Comparação de Modelos Classificadores na Detecção de Anomalias em Sistemas de Detecção de Instrusão

Lázaro Cunha
Departamento de Ciência e Tecnologia
Universidade Federal de São Paulo
São José dos Campos, Brasil
robert.silva10@unifesp.br

I. Introdução e Motivação

Nos dias atuais, a segurança da informação é uma preocupação primordial para organizações de todos os portes e setores. Com o aumento exponencial da quantidade de dados trocados e armazenados digitalmente, as ameaças cibernéticas têm se tornado cada vez mais sofisticadas e frequentes podendo causar prejuízos milionários para empresas [1]. Sistemas de Detecção de Intrusão (*Intrusion Detection Systems* - IDS) desempenham um papel crucial na proteção de redes e sistemas computacionais contra acessos não autorizados, ataques maliciosos e outras atividades anômalas que podem comprometer a integridade, confidencialidade e disponibilidade dos dados.

Entre as diversas abordagens empregadas em IDS, os algoritmos de classificação têm se destacado como ferramentas essenciais para distinguir entre atividades normais e anômalas. Esses algoritmos utilizam técnicas de aprendizado de máquina para analisar padrões de dados e identificar potenciais ameaças. Contudo, a escolha do algoritmo mais adequado para uma determinada aplicação pode ser desafiadora, dada a variedade de métodos disponíveis e suas especificidades.

Nesse sentido, a escolha correta de um algoritmo classificador é fundamental para o desempenho de um IDS, pois um modelo inadequado pode resultar em altas taxas de falsos positivos ou falsos negativos, comprometendo a segurança e a eficiência do sistema. Por isso, é imperativo realizar uma avaliação comparativa dos diferentes algoritmos classificadores disponíveis, a fim de identificar quais oferecem a melhor performance na detecção de anomalias em cenários específicos.

II. Fundamentos

A. Sistema de Detecção de Intrusão

Um sistema de detecção de intrusão (IDS, na sigla em inglês) é uma solução de segurança que monitora a rede ou sistemas de computadores para atividades maliciosas ou violações de políticas [2]. Esses sistemas são projetados para identificar possíveis ameaças e alertar os administradores de rede sobre atividades suspeitas, permitindo uma resposta rápida para proteger os ativos digitais.

B. Tipos de Sistemas de Detecção de Intrusão

Existem dois tipos principais de IDS. O primeiro, Sistema de Detecção de Intrusão Baseado em Rede (NIDS - *Network-based Intrusion Detection System*): monitora o tráfego de rede em busca de atividades suspeitas. Esses sistemas são colocados em pontos estratégicos dentro da rede para monitorar o tráfego que se move para todos os dispositivos na rede. O segundo, Sistema de Detecção de Intrusão Baseado em Host (HIDS - *Host-based Intrusion Detection System*): monitora as atividades em um único host ou dispositivo, como um servidor ou computador individual. O HIDS analisa os logs de eventos e outras informações de um dispositivo específico para detectar comportamentos anômalos.

C. Formas de Operação de Um Sistema de Detecção de Intrusão

Os IDS podem operar de duas formas principais. Primeiramente, Detecção Baseada em Assinaturas: identifica intrusões com base em padrões conhecidos de ataques (assinaturas), semelhantes ao funcionamento de um antivírus. Esse método é eficaz contra ameaças conhecidas, mas pode não detectar ataques novos ou modificados. Em segundo lugar, Detecção Baseada em Anomalias: monitora a atividade do sistema e sinaliza qualquer comportamento que se desvie do padrão normal (anômalo). Este método pode detectar novos tipos de ataques, mas também pode gerar falsos positivos se o comportamento normal não for bem definido.

Alguns sistemas de detecção de intrusão avançados combinam ambos os métodos para aumentar a precisão e a eficácia na detecção de ameaças. Além disso, é importante notar que um IDS é principalmente uma ferramenta de monitoramento e alerta; ele não bloqueia automaticamente as intrusões (essa função é geralmente desempenhada por um Sistema de Prevenção de Intrusão - IPS, *Intrusion Prevention System*).

Em resumo, um IDS é uma ferramenta crucial para a segurança cibernética, fornecendo uma camada adicional de defesa ao identificar atividades suspeitas e permitindo que os administradores tomem medidas para proteger suas redes e sistemas.

D. Modelos de Classificação

Um modelo de classificação é uma ferramenta utilizada em aprendizado de máquina e inteligência artificial para categorizar dados em classes predefinidas. Esse tipo de modelo é treinado utilizando um conjunto de dados rotulados, onde cada exemplo de treinamento é composto por um conjunto de características (ou atributos) e uma etiqueta de classe correspondente. A função principal de um modelo de classificação é aprender a relação entre as características dos dados e suas respectivas classes, de forma que, ao receber novos dados não rotulados, ele possa prever corretamente a classe a que esses dados pertencem.

Existem diversos modelos de classificação, algunns são modelos menos robustos como Árvores de Decisão, Naïve Bayes etc. e modelos mais robustos como *Support Vector Machines* (SVM), modelos *emsemble*, esses unem diversos outros modelos e suas tomadas de decisão para tomar uma decisão mais robusta, modelos baseados em redes neurais artificiais e também modelos de classificação hierárquica que levarão em consideração a hierarquia do exemplo para classificá-lo.

E. Métricas de Avaliação

As métricas utilizadas para comparação dos modelos são acurácia, precisão, recall e f1-score que são calculadas da seguinte forma:

Acurácia,

$$\frac{VP + VN}{VP + VN + FP + FN}$$

Precisão,

$$\frac{VP}{VP + FP}$$

Recall,

$$\frac{VP}{VP + FN}$$

F1-score,

$$2 * \frac{Precisão * Recall}{Precisão + Recall}$$

em que VP são verdadeiros positivos, VN são verdadeiros negativos, FP são falsos positivos e FV são falsos verdadeiros. Essas medidas podem ser resumidas em: acurácia é uma taxa de acerto geral, precisão é a proporcão de acertos de uma classe, recall é a quantidade de exemplos de uma classe que o modelo conseguiu identificar e f1-score uma medida de relação entre precisão e recall. Os valores dessas métricas vão de 0 até 1, sendo 1 um valor ótimo para todas.

F. Técnicas de Validação

A técnica de validação usada foi validação cruzada em que o modelo de classificação é treinado e testado em diferentes repartições do mesmo conjunto de dados por várias vezes consecutivas para avaliar a boa generalização do modelo e se podem existir possíveis deficiências no modelo treinado em razão da variação do conjunto de dados.

