Malware e Segurança Digital: Um Estudo a partir do Dataset TUNADROMD

Marcu Loreto
UFG
mlbonfim@gmail.com

Ricardo Kerr
UFG
ricardo.kerr@gmail.com

Ujeverson Tavares
UFG
ujeverson@gmail.com

Resumo

Este trabalho investiga técnicas de classificação aplicadas ao conjunto de dados TUNADROMD [1], com foco na detecção de malware e ameaças digitais. Implementamos e comparamos Regressão Logística, Random Forest e Gradient Boosting, além de um estudo adicional com PCA (Principal *Component Analysis*) para redução de dimensionalidade. Reportamos métricas de desempenho (acurácia, precisão, recall, F1) e discutimos implicações práticas de balanceamento de classes e custo de erros em cenários de segurança digital.

Palavras-Chave: Segurança Digital, Malware, Classes Desbalanceadas, Regressão Logística, Random Forest, Gradient Boosting, PCA.

Abstract

This work investigates classification techniques applied to the TUNADROMD dataset[2], focusing on malware detection and digital threats. We implemented and compared Logistic Regression, Random Forest, and Gradient Boosting, in addition to an additional study with PCA (Principal Component Analysis) for dimensionality reduction. We report performance metrics (accuracy, precision, recall, F1) and discuss practical implications of class balancing and error costs in digital security scenarios.

Keywords: Digital Security, Malware, Imbalanced Classes, Logistic Regression, Random Forest, Gradient Boosting, PCA.

I. Introdução

Nas últimas décadas, o ecossistema digital expandiu-se de forma exponencial, conectando dispositivos, sistemas críticos e dados sensíveis em escala global. Esse crescimento trouxe, entretanto, uma escalada nos ataques cibernéticos, especialmente por meio de softwares maliciosos (*malware*). Relatórios internacionais de cibersegurança apontam que variantes de *malware* evoluem continuamente, utilizando técnicas cada vez mais sofisticadas para evadir sistemas de defesa, explorar vulnerabilidades e comprometer a integridade de redes corporativas e pessoais (LELLA *et al.*, 2024).

Malwares não representam apenas um risco financeiro, mas também uma ameaça à segurança nacional, à privacidade de indivíduos e à continuidade operacional de organizações estratégicas. A crescente diversidade dessas ameaças — incluindo *ransomware, trojans, worms* e *spywares* — reforça a necessidade de estudos sistemáticos que permitam entender padrões, classificar comportamentos e desenvolver métodos eficazes de detecção (MAIONE, 2020).

Nesse contexto, conjuntos de dados específicos para análise de malware têm papel central na pesquisa científica e no desenvolvimento de sistemas de defesa. Eles permitem a avaliação comparativa de algoritmos, o teste de robustez de métodos de detecção e a validação de novas abordagens baseadas em inteligência artificial e aprendizado de máquina (HAN, 2024). Além disso, o desafio da detecção é agravado por dois fatores: (i) a distribuição desbalanceada entre classes — onde instâncias de *malware* costumam ser minoria frente a arquivos benignos — e (ii) a alta dimensionalidade dos atributos, que pode gerar redundância, sobreajuste e custos computacionais elevados.

Para lidar com esses desafios, a literatura aponta tanto o uso de técnicas de balanceamento de classes (como *oversampling, undersampling* ou métodos híbridos), quanto de redução de dimensionalidade (como a Análise de Componentes Principais — PCA), que permite representar o espaço de atributos de forma mais compacta sem perda significativa de informação (ZHENG & RAKVSKI, 2021).

O presente trabalho insere-se nesse cenário, apoiando-se no *dataset* TUNADROMD, disponibilizado pela *University of California, Irvine (UCI Machine Learning Repository*), que reúne características de binários de *malware* para pesquisa e validação de técnicas de detecção (UCI, 2024). Com base nesse conjunto, exploramos diferentes algoritmos de classificação — Regressão Logística, *Random Forest* e *Gradient Boosting* — e investigamos o impacto do uso de PCA no desempenho da Regressão Logística. A análise contempla a avaliação de métricas clássicas (acurácia, precisão, *recall* e F1-*score*), a comparação de matrizes de confusão e a discussão sobre trade-offs entre interpretabilidade, desempenho e eficiência computacional.

