de negativos reais, indicando o custo de "alarmar incorretamente". Já o Eixo Y mede a proporção de verdadeiros positivos em relação ao total de positivos reais. A linha diagonal reflete o desempenho de um modelo aleatório (AUC = 0,5). Quanto mais próxima do canto superior esquerdo, melhor o desempenho. A AUC sintetiza a curva ROC em um único valor, onde valores próximos a 1 indicam excelente separação. A Figura 2 ilustra as curvas ROC utilizadas na comparação dos modelos. A curva ROC (a), para *anomaly-spam* (AUC = 0,99), mostra excelente separabilidade entre dados normais e tráfego de spam. A curva (b), *background* (AUC = 0,90), indica bom desempenho, embora inferior ao da classe anterior, refletindo possível sobreposição entre padrões. A curva (c), *blacklist* (AUC = 0,91), revela desempenho levemente superior ao do *background*, sendo mais eficaz na separação de domínios bloqueados e tráfego legítimo. Os resultados sugerem que a classe *anomaly-spam* é a mais bem discriminada pelo modelo, sendo a mais indicada para cenários críticos.

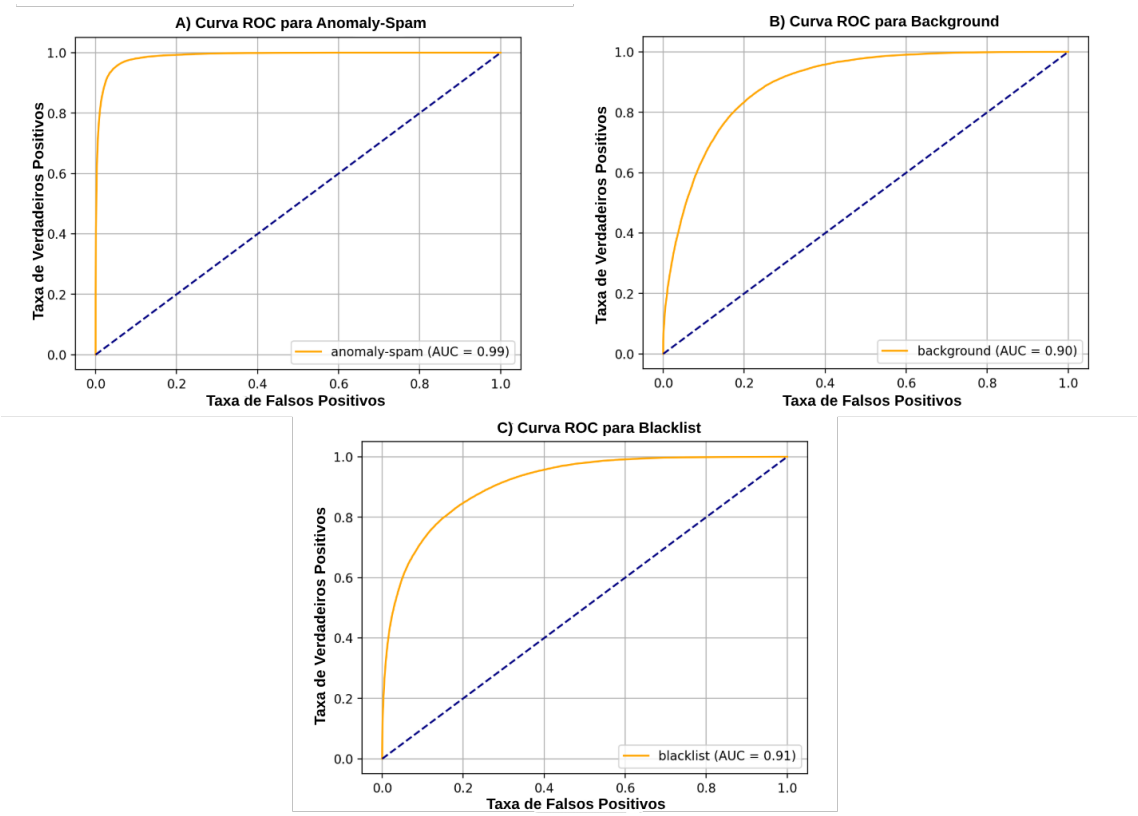

Figura 2. Comparação dos modelos individuais: (a) ROC para *anomaly-spam*, (b) *background*, (c) *blacklist*.

Comparamos o desempenho ROC para diferentes arquiteturas: *Stacking Autoencoder* (AUC = 0,97), VAE (0,96), GBT (0,91) e o modelo proposto *Random Forest Stacked* (AUC = 0,90). Embora o modelo proposto apresente AUC inferior, sua arquitetura

favorece interpretabilidade, modularidade e menor custo computacional. Testes mostraram tempo médio de inferência de 34 ms por lote de 512 instâncias, contra até 120 ms observados com *autoencoders* profundos. Também analisamos a classe 0 (tráfego benigno), com AUC de 0,90. A curva revela alta taxa de verdadeiros positivos, com baixo crescimento na taxa de falsos positivos, indicando que o modelo é eficaz também na detecção de tráfego legítimo.