III. Trabalhos Relacionados

O artigo *Anomaly based Network Intrusion Detection using Machine Learning Techniques* [3] aborda a detecção de intrusões em redes utilizando técnicas de aprendizado de máquina. Os autores propõem um modelo de classificação baseado nos algoritmos *Random Forest* e *Support Vector Machines* (SVM). Eles utilizaram o dataset NSL-KDD, uma versão aprimorada do KDD Cup 99, que contém 41 características para descrever padrões de tráfego de rede. A metodologia envolve a classificação do tráfego de rede como normal ou ataque.

Os resultados mostram que o algoritmo *Random Forest* superou o algoritmo de *Support Vector Machines* (SVM) em termos de precisão e taxa de detecção. A precisão do Random Forest foi de 99,86%, enquanto a do SVM foi de 95,53%. A análise dos resultados indica que o *Random Forest* é mais eficaz na detecção de intrusões considerando o conjunto de dados utilizado.

Já o artigo Análise do Desempenho de Algoritmos Classificadores para Detecção de Anomalias em Sistemas de Detecção de Intrusão [4] aborda o mesmo problema, no entanto, o autor utiliza um conjunto de dados diferente, o conjunto UNSW-NB15. Ademais, utiliza, além de *Support Vector Machines* (SVM), os algoritmos *Naïve Bayes* e *Adaptive Boosting* com *Naïve Bayes* como estimador base. Nesse contexto, o artigo demonstra que todos os três algoritmos obtiveram bons desempenhos, no entanto os algortimos *Adaptive Boosting* e *Naïve Bayes* obtiveram acurácia, *Recall*, *f1-score* e AUC ROC melhores que o SVM que foi melhor apenas na métrica precisão.

Considerando a literatura supracitada, podemos observar que os modelos tradicionais tem bons desempenhos, no entanto, uma pequena taxa de erro no modelo pode levar a grandes brechas em razão da grande quantidade de dados que podem passar pelas redes. Assim, continua aberto o problema de achar um modelo ideal para a solução do problema de detecção de anomalias em sistemas de detecção de intrusão.

IV. Objetivo

Este trabalho tem como objetivo comparar diferentes modelos de classificação *emsemble* como: *Adaptive Boosting, Extreme Gradient Boosting, Bagging, Random Forest*, o modelo baseado em redes neurais *Multilayer Perceptron* e modelos de classificação hierárquica como classificadores locais por nó, por nó pai e por nível na detecção de anomalias em sistemas de detecção de intrusão. Através de uma análise comparativa detalhada, busca-se identificar quais métodos são mais eficazes através de diversas métricas de avaliação como acurácia, precisão, recall e f1-score.

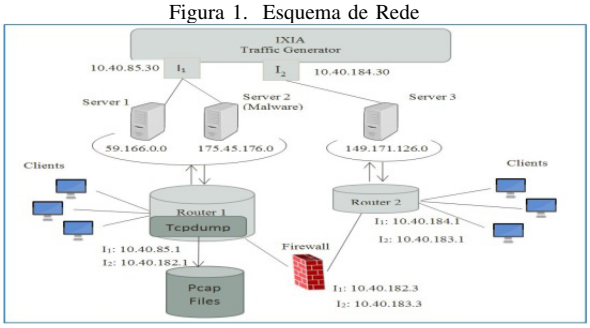
V. Metodologia Experimental

Primeiramente, as ferramentas utilizadas serão os *Notebooks Jupyter* aliados as bibliotecas da linguagem de programação *Python* como *Pandas*, *Numpy*, *Matplotlib*, *Scikit-Learn* e *HiClass* [7]. Além disso, a base de dados a ser utilizada é a

UNSW-NB15 [5], [6], base experimental com aproximadamente 2 milhões e 500 mil exemplos e que tem 47 características, 2 classes de estado ataque e normal e 9 categorias de ataques, esses sendo *Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode* e *Worms*.

A figura 1 nos mostra como os dados da rede foram obtidos [5], [6] através da geração de tráfego normal e anômalo com a ferramenta IXIA. A rede consiste em três servidores, os servidores 1 e 3 espalham tráfego normal enquanto o servidor 2 espalha tráfego anormal. Os dados de tráfego são coletados em formato .pcap pela ferramenta Tcpdump.

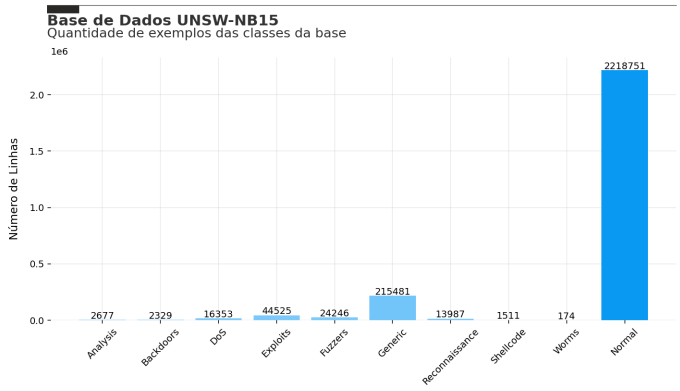
Figura 1. Esquema de Rede



Fonte: Moustafa e Slay

A figura seguinte retrata a quantidade de exemplos na base para cada um dos nove tipos de ataques e o estado de normalidade.

Figura 2. Distribuição de Estados da Base



Fonte: Autor

A. Pré-processamento dos Dados

Para o pré-processamento dos dados foram utilizadas principalmente as bibliotecas *Pandas*, *Numpy* e *Scikit-Learn*.

Primeiramente, foi feita a limpeza dos dados, através da retificação de algumas inconsistências como de tipos de dados, dados faltantes e inconsistências de multiplos nomes para um mesmo tipo de categoria. Após isso, foi feita a codificação de dados não numéricos para tipos nummericos para aceitação dos modelos de aprendizagem através de codificação discreta. Além disso, foi feita a normalização dos dados para que os modelos de aprendizagem tivessem melhor eficiência no momento de treinamento.

B. Análise Exploratória dos Dados

Para a análise exploratória dos dados foi utilizada a biblioteca *Sweetviz* para analisar distribuições e métricas descritivas como média, moda, mediana, variância, desvio padrão, mínimos, máximos e quantidade de elementos distintos. Nesse sentido, foi visualizada o grande desbalanceamento da base em relação a estados de normalidade e ataque da rede e também do grande desbalanceamento dos tipos de ataques disponíveis na base que vão influenciar os modelos de classificação multiclasse.

C. Seleção de Características

Para a etapa de seleção de características foi usada principalmente a biblioteca *Scikit-Learn* com o método *GridSearchCV* com o modelo de classificação *Random Forest* binário para encontrar os melhores parâmetros para o modelo e, assim, fazer um processo através de validação cruzada de encontrar a média das importâncias das características do modelo *Random Forest* por meio de vários treinamentos em diferentes separações dos dados para um melhor conhecimento do valor da importância das características. Para os modelos multiclasse optou-se após testes por utilizar todas as características da base de dados.