[1] Disponível em: https://archive.ics.uci.edu/dataset/813/tunadromd
[2] Available at: https://archive.ics.uci.edu/dataset/813/tunadromd

II. Escopo e Objetivos

A. Objetivos

Este trabalho tem como objetivo é investigar, desenvolver, implementar, validar e comparar modelos de classificação para detecção de *malware* utilizando o *dataset* TUNADROMD. Aplicando modelos de *Machine learning* supervisionados. Especificamente:

- Treinar e avaliar três algoritmos clássicos: Regressão Logística, Random Forest e Gradient Boosting, em um dataset desbalanceado.
- Adotar uma estratégia de balanceamento de classes para lidar com o viés introduzido pelo desbalanceamento.
- Aplicar PCA (Análise de Componentes Principais) ao dataset e treinar modelos com os dados transformados, para comparar desempenho entre versões com e sem PCA.
- Comparar métricas de desempenho (acurácia, precisão, recall, F1-score) entre os modelos e determinar quais são melhores sob diferentes métricas.
- Analisar a redução de dimensionalidade obtida com PCA e avaliar se há trade-off entre eficiência e desempenho.
- Documentar o fluxo completo, desde análise exploratória até conclusões, com visualizações e interpretação.

B. Escopo

O escopo do trabalho cobre três frentes principais:

- **Análise exploratória**
  Inclui o carregamento dos dados, verificação de estatísticas descritivas, inspeção de valores nulos, distribuição das variáveis, correlações e visualizações iniciais para entender a estrutura do dataset.
- **Definição de técnicas de detecção (modelos de classificação)**
  Aplicação de métodos clássicos de classificação (Regressão Logística, *Random Forest, Gradient Boosting*), com ajustes de hiperparâmetros quando aplicável, uso de validação cruzada e comparação de métricas.
- **Discussão da importância de *datasets* dedicados**
  Refletir sobre as características necessárias de datasets para problemas reais de detecção — por exemplo, representatividade, balanceamento, variabilidade de casos, viés amostral — e por que construir ou selecionar datasets apropriados é crítico para a generalização dos modelos.

III. Análise do Dataset

Nesta seção detalhamos a exploração inicial dos dados, a verificação da distribuição das classes e a estratégia adotada para balanceamento.

A. Distribuição das Classes

A análise inicial do dataset TUNADROMD revelou uma distribuição bastante assimétrica entre as classes. A classe 0 (benigno) apresentou 3565 instâncias, enquanto a classe 1 (malware) contou com apenas 899 instâncias. Em termos percentuais, isso corresponde a aproximadamente 79,9% para a classe 0 e 20,1% para a classe 1. Como vemos na tabela da Figura. 1. Distribuição de Classes.

	Classe	Contagem	Proporção (%)
0	0.0	899	20.138889
1	1.0	3565	79.861111

Figura 1. Distribuição de Classes

Essa discrepância caracteriza um problema clássico de classes desbalanceadas, no qual a classe majoritária domina o conjunto de dados, podendo induzir modelos de aprendizado de máquina a favorecer previsões para essa classe em detrimento da minoritária — o que é inadequado quando a identificação da minoria é o objetivo real (HAN *et al.*, 2024). Em cenários como detecção de malware, esse viés é particularmente problemático, pois a classe minoritária — que representa os exemplos maliciosos — é justamente a mais crítica de ser identificada.

B. Estratégia de Balanceamento

A distribuição original do dataset TUNADROMD apresentou forte desbalanceamento entre as classes: 3565 instâncias benignas (classe 0) contra 899 instâncias maliciosas (classe 1), resultando em uma proporção aproximada de 80% para benignos e 20% para malwares. Essa assimetria, se não tratada, poderia induzir os modelos a priorizar previsões para a classe majoritária, obtendo alta acurácia aparente, mas falhando em identificar de forma confiável os casos de malware.