Complementarmente à avaliação conduzida no UGR'16, foram realizados experimentos utilizando o dataset CICIDS2017. Utilizando o mesmo pipeline de *ensemble stacking*, os resultados obtidos revelaram melhorias nas principais métricas avaliadas. Observou-se elevação do AUC de 0,91 para 0,95, aumento da precisão de 0,89 para 0,94, da revocação de 0,90 para 0,93 e do F1-score de 0,89 para 0,935. A Tabela 1 sintetiza os resultados comparativos.

Tabela 1. Comparativo de desempenho entre os datasets UGR'16 e CICIDS2017

Métrica	**UGR'16**	**CICIDS2017**
Área sob a Curva (AUC)	0,91	0,95
Precisão	0,89	0,94
Revocação	0,90	0,93
F1-score	0,89	0,93

Conforme observamos ainda na Tabela 1, a aplicação do *pipeline* de *ensemble stacking* revelou um desempenho progressivamente superior ao longo das iterações, evidenciando a robustez do modelo proposto e sua capacidade de generalizar para domínios distintos, mantendo uma boa eficácia mesmo diante de diferentes distribuições estatísticas de tráfego. Os resultados obtidos nas execuções com o `CICIDS2017` apontam para um processo de aprendizado mais estável e com menor variância entre as métricas de avaliação, com destaque para ganhos consistentes em AUC e F1-score. Embora a maior regularidade do tráfego benigno e a qualidade da rotulagem dos ataques no `CICIDS2017` contribuam para um cenário mais controlado, os ganhos observados decorrem, em grande medida, da arquitetura técnica do modelo, cuja combinação de detectores complementares potencializa a capacidade discriminativa mesmo em contextos distintos.

5. Conclusão

Os resultados experimentais demonstraram que o modelo proposto possui uma boa capacidade de generalização, mesmo em cenários desbalanceados e de alta complexidade. As métricas obtidas destacam seu desempenho expressivo, com AUC de 0,99 para a classe *anomaly-spam*, 0,90 para *background* e 0,91 para *blacklist*. Esses valores reforçam a eficácia do modelo na discriminação de diferentes padrões de tráfego, consolidando sua aplicabilidade prática em contextos reais de segurança cibernética.

Adicionalmente, a análise por classe revela elevados índices de Precisão, Revocação e F1-score, confirmando a consistência do modelo em múltiplas categorias. Diferentemente de modelos baseados em redes profundas, como VAE e SAE, o *ensemble* proposto oferece vantagens em termos de interpretabilidade, modularidade e escalabilidade. Seu tempo médio de inferência, inferior a 35 ms por lote, demonstra sua viabilidade para aplicações em tempo quase real, com baixo custo computacional e fácil integração em *pipelines* de segurança já existentes.

Em síntese, a abordagem proposta representa um avanço significativo na detecção de anomalias em tráfego de rede, conciliando alto desempenho, baixo custo operacional e facilidade de implantação. Para trabalhos futuros, pretendemos explorar mecanismos de adaptação contínua, avaliação em ambientes com tráfego criptografado e integração com técnicas baseadas em aprendizado profundo para aprimorar ainda mais a precisão e a abrangência da solução.

Agradecimentos

Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (Proc. 162441/2021-5), e pela Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP (Proc. 2018/23098-0).

Referências

Chliah, H., Battou, A., Laoufi, A., et al. (2023). Hybrid machine learning-based approach for anomaly detection using apache spark. *International Journal of Advanced Computer Science and Applications*, 14(4).

Jeffrey, N., Tan, Q., and Villar, J. R. (2024). Using ensemble learning for anomaly detection in cyber–physical systems. *Electronics*, 13(7).

Li, J., Chen, R., and Sun, J. (2020). A stacking ensemble framework for imbalanced network anomaly detection. *Computers & Security*, 95:101847.

Lunardi, W. T., Lopez, M. A., and Giacalone, J.-P. (2022). Arcade: Adversarially regularized convolutional autoencoder for network anomaly detection. *arXiv preprint arXiv:2205.01432*.

Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P., and Therón, R. (2018). UGR'16: A new dataset for the evaluation of cyclostationarity-based network "idss". *Computers and Security*, 73:411–424.

Ness, S. (2024). Anomaly detection in network traffic using advanced machine learning techniques. *IEEE Access*, 12:1–10.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116. SciTePress.

Tokmak, M. and Nkongolo, M. (2023). Stacking an autoencoder for feature selection of zero-day threats. *arXiv preprint arXiv:2311.00304*.

Wang, Y., Li, Z., and Zhang, W. (2021). Improving IoT anomaly detection through stacking ensemble learning. *Journal of Network and Computer Applications*, 173:102854.