A figura seguinte nos traz as importâncias das características para o modelo *Random Forest* de classificação binária.

Tabela I
IMPORTÂNCIA DAS CARACTERÍSTICAS

Feature	Random Forest Feature Importance
dstip	0.171892
sttl	0.171047
ct_state_ttl	0.158517
sbytes	0.085213
srcip	0.080752
dmeansz	0.031187
smeansz	0.030895
dbytes	0.025733

Fonte: Autor

Em que dstip é o endereço IP de origem, sttl é o tempo de vida do pacote da requisição, ct_state_ttl é o número para cada estado de acordo com um intevalo de valores para o tempo de sobrevivência dos pacotes de origem e destino, sbytes são os bytes da requisição, scrip é o endereço de IP da origem, dmeansz é a média do tamanho do pacote de dados transmitidos pelo destino, smeansz é a média do tamanho do pacote de dados transmitidos pela origem e dbytes são os bytes de resposta.

D. Execução dos Modelos e Validação

Para a etapa de execução dos modelos as principais bibliotecas usadas foram *Scikit-Learn*, *HiClass* e *XGBoost*. Nesse sentido, todos os modelos foram executados com seus melhores parâmetros achados na etapa de seleção de características e salvos em arquivos pickle. Após isso foi feita a validação

cruzada para todos os modelos para extração de métricas como acurácia, acurácia balanceada, precisão, recall e f1-score.

E. Consolidação de Métricas e Visualização dos Desempenhos

Durante a etapa de validação cruzada foram coletadas todas as métricas de cada treino e teste para futura consolidação e avaliação dos modelos.

VI. Resultados

Primeiramente, são mostrados os resultados das médias de acurácia, precisão, recall e f1-score dos modelos Adaptive Boosting, Random Forest, Extreme Gradient Boosting, Bagging e Multilayer Perceptron. Posteriormente, os resultados dos modelos hierárquicos locais por nó, por nó pai e por nível são mostrados. Além disso, as matrizes de confusão são mostradas e trazem as quantidades de acertos e erros de classificação em uma perspectiva diferente e mais intuitiva das diferenças dos modelos das medidas unitárias como acurácia, precisão, recall e f1-score.

A. Métricas para os Modelos Binários

As próximas tabelas mostram as métricas de acurácia, precisão, recall e f1-score dos modelos Adaptive Boosting, Random Forest, Extreme Gradient Boosting, Bagging e Multilayer Perceptron na classificação binária.

Tabela II
Médias das Métricas para Adaptive Boosting

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Normal	0.99008	0.984432	0.996627	0.991997	0.994306
Attack	0.99008	0.984432	0.946629	0.976867	0.961496

Fonte: Autor

Tabela III
Desvio Padrão das Métricas para Adaptive Boosting

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Normal	0.000468	0.001027	0.000414	0.000907	0.000275
Attack	0.000468	0.001027	0.005474	0.002891	0.001539

Fonte: Autor

Tabela IV
Médias das Métricas para Random Forest

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Normal	0.992835	0.981564	0.995139	0.996663	0.995900
Attack	0.992835	0.981564	0.976767	0.966465	0.971588

Fonte: Autor

Tabela V
Desvio Padrão das Métricas para Random Forest

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Normal	0.00021	0.000718	0.000209	0.000068	0.000121
Attack	0.00021	0.000718	0.000350	0.001418	0.000798

Fonte: Autor

Tabela VI
Médias das Métricas para Extreme Gradient Boosting

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Normal	0.992021	0.982725	0.995682	0.995178	0.995430
Attack	0.992021	0.982725	0.966901	0.970272	0.968583

Fonte: Autor

Tabela VII
Desvio Padrão das Métricas para Extreme Gradient Boosting

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Normal	0.000179	0.000221	0.000072	0.000191	0.000106
Attack	0.000179	0.000221	0.001108	0.000439	0.000543

Fonte: Autor

Tabela VIII
Médias das Métricas para Bagging

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Normal	0.986528	0.991463	0.999716	0.984851	0.992228
Attack	0.986528	0.991463	0.905347	0.998075	0.949452

Fonte: Autor

Tabela IX
Desvio Padrão das Métricas para Bagging

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Normal	0.000228	0.000263	0.000066	0.000258	0.000138
Attack	0.000228	0.000263	0.001042	0.000443	0.000595

Fonte: Autor

Tabela X
Médias das métricas para Multilayer Perceptron

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Normal	0.989932	0.981786	0.995765	0.992697	0.994227
Attack	0.989932	0.981786	0.950908	0.970876	0.960703

Fonte: Autor

Tabela XI
Desvio Padrão das Métricas para Multilayer Perceptron

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Normal	0.000876	0.005266	0.001697	0.001580	0.000503
Attack	0.000876	0.005266	0.009373	0.011786	0.003441

Fonte: Autor

Pode-se observar que os modelos que trabalham com a classificação binária tem bons resultados em suas métricas, no entanto dois modelos se destacam: Random Forest e Extreme Gradient Boosting.

B. Métricas para os Modelos Multiclasse

As próximas tabelas mostram as métricas de acurácia, precisão, recall e f1-score dos modelos Adaptive Boosting, Random Forest, Extreme Gradient Boosting, Bagging e Multilayer Perceptron na classificação multiclasse.

Tabela XII
MÉDIAS DAS MÉTRICAS PARA ADAPTIVE BOOSTING

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Analysis	0.963769	0.42039	0.049700	0.460048	0.089426
Backdoors	0.963769	0.42039	0.033755	0.036515	0.025713
DoS	0.963769	0.42039	0.247432	0.138318	0.135973
Exploits	0.963769	0.42039	0.643191	0.561701	0.585021
Fuzzers	0.963769	0.42039	0.357247	0.319173	0.321097
Generic	0.963769	0.42039	0.987042	0.974678	0.980808
Reconnaissance	0.963769	0.42039	0.560558	0.650585	0.599229
Shellcode	0.963769	0.42039	0.000000	0.000000	0.000000
Worms	0.963769	0.42039	0.069811	0.074762	0.062739
Normal	0.963769	0.42039	0.991638	0.988117	0.989862

Fonte: Autor

Tabela XIII
DESVIO PADRÃO DAS MÉTRICAS PARA ADAPTIVE BOOSTING

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Analysis	0.00366	0.01605	0.007408	0.065894	0.011862
Backdoors	0.00366	0.01605	0.030579	0.056022	0.024922
DoS	0.00366	0.01605	0.025277	0.185867	0.107453
Exploits	0.00366	0.01605	0.044040	0.154408	0.083302
Fuzzers	0.00366	0.01605	0.051920	0.131923	0.070385
Generic	0.00366	0.01605	0.007388	0.001327	0.003360
Reconnaissance	0.00366	0.01605	0.027525	0.080660	0.031176
Shellcode	0.00366	0.01605	0.000000	0.000000	0.000000
Worms	0.00366	0.01605	0.076923	0.054022	0.045643
Normal	0.00366	0.01605	0.002895	0.005217	0.001479