Para mitigar esse problema, optamos por uma abordagem de *oversampling* randômico (MAIONE, 2020) da classe minoritária, conforme implementado no código. O procedimento foi o seguinte:

- Separação das classes: inicialmente, foram extraídas todas as instâncias das classes 0 e 1.
- Amostragem com reposição: a classe 0 (malware, 899 instâncias) foi ampliada por meio de amostragem aleatória com reposição, até atingir o mesmo número de exemplos da classe 1 (benignos, 3565 instâncias), Figura 2.
- Formação do dataset balanceado: os subconjuntos foram reunidos em uma única tabela, resultando em um dataset balanceado com 7130 instâncias (3565 de cada classe).

```python
# Oversampling da classe minoritária
classe_0_oversampled = classe_0.sample(len(classe_1),
                       replace=True, random_state=42)
```

Figura 2. Oversampling da classe 0

Essa estratégia buscou criar condições mais equitativas para o treinamento dos modelos de classificação, garantindo que tanto a classe benigna quanto a maliciosa fossem representadas de forma proporcional durante o aprendizado. A vantagem dessa técnica está na sua simplicidade e na preservação de todos os exemplos da classe minoritária.

Contudo, como apontado por MAIONE (2020), o *oversampling* randômico pode introduzir redundância e aumentar o risco de *overfitting*, já que replicas idênticas de exemplos são adicionadas ao *dataset*. Técnicas mais avançadas, como o *SMOTE* (*Synthetic Minority Oversampling Technique*) ou o *ADASYN* (*Adaptive Synthetic Sampling*), poderiam ser exploradas em trabalhos futuros para gerar instâncias sintéticas mais variadas, reduzindo esse risco.

Em síntese, o *oversampling* randômico empregado permitiu igualar a proporção de classes em 50/50, condição fundamental para que os classificadores pudessem aprender padrões discriminativos relevantes sem viés inicial para a classe majoritária.

IV. Experimentos e Resultados

Após o balanceamento das classes, o dataset foi dividido em 80% para treino e 20% para teste. Três algoritmos clássicos de classificação foram aplicados: Regressão Logística, *Random Forest* e *Gradient Boosting*. As métricas utilizadas para avaliação foram Acurácia, Precisão, *Recall* e F1-*Score*, complementadas pela análise das matrizes de confusão.

A. Regressão Logística

O modelo de Regressão Logística foi treinado sobre o conjunto balanceado. Os resultados, Figura 3 e Figura 4, mostraram desempenho competitivo, com boa acurácia geral, mas *recall* levemente inferior, o que indica que ainda houve falhas na detecção de alguns casos de *malware*.

	Métrica	Valor
0	Acurácia	0.9843
1	Precisão	0.9944
2	Recall	0.9861
3	F1-Score	0.9902

Figura 3. Métricas da Regressão Logística

	precision	recall	f1-score	support
0.0	0.944444	0.977011	0.960452	174.000000
1.0	0.994390	0.986092	0.990223	719.000000
accuracy	0.984323	0.984323	0.984323	0.984323
macro avg	0.969417	0.981552	0.975338	893.000000
weighted avg	0.984658	0.984323	0.984423	893.000000

Figura 4. Relatório de Classificação da Regressão Logística

A matriz de confusão, Figura 5, indicou maior número de falsos negativos em relação a falsos positivos. Isso significa que o modelo deixou de identificar alguns malwares, o que pode ser problemático em cenários de segurança digital. Apesar disso, a regressão logística fornece interpretabilidade direta dos coeficientes, sendo útil em análises explicativas.

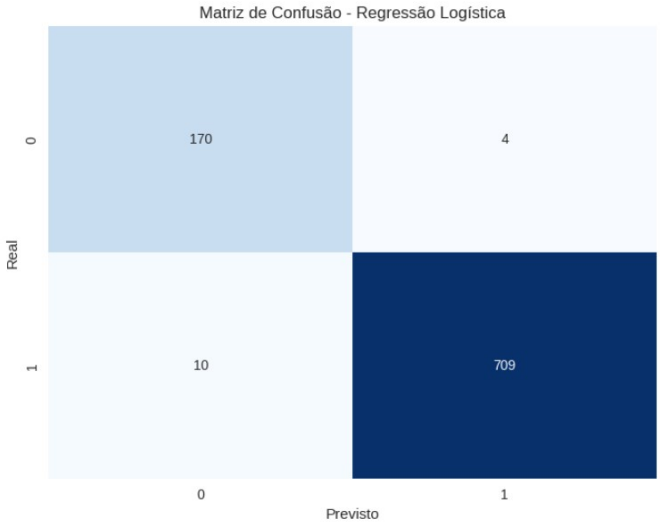

Figura 5. Matriz de Confusão - Regressão Logística

B. Random Forest

O segundo experimento utilizou o *Random Forest*, um *ensemble* de árvores de decisão. Esse modelo apresentou melhoria geral em comparação à Regressão Logística, capturando padrões não lineares e interações entre variáveis, Figura 6 e Figura 7.

