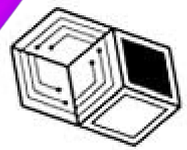
Dados, Questões de pesquisa e Resoluções



Programa de
Pós-Graduação em
Ciência da Computação

UEM Universidade Estadual de Maringá

Progresso de desenvolvimento

import pandas as pd

Biblioteca fundamental para análise de dados em Python. Ela fornece estruturas de dados eficientes, como DataFrame e Series, que facilitam a manipulação e a análise de dados tabulares.

Progresso de desenvolvimento

from sklearn.model_selection import train_test_split

Esta função divide um conjunto de dados em conjuntos de treinamento e teste. Isso é essencial para avaliar o desempenho de modelos de aprendizado de máquina.

Progresso de desenvolvimento

from sklearn.ensemble import RandomForestClassifier

RandomForestClassifier é um modelo de aprendizado de máquina que utiliza uma combinação de várias árvores de