Fonte: Autor

Tabela XIV
MÉDIAS DAS MÉTRICAS PARA RANDOM FOREST

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Analysis	0.983547	0.584948	0.871317	0.121515	0.212761
Backdoors	0.983547	0.584948	0.817051	0.097931	0.174462
DoS	0.983547	0.584948	0.309749	0.235063	0.267138
Exploits	0.983547	0.584948	0.631257	0.824145	0.714890
Fuzzers	0.983547	0.584948	0.779630	0.683635	0.728400
Generic	0.983547	0.584948	0.997990	0.987289	0.992611
Reconnaissance	0.983547	0.584948	0.925980	0.780141	0.846814
Shellcode	0.983547	0.584948	0.844426	0.901570	0.871401
Worms	0.983547	0.584948	0.644286	0.219683	0.323480
Normal	0.983547	0.584948	0.997471	0.998512	0.997991

Fonte: Autor

Tabela XV
DESVIO PADRÃO DAS MÉTRICAS PARA RANDOM FOREST

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Analysis	0.00019	0.006296	0.056395	0.018372	0.027934
Backdoors	0.00019	0.006296	0.094251	0.025596	0.042614
DoS	0.00019	0.006296	0.009501	0.014207	0.011467
Exploits	0.00019	0.006296	0.004040	0.008044	0.002619
Fuzzers	0.00019	0.006296	0.013173	0.004958	0.003216
Generic	0.00019	0.006296	0.000341	0.000785	0.000449
Reconnaissance	0.00019	0.006296	0.004943	0.003995	0.002341
Shellcode	0.00019	0.006296	0.032506	0.027736	0.015410
Worms	0.00019	0.006296	0.126006	0.052427	0.066074
Normal	0.00019	0.006296	0.000171	0.000151	0.000106

Fonte: Autor

Tabela XVI
MÉDIAS DAS MÉTRICAS PARA EXTREME GRADIENT BOOSTING

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Analysis	0.98352	0.610389	0.848095	0.084735	0.153981
Backdoors	0.98352	0.610389	0.893626	0.096329	0.173710
DoS	0.98352	0.610389	0.434234	0.300773	0.355235
Exploits	0.98352	0.610389	0.650545	0.864893	0.742513
Fuzzers	0.98352	0.610389	0.746045	0.604409	0.667615
Generic	0.98352	0.610389	0.997497	0.987327	0.992386
Reconnaissance	0.98352	0.610389	0.921843	0.787112	0.849157
Shellcode	0.98352	0.610389	0.878125	0.906155	0.891304
Worms	0.98352	0.610389	0.622240	0.474127	0.536058
Normal	0.98352	0.610389	0.996349	0.998032	0.997190

Fonte: Autor

Tabela XVII
DESVIO PADRÃO DAS MÉTRICAS PARA EXTREME GRADIENT BOOSTING

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Analysis	0.000279	0.011347	0.052733	0.010251	0.017436
Backdoors	0.000279	0.011347	0.068484	0.017689	0.029928
DoS	0.000279	0.011347	0.014471	0.011609	0.009958
Exploits	0.000279	0.011347	0.009170	0.007539	0.006011
Fuzzers	0.000279	0.011347	0.018360	0.011898	0.007811
Generic	0.000279	0.011347	0.000371	0.000735	0.000483
Reconnaissance	0.000279	0.011347	0.005421	0.003874	0.003414
Shellcode	0.000279	0.011347	0.029204	0.034638	0.018721
Worms	0.000279	0.011347	0.110054	0.100276	0.102827
Normal	0.000279	0.011347	0.000129	0.000085	0.000080

Fonte: Autor

Tabela XVIII
MÉDIAS DAS MÉTRICAS PARA BAGGING

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Analysis	0.873237	0.1	NaN	0.0	0.000000
Backdoors	0.873237	0.1	NaN	0.0	0.000000
DoS	0.873237	0.1	NaN	0.0	0.000000
Exploits	0.873237	0.1	NaN	0.0	0.000000
Fuzzers	0.873237	0.1	NaN	0.0	0.000000
Generic	0.873237	0.1	NaN	0.0	0.000000
Reconnaissance	0.873237	0.1	NaN	0.0	0.000000
Shellcode	0.873237	0.1	NaN	0.0	0.000000
Worms	0.873237	0.1	NaN	0.0	0.000000
Normal	0.873237	0.1	0.873237	1.0	0.932329

Fonte: Autor

Tabela XIX
DESVIO PADRÃO DAS MÉTRICAS PARA BAGGING

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Analysis	0.0006	0.0	NaN	0.0	0.000000
Backdoors	0.0006	0.0	NaN	0.0	0.000000
DoS	0.0006	0.0	NaN	0.0	0.000000
Exploits	0.0006	0.0	NaN	0.0	0.000000
Fuzzers	0.0006	0.0	NaN	0.0	0.000000
Generic	0.0006	0.0	NaN	0.0	0.000000
Reconnaissance	0.0006	0.0	NaN	0.0	0.000000
Shellcode	0.0006	0.0	NaN	0.0	0.000000
Worms	0.0006	0.0	NaN	0.0	0.000000
Normal	0.0006	0.0	0.0006	0.0	0.000342

Fonte: Autor

Tabela XX
MÉDIAS DAS MÉTRICAS PARA MULTILAYER PERCEPTRON

	Acurácia no Teste	Acurácia Balanceada no Teste	Precisão	Recall	F1
Analysis	0.979068	0.510448	0.609618	0.023673	0.044621
Backdoors	0.979068	0.510448	0.644488	0.059450	0.106346
DoS	0.979068	0.510448	0.417983	0.330088	0.273759
Exploits	0.979068	0.510448	0.658738	0.766836	0.687759
Fuzzers	0.979068	0.510448	0.670194	0.437320	0.523099
Generic	0.979068	0.510448	0.993927	0.983560	0.988716
Reconnaissance	0.979068	0.510448	0.819709	0.754799	0.784980
Shellcode	0.979068	0.510448	0.515739	0.508747	0.504477
Worms	0.979068	0.510448	0.360000	0.242540	0.221685
Normal	0.979068	0.510448	0.994035	0.997467	0.995748