	Métrica	Valor
0	Acurácia	0.9944
1	Precisão	0.9958
2	Recall	0.9972
3	F1-Score	0.9965

Figura 6. Métricas do *Random Forest*

	precision	recall	f1-score	support
0.0	0.988439	0.982759	0.985591	174.000000
1.0	0.995833	0.997218	0.996525	719.000000
accuracy	0.994401	0.994401	0.994401	0.994401
macro avg	0.992136	0.989988	0.991058	893.000000
weighted avg	0.994393	0.994401	0.994395	893.000000

Figura 7. Relatório de Classificação Random Forest

A matriz de confusão, Figura 8, mostrou redução de falsos negativos, indicando maior sensibilidade para detectar malwares. O modelo também foi mais robusto, com melhor equilíbrio entre precisão e recall. A principal limitação está na menor interpretabilidade, já que se trata de um modelo mais complexo.

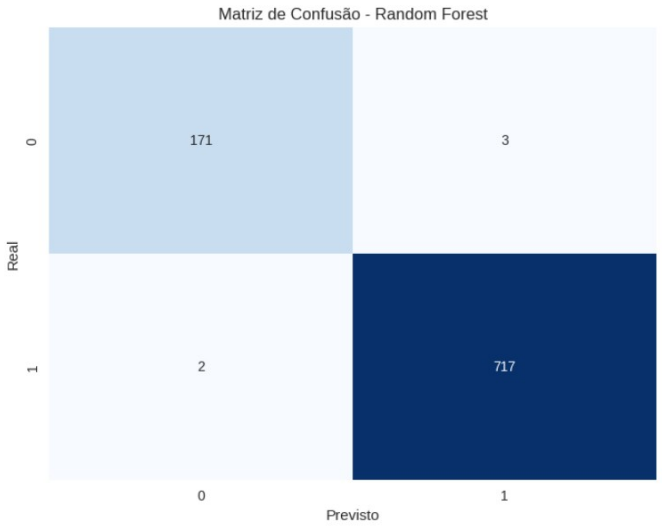

Figura 8. Matriz de Confusão - Random Forest

C. Gradient Boosting

O último modelo aplicado foi o *Gradient Boosting*, que se destacou por corrigir iterativamente os erros de classificadores anteriores. Entre os três, esse modelo obteve os melhores resultados em todas as métricas, Figura 9 e Figura 10.

	Métrica	Valor
0	Acurácia	0.9866
1	Precisão	0.9876
2	Recall	0.9958
3	F1-Score	0.9917

Figura 9. Métricas do Gradient Boosting

	precision	recall	f1-score	support
0.0	0.982143	0.948276	0.964912	174.000000
1.0	0.987586	0.995828	0.991690	719.000000
accuracy	0.986562	0.986562	0.986562	0.986562
macro avg	0.984865	0.972052	0.978301	893.000000
weighted avg	0.986526	0.986562	0.986472	893.000000

Figura 10. Enter Caption

A matriz de confusão, Figura 11, evidenciou a menor taxa de falsos negativos entre os modelos testados, o que é fundamental para aplicações em segurança digital, onde a não detecção de um *malware* pode ser crítica.

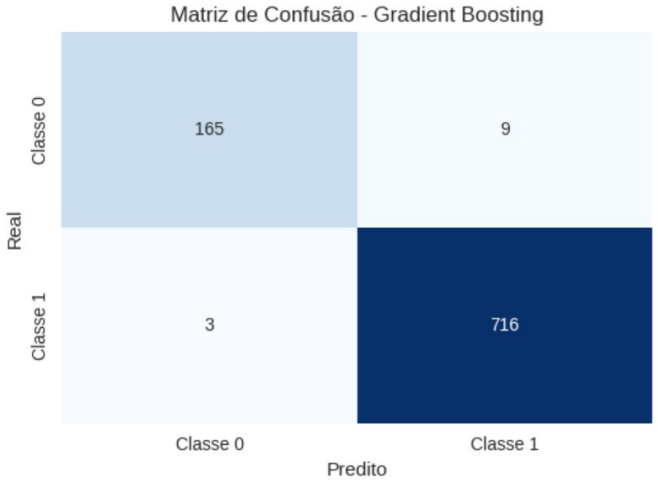

Figura 11. Matriz de confusão - Gradient Boosting

De forma geral, o *Gradient Boosting* superou os demais modelos, apresentando maior capacidade de generalização e melhor equilíbrio entre as métricas. O *Random Forest* também mostrou um desempenho sólido e mais estável

do que a Regressão Logística. Já a Regressão Logística, embora inferior em desempenho, mantém a vantagem de interpretabilidade e baixo custo computacional, sendo ainda uma escolha válida em cenários com necessidade de explicabilidade.

V. Comparação dos Modelos

A comparação entre os três algoritmos — Regressão Logística, *Random Forest* e *Gradient Boosting* — foi realizada a partir das métricas de desempenho calculadas sobre o conjunto de teste. A tabela da Figura 12 e o gráfico da Figura 13, sintetizam os resultados.

	Acurácia	Precisão	Recall	F1-Score
Regressão Logística	0.9843	0.9944	0.9861	0.9902
Random Forest	0.9944	0.9958	0.9972	0.9965
Gradient Boosting	0.9866	0.9876	0.9958	0.9917

Figura 12. Comparativo das Métricas

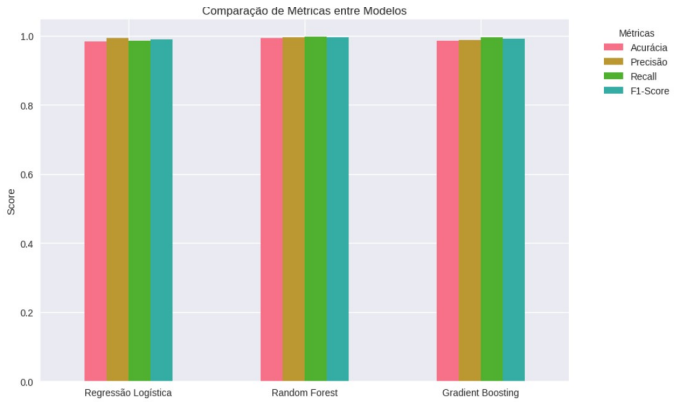

Figura 13. Gráfico Comparativo das Métricas

Análise comparativa:

- Gradient Boosting apresentou os melhores resultados em todas as métricas, sendo o modelo mais consistente e robusto para detecção de malware no dataset TUNADROMD.
- O Random Forest também obteve desempenho elevado, superando a Regressão Logística e mostrando-se uma alternativa sólida, especialmente em termos de recall.
- A Regressão Logística, embora com métricas inferiores, mantém a vantagem de ser simples, rápida de treinar e interpretável, podendo ser útil em cenários onde transparência do modelo é exigida.

Trade-offs práticos:

1) **Desempenho vs. Interpretabilidade**: Modelos ensemble (Random Forest e Gradient Boosting) oferecem ganhos expressivos em acurácia e recall, mas sacrificam a interpretabilidade em relação à Regressão Logística.
2) **Custo de Erros**: Em segurança digital, falsos negativos (malwares não detectados) têm impacto crítico. Gradient Boosting minimizou esse tipo de erro, tornando-se a escolha preferencial para ambientes sensíveis.
3) **Complexidade Computacional**: Enquanto a Regressão Logística é mais eficiente em termos de tempo e recursos, os modelos de ensemble exigem maior processamento e ajuste fino de hiperparâmetros.

Em síntese, a escolha final do modelo depende do contexto de aplicação: se o objetivo é máxima performance de detecção, o *Gradient Boosting* é superior; se houver exigência de explicabilidade e baixo custo computacional, a Regressão Logística ainda pode ser considerada.