Fonte: Autor

Tabela XXI
DESVIO PADRÃO DAS MÉTRICAS PARA MULTILAYER PERCEPTRON

	Acurácia no Teste Std	Acurácia Balanceada no Teste Std	Precisão Std	Recall Std	F1 Std
Analysis	0.000585	0.040709	0.265125	0.020212	0.036654
Backdoors	0.000585	0.040709	0.158881	0.023299	0.038498
DoS	0.000585	0.040709	0.106859	0.318446	0.173810
Exploits	0.000585	0.040709	0.099236	0.177334	0.028693
Fuzzers	0.000585	0.040709	0.065906	0.066727	0.027387
Generic	0.000585	0.040709	0.000886	0.000863	0.000483
Reconnaissance	0.000585	0.040709	0.043247	0.027294	0.018071
Shellcode	0.000585	0.040709	0.069056	0.098002	0.050255
Worms	0.000585	0.040709	0.261367	0.271713	0.134337
Normal	0.000585	0.040709	0.000400	0.000479	0.000125

Fonte: Autor

Novamente os modelos que melhor performam são *Extreme Gradient Boosting* e *Random Forest*. No entanto, desta vez a maior acurácia balanceada se encontra no modelo Extreme Gradient Boosting com aproximadamente 61%. Os tipos estados que conseguem ser mais reconhecidos pelos modelos são ataque *Generic* e o estado de normalidade da rede que são os dois estados mais bem representados pela base.

C. Métricas dos Modelos Hierárquicos

As próximas tabelas mostram as acurácias em classificação binária e multiclasse dos modelos de classificação local por nó, por nó pai e por nível.

Tabela XXII
MÉDIAS DAS ACURÁCIAS PARA CLASSIFICADOR POR NÓ BINÁRIO

	Acurácia_Binário	Acurácia_Multiclasse	Acurácia_Balanceada_Binário	Acurácia_Balanceada_Multiclasse
Mean	0.996450	0.983481	0.991431	0.594525
Std	0.000136	0.000293	0.000420	0.012811

Fonte: Autor

Tabela XXIII
MÉDIAS DAS ACURÁCIAS PARA CLASSIFICADOR POR NÓ PAI BINÁRIO

	Acurácia_Binário	Acurácia_Multiclasse	Acurácia_Balanceada_Binário	Acurácia_Balanceada_Multiclasse
Mean	0.996447	0.983456	0.991381	0.587681
Std	0.000133	0.000278	0.000384	0.007172

Fonte: Autor

Tabela XXIV
MÉDIAS DAS ACURÁCIAS PARA CLASSIFICADOR POR NÍVEL BINÁRIO

	Acurácia_Binário	Acurácia_Multiclasse	Acurácia_Balanceada_Binário	Acurácia_Balanceada_Multiclasse
Mean	0.996447	0.983427	0.991381	0.587190
Std	0.000133	0.000225	0.000384	0.006871

Fonte: Autor

As próximas tabelas mostram as métricas de precisão, recall e f1-score em classificação binária e multiclasse dos modelos de classificação local por nó, por nó pai e por nível.

Tabela XXV
MÉDIAS DAS MÉTRICAS PARA CLASSIFICADOR POR NÓ BINÁRIO

	Precision_Mean	Recall_Mean	F1_Mean
Normal	0.997781	0.998155	0.997968
Attack	0.987262	0.984706	0.985982

Fonte: Autor

Tabela XXVI
DESVIO PADRÃO DAS MÉTRICAS PARA CLASSIFICADOR POR NÓ BINÁRIO

	Precision_Std	Recall_Std	F1_Std
Normal	0.000128	0.000169	0.000079
Attack	0.001105	0.000915	0.000513

Fonte: Autor

Tabela XXVII
MÉDIAS DAS MÉTRICAS PARA CLASSIFICADOR POR NÓ PAI BINÁRIO

	Precision_Mean	Recall_Mean	F1_Mean
Normal	0.997765	0.998168	0.997966
Attack	0.987347	0.984595	0.985968

Fonte: Autor

Tabela XXVIII
DESVIO PADRÃO DAS MÉTRICAS PARA CLASSIFICADOR POR NÓ PAI BINÁRIO

	Precision_Std	Recall_Std	F1_Std
Normal	0.000120	0.000175	0.000077
Attack	0.001148	0.000850	0.000492

Fonte: Autor

Tabela XXIX
MÉDIAS DAS MÉTRICAS PARA CLASSIFICADOR POR NÍVEL BINÁRIO

	Precision_Mean	Recall_Mean	F1_Mean
Normal	0.997765	0.998168	0.997966
Attack	0.987347	0.984595	0.985968

Fonte: Autor

Tabela XXX
DESVIO PADRÃO DAS MÉTRICAS PARA CLASSIFICADOR POR NÍVEL BINÁRIO

	Precision_Std	Recall_Std	F1_Std
Normal	0.000120	0.000175	0.000077
Attack	0.001148	0.000850	0.000492

Fonte: Autor

Tabela XXXI
MÉDIAS DAS MÉTRICAS PARA CLASSIFICADOR POR NÓ MULTICLASSE

	Precision_Mean	Recall_Mean	F1_Mean
Analysis	0.817582	0.127165	0.219782
Backdoors	0.843748	0.103787	0.184486
DoS	0.312674	0.239655	0.271189
Exploits	0.630403	0.824973	0.714655
Fuzzers	0.757627	0.699358	0.727214
Generic	0.997945	0.987798	0.992845
Reconnaissance	0.928426	0.779149	0.847249
Shellcode	0.868891	0.900768	0.884078
Worms	0.564286	0.284444	0.376370
Normal	0.997781	0.998155	0.997968

Fonte: Autor

Tabela XXXII
DESVIO PADRÃO DAS MÉTRICAS PARA CLASSIFICADOR POR NÓ MULTICLASSE

	Precision_Std	Recall_Std	F1_Std
Analysis	0.045897	0.016800	0.025665
Backdoors	0.094951	0.027023	0.045023
DoS	0.008938	0.013972	0.010784
Exploits	0.006092	0.006938	0.004239
Fuzzers	0.018706	0.002890	0.007712
Generic	0.000367	0.000786	0.000463
Reconnaissance	0.006374	0.003742	0.003250
Shellcode	0.031323	0.029924	0.021099
Worms	0.115230	0.095280	0.108476
Normal	0.000128	0.000169	0.000079

Fonte: Autor

Tabela XXXIII
MÉDIAS DAS MÉTRICAS PARA CLASSIFICADOR POR NÓ PAI MULTICLASSE

	Precision_Mean	Recall_Mean	F1_Mean
Analysis	0.851156	0.126527	0.219837
Backdoors	0.891321	0.098019	0.176129
DoS	0.314806	0.242599	0.273835
Exploits	0.630253	0.823923	0.714164
Fuzzers	0.757163	0.698749	0.726685
Generic	0.997898	0.987429	0.992635
Reconnaissance	0.928417	0.781137	0.848419
Shellcode	0.845588	0.900579	0.871759
Worms	0.603175	0.219683	0.319201
Normal	0.997765	0.998168	0.997966

Fonte: Autor

Tabela XXXIV
DESVIO PADRÃO DAS MÉTRICAS PARA CLASSIFICADOR POR NÓ PAI MULTICLASSE

	Precision_Std	Recall_Std	F1_Std
Analysis	0.069354	0.017075	0.025853
Backdoors	0.072175	0.025535	0.042184
DoS	0.007815	0.015183	0.011168
Exploits	0.005031	0.007249	0.003268
Fuzzers	0.018280	0.003410	0.008181
Generic	0.000371	0.000804	0.000461
Reconnaissance	0.005170	0.003782	0.002222
Shellcode	0.025833	0.027813	0.015313
Worms	0.127440	0.052427	0.069697
Normal	0.000120	0.000175	0.000077

Fonte: Autor

Tabela XXXV
MÉDIAS DAS MÉTRICAS PARA CLASSIFICADOR POR NÍVEL MULTICLASSE

	Precision_Mean	Recall_Mean	F1_Mean
Analysis	0.824467	0.126545	0.218778
Backdoors	0.817051	0.097931	0.174462
DoS	0.309394	0.235067	0.267005
Exploits	0.628746	0.826129	0.714019
Fuzzers	0.761097	0.697845	0.728005
Generic	0.997990	0.987326	0.992630
Reconnaissance	0.925832	0.780283	0.846838
Shellcode	0.841262	0.902922	0.870382
Worms	0.644286	0.219683	0.323480
Normal	0.997765	0.998168	0.997966

Fonte: Autor

Tabela XXXVI
DESVIO PADRÃO DAS MÉTRICAS PARA CLASSIFICADOR POR NÍVEL MULTICLASSE

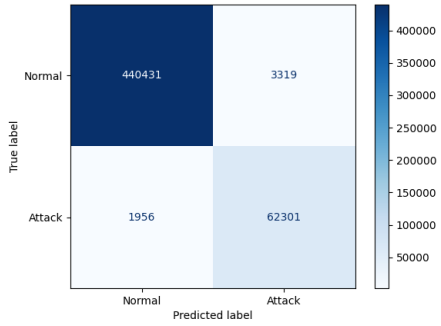
	Precision_Std	Recall_Std	F1_Std
Analysis	0.068442	0.017045	0.024754
Backdoors	0.094251	0.025596	0.042614
DoS	0.009163	0.014214	0.011343
Exploits	0.004194	0.007566	0.002421
Fuzzers	0.015837	0.005193	0.006445
Generic	0.000341	0.000765	0.000437
Reconnaissance	0.004561	0.003970	0.002522
Shellcode	0.031692	0.026886	0.015278
Worms	0.126006	0.052427	0.066074
Normal	0.000120	0.000175	0.000077

Fonte: Autor

Pode-se perceber uma boa acurácia para as classificações em níveis binários (ataque ou normalidade) com o classificador local por nó tendo uma acurácia balanceada levemente superior ao dos modelos por nó pai e nível. Já nas precisões da classificação multiclasse, em comparação com o modelo Extreme Gradient Boosting, não existe grande diferença.

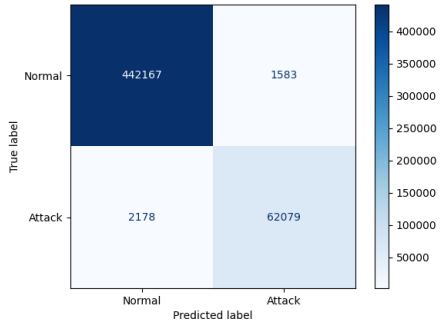
D. Matrizes de Confusão dos Modelos Binários

Figura 3. Matriz de Confusão do Modelo Adaptive Boosting



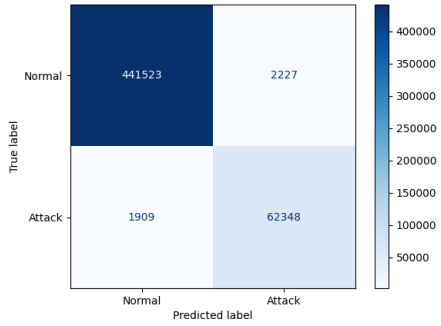
Fonte: Autor

Figura 4. Matriz de Confusão do Modelo Random Forest



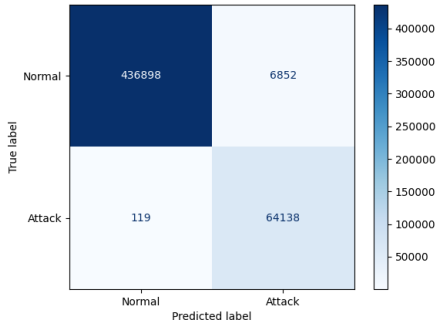
Fonte: Autor

Figura 5. Matriz de Confusão do Modelo Extreme Gradient Boosting



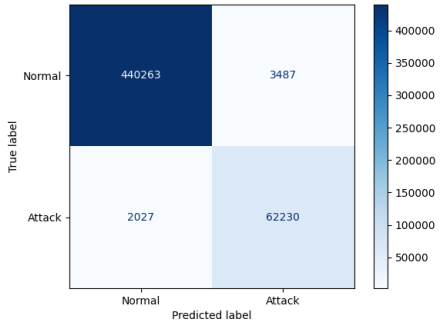
Fonte: Autor

Figura 6. Matriz de Confusão do Modelo Bagging



Fonte: Autor

Figura 7. Matriz de Confusão do Modelo Multilayer Perceptron

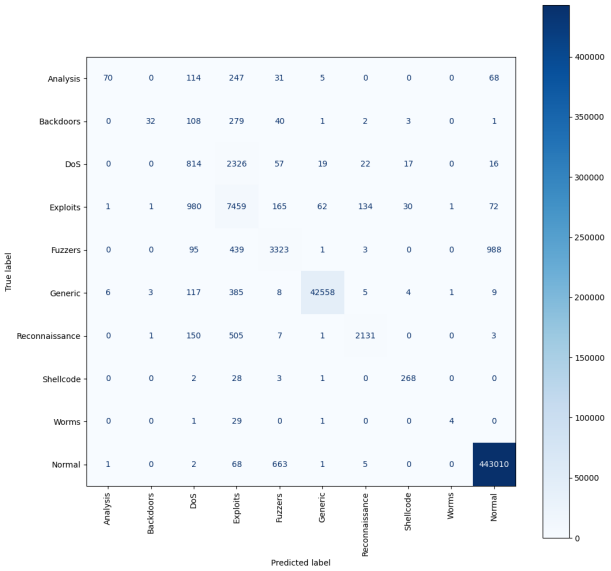


Fonte: Autor

Como esperado, os modelos mais notáveis são *Random Forest* e *Extreme Gradient Boosting*. Porém, o modelo *Bagging* traz uma situação de identifcar pouquíssimas situações de ataque como normal, no entanto, confunde muitas vezes o estado normal com o anormal.