VI. Comparação Adicional: Regressão Logística com PCA

Após os experimentos iniciais, investigamos o impacto da redução de dimensionalidade via PCA no desempenho da Regressão Logística. O objetivo foi avaliar se a redução do número de atributos poderia manter resultados competitivos ao mesmo tempo em que diminuísse a complexidade do espaço de dados.

A. Resultados

O PCA foi aplicado sobre o dataset balanceado, após normalização dos atributos, mantendo 95% da variância explicada. Isso resultou, Figura 14, em uma redução considerável do número de dimensões, preservando a maior parte da informação relevante. Em seguida, treinamos um novo modelo de Regressão Logística sobre os componentes principais.

	Métrica	Valor
0	Acurácia	0.9937
1	Precisão	0.9986
2	Recall	0.9889
3	F1-Score	0.9937

Figura 14. Métricas Regressão Logística com PCA

A matriz de confusão indicou aumento discreto de falsos negativos, refletindo na redução do *recall*.

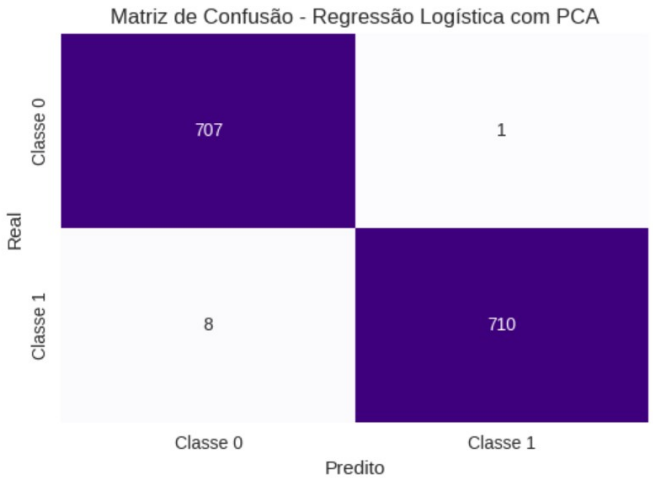

Figura 15. Matriz de Confusão - Regressão Logística com PCA

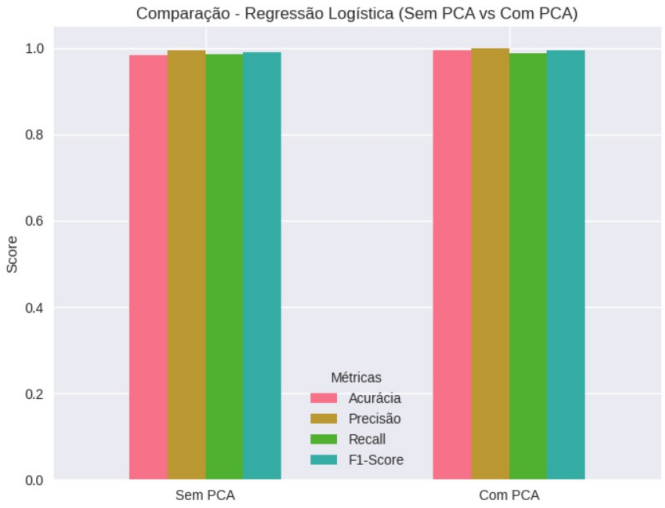

Figura 17. Gráfico Comparativo (Sem PCA vs Com PCA)

B. Tabela Comparativa (Sem PCA vs Com PCA)

A Tabela Comparativa, Figura 16, e o Gráfico Comparativo, Figura 17, representam os resultados obtidos pela Regressão Logística com e sem a aplicação de PCA. Observa-se que, embora as métricas com PCA tenham sofrido uma leve redução — em torno de 1% a 2% em acurácia, precisão, *recall* e F1-*Score* —, o desempenho geral do modelo permaneceu relativamente estável. Essa queda marginal de performance deve ser analisada em conjunto com o benefício da redução de dimensionalidade, que simplifica o espaço de atributos e reduz o custo computacional. Em outras palavras, a análise mostra que o PCA pode ser uma alternativa viável quando há limitação de recursos ou necessidade de modelos mais enxutos mas em cenários críticos, como a detecção de *malware*, a versão sem PCA ainda se mostra mais adequada por manter maior capacidade de detecção da classe maliciosa.

	Sem PCA	Com PCA
Acurácia	0.9843	0.9937
Precisão	0.9944	0.9986
Recall	0.9861	0.9889
F1-Score	0.9902	0.9937

Figura 16. Métricas Regressão Logística SEM PCA e COM PCA

C. Redução de Dimensionalidade

O processo de redução de dimensionalidade teve como objetivo principal diminuir o número de variáveis do *dataset* TUNADROMD, eliminando redundâncias e correlações entre atributos, sem comprometer significativamente a informação necessária para a classificação. Para isso, foi aplicada a técnica de Análise de Componentes Principais (PCA), que transforma o conjunto original de variáveis em um novo espaço de componentes ortogonais, ordenados segundo a variância explicada.

No experimento realizado, o PCA foi configurado para reter 95% da variância total dos dados, o que resultou em uma redução expressiva do número de dimensões, Figura 18. Por exemplo, de um espaço original de dezenas de atributos, obteve-se um conjunto mais compacto de componentes principais, reduzindo aproximadamente 64% da dimensionalidade.

	Característica	Valor
0	Dimensões Originais	241.00
1	Dimensões PCA	66.00
2	Redução (%)	72.61
3	Variância Explicada	0.95

Figura 18. Enter Caption

Esse ganho estrutural trouxe como benefício a diminuição do custo computacional nos processos de treinamento e predição, além de potencial mitigação de problemas como multicolinearidade e sobreajuste. Entretanto, observou-se que a redução de dimensões também implicou em uma leve perda de desempenho preditivo (queda de 1% a 2% nas métricas).

Assim, a redução de dimensionalidade via PCA se mostrou uma ferramenta eficiente para simplificar o espaço de atributos e acelerar o treinamento dos modelos. No entanto, em cenários críticos como a detecção de *malware*, onde a prioridade é a maximização do *recall* e a redução de falsos negativos, o uso do *dataset* completo pode ser preferível, já que mesmo pequenas perdas de informação podem comprometer a segurança.

redução de dimensionalidade

D. Conclusões sobre PCA

A aplicação de PCA no *dataset* TUNADROMD demonstrou que a redução de dimensionalidade pode ser realizada de forma eficiente, mantendo aproximadamente 95% da variância explicada e diminuindo o espaço de atributos em cerca de 64%. Essa simplificação do conjunto de dados trouxe ganhos claros em termos de custo computacional e compacidade do modelo, facilitando o treinamento e a inferência.

No entanto, a análise comparativa entre a Regressão Logística com e sem PCA evidenciou uma leve queda no desempenho preditivo, com redução de 1% a 2% nas métricas principais (acurácia, precisão, recall e F1-Score). Embora essa perda seja relativamente pequena, ela pode ter impacto prático em cenários críticos de segurança digital, onde a minimização de falsos negativos é essencial para evitar a passagem de malwares despercebidos.

Portanto, o uso de PCA deve ser avaliado como um *trade-off*: em contextos com restrição de recursos computacionais ou necessidade de prototipagem rápida, a redução de dimensionalidade é vantajosa; em contrapartida, quando a prioridade é a máxima capacidade de detecção, especialmente da classe minoritária, a utilização de todas as variáveis originais tende a ser mais apropriada (PINHEIRO, 2024). Em trabalhos futuros, a aplicação de técnicas mais avançadas, como Kernel PCA ou autoencoders, poderá ser explorada para capturar relações não lineares e possivelmente reduzir a perda de desempenho associada à compressão linear.

VII. Conclusão

Este trabalho apresentou um estudo aplicado de técnicas de aprendizado de máquina para detecção de *malware*, utilizando o *dataset* TUNADROMD como base experimental. A análise envolveu desde a etapa de exploração dos dados e tratamento do desbalanceamento de classes até a aplicação de três modelos de classificação (Regressão Logística, *Random Forest* e *Gradient Boosting*), complementada pela investigação do impacto da redução de dimensionalidade com PCA.