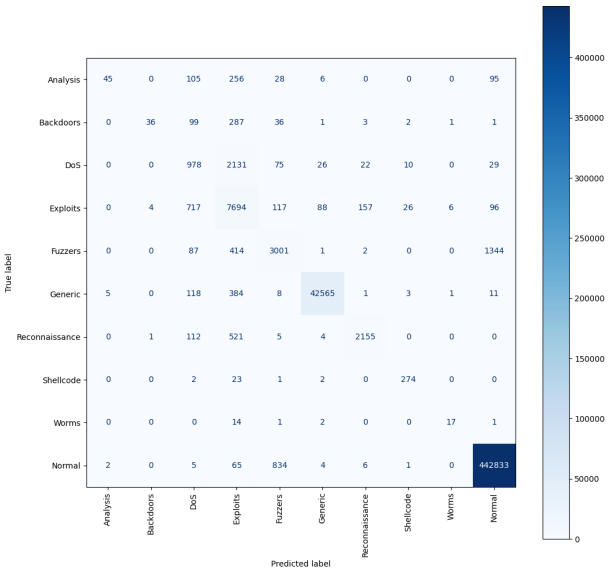
E. Matrizes de Confusão dos Modelos Multiclasse

Figura 8. Matriz de Confusão do Modelo Random Forest



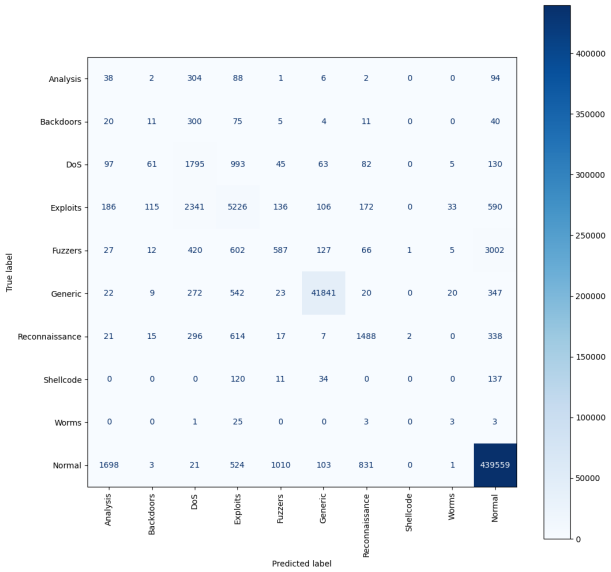
Fonte: Autor

Figura 9. Matriz de Confusão do Modelo Extreme Gradient Boosting



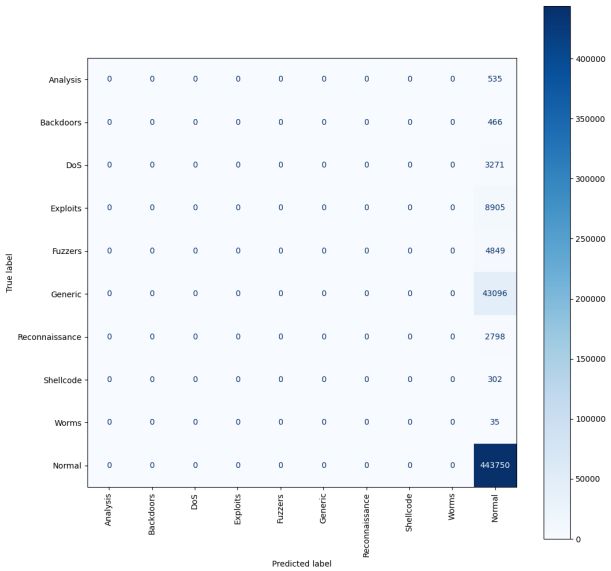
Fonte: Autor

Figura 10. Matriz de Confusão do Modelo Adaptive Boosting



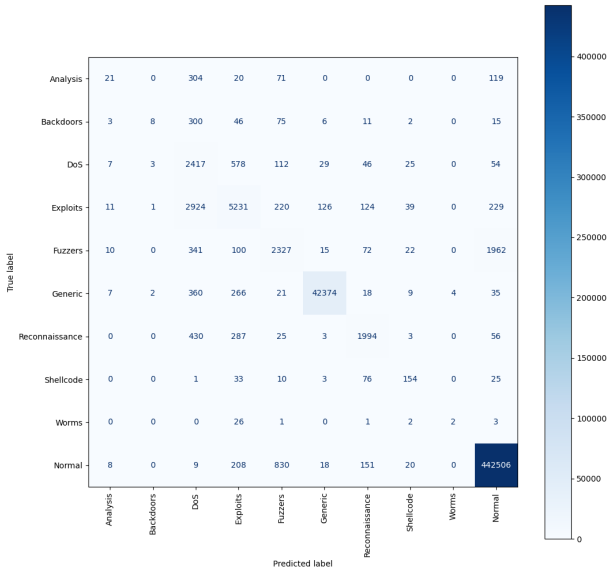
Fonte: Autor

Figura 11. Matriz de Confusão do Modelo Bagging



Fonte: Autor

Figura 12. Matriz de Confusão do Modelo Multilayer Perceptron



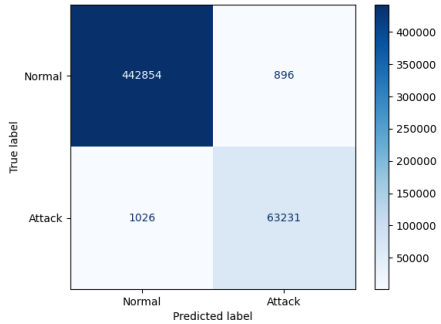
Fonte: Autor

Com as matrizes de confusão conseguimos observar que os modelos *Random Forest* e *Extreme Gradient Boosting* tem as menores dispersões de classificações erradas por classe.

F. Matrizes de Confusão dos Modelos Hierárquicos

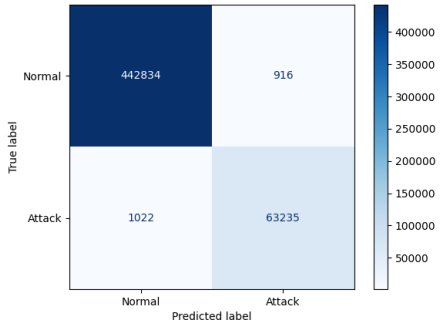
As figuras seguintes trazem as matrizes das classificações binárias dos modelos de classificação hierárquicos.