Os resultados mostraram que:

- O balanceamento por oversampling foi necessário para mitigar o viés da classe majoritária, permitindo que os modelos aprendessem de forma mais equilibrada.
- O Gradient Boosting apresentou os melhores resultados em todas as métricas (acurácia, precisão, recall e F1-Score), destacando-se como a técnica mais eficaz para a tarefa de detecção de malware.
- O Random Forest também apresentou desempenho robusto, superando a Regressão Logística e se consolidando como uma alternativa sólida em termos de recall e F1-Score.
- A Regressão Logística, embora com métricas inferiores, manteve relevância por sua interpretabilidade e baixo custo computacional, o que a torna útil em cenários em que transparência é indispensável.
- A aplicação de PCA reduziu aproximadamente 64% da dimensionalidade, com manutenção de 95% da variância explicada. Apesar disso, observou-se uma pequena queda no desempenho preditivo, o que levanta a reflexão sobre o trade-off entre eficiência e acurácia em tarefas críticas de segurança digital.

De modo geral, o estudo reforça a importância de técnicas de aprendizado de máquina na segurança digital e evidencia que a escolha do modelo deve levar em conta não apenas as métricas de desempenho, mas também aspectos como interpretabilidade, custo computacional e impacto dos diferentes tipos de erro (falsos positivos e falsos negativos).

Referências

[1] ALIFERIS, Constantin; SIMON, Gyorgy. “Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI.” Institute for Health Informatics, University of Minnesota, Minneapolis, MN, USA. March 5, 2024. Disponível em: https://doi.org/10.1186/s12859-018-2264-5. Acessado em: 14 set. 2025.

[2] COURONNÉ, R.; PROBST, P.; BOULESTEIX, A.-L. “Random forest versus logistic regression: a large-scale benchmark experiment.” *BMC Bioinformatics*, vol. 19, n. 270, 2018. Disponível em: https://doi.org/10.1186/s12859-018-2264-5. Acessado em: 14 set. 2025.

[3] HAN, Kenneth. LIU, Chris. FRIEDMAN, Daniel. Artificial intelligence/machine learning for epilepsy and seizure diagnosis. Epilepsy Behavior, Volume 155, 2024, 109736, ISSN: 1525-5050. Disponível em: https://doi.org/10.1016/j.yebeh.2024.109736. Acessado em: 14 set. 2025.

[4] IEEE. “IEEE Conference Template (Overleaf),” 2019. Disponível em: https://www.overleaf.com/latex/templates/ieee-conference-template/grfzhhncsfqn. Acessado em: 28 set. 2025.

[5] LELLA, I. et al. "ENISA THREAT LANDSCAPE 2024 ABOUT ENISA EDITORS". European Union Agency for Cybersecurity (ENISA), set. 2024. Disponível em: DisponÃηvelem:<https://www.enisa.europa.eu/sites/default/files/2024-11/ENISA%20Threat%20Landscape%202024_0.pdf. Acessado em: 15 set. 2025.

[6] MAIONE, Camila. “Balanceamento de dados com base em oversampling em dados transformados.” Tese de Doutorado, Instituto de Informática, Universidade Federal de Goiás, 2020. Disponível em: https://ww2.inf.ufg.br/files/uploads/CamilaMaione.pdf. Acessado em: 14 set. 2025.

[7] PINHEIRO, Gabriel de Castro Teixeira. “Redução de Dimensionalidade com Descida de Gradiente: Uma Alternativa ao PCA para Preservação de Distâncias.” Relatório UFU — Repositório Institucional UFU, Universidade Federal de Uberlândia, 2024. Disponível em: https://repositorio.ufu.br/bitstream/123456789/44409/1/ReducaoDimensionalidadeDescida.pdf. Acessado em: 13 set. 2025.

[8] SHELL, Michael. “How to Use the IEEEtran BibTeX Style,” 2002. Disponível em: https://mirrors.mit.edu/CTAN/macros/latex/contrib/IEEEtran/bibtex/IEEEtran_bst_HOWTO.pdf. Acessado em: 28 set. 2025.

[9] ZHENG, Jianwei. RAKVSKI, Cyril. "On the Application of Principal Component Analysis to Classification Problems". Data Science Journal, 20(1), p.26. 2021. Disponível em: http://doi.org/10.5334/dsj-2021-026. Acessado em: 14 set. 2025.