Figura 13. Matriz de Confusão do Modelo Por Nó



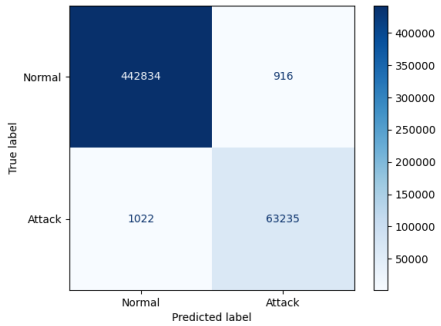
Fonte: Autor

Figura 14. Matriz de Confusão do Modelo Por Nó Pai



Fonte: Autor

Figura 15. Matriz de Confusão do Modelo Por Nível

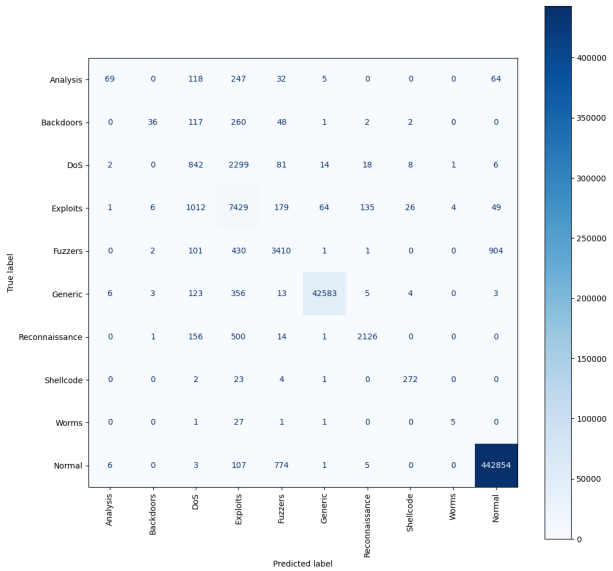


Fonte: Autor

As matrizes mostradas evidenciam como os modelos hierárquicos reduziram os falsos positivos e falsos negativos de modo geral na classificação binária em comparação com os modelos emsemble e Multilayer Perceptron.

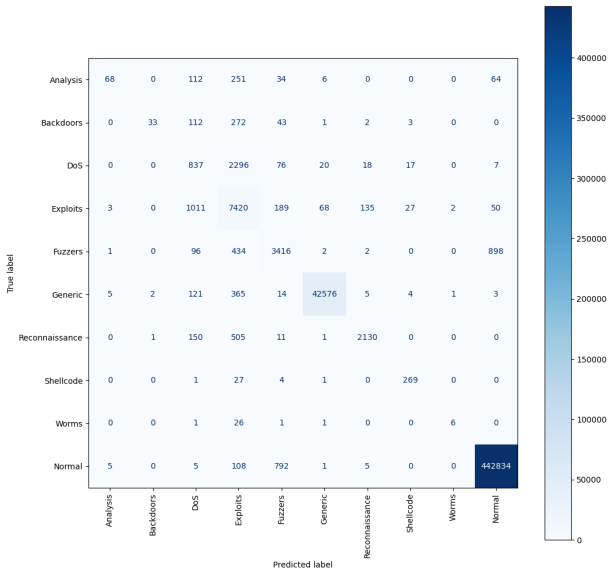
As próximas figuras trazem as matrizes das classifcações multiclasse dos modelos de classificação hierárquicos.

Figura 16. Matriz de Confusão do Modelo Por Nó



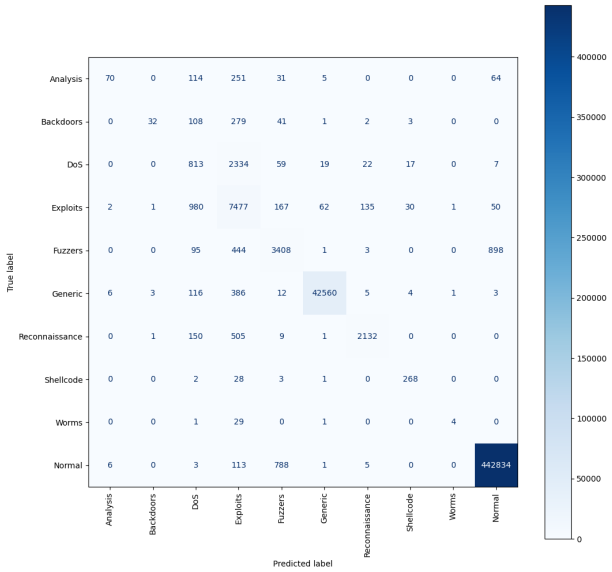
Fonte: Autor

Figura 17. Matriz de Confusão do Modelo Por Nó Pai



Fonte: Autor

Figura 18. Matriz de Confusão do Modelo Por Nível



Fonte: Autor

VII. Conclusão

Conclui-se no final que todos os modelos foram bastante eficazes, com evidência maior para os modelos *Extreme Gradient Boosting*, *Random Forest* e hierárquicos na detecção de estados binários de ataque ou normalidade. No entanto, na diferenciação de estado normal e tipos de ataques os modelos não são ideais para uso, isso muito em razão da falta de exemplos de ataques na base. Em outro contexto, com a expansão da digitalização em todos os setores da sociedade desde serviços básicos até o consumo de serviços de nuvem por grandes empresas, é de interesse geral que existam bons sistemas de detecção de intrusões munidos com algoritmos que consigam detectar tanto ataques conhecidos quanto novos. Nesse sentido, a criação de novas bases de dados que simulam comportamentos de redes atuais é imprescindível e a aplicação de algoritmos tanto clássicos quanto mais modernos, como os modelos baseados em redes neurais, podem ajudar a pelo menos mititgar o grande trabalho que é detectar essas ameaças cibernéticas.

Referências

[1] CHANG, Brittany. One of the biggest US insurance companies reportedly paid hackers $40 million ransom after a cyberattack. Business Insider. Disponível em: https://www.businessinsider.com/cna-financial-hackers-40-million-ransom- cyberattack-2021-5

[2] AXELSSON, Stefan. Intrusion Detection Systems: A Survey and Taxonomy. CiteSeer. Abril, 2000.

[3] RAKSHE, Tushar. GONJARI, Vishal. Anomaly based Network Intrusion Detection using Machine Learning Techniques. International Journal of Engineering Research & Technology (IJERT). Maio, 2017.

[4] AKASHI SATO, Felipe Yuzo. Análise do Desempenho de Algoritmos Classificadores para Detecção de Anomalias em Sistemas de Detecção de Intrusão. Universidade Tecnológica Federal do Paraná (UTFPR). Junho, 2023.

[5] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)."Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.
[6] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set."Information Security Journal: A Global Perspective (2016): 1-14.
[7] MIRANDA, Fabio M. KÖHNECKE, Niklas. RENARD, Bernhard Y. HiClass: a Python Library for Local Hierarchical Classification Compatible with Scikit-learn. Janeiro, 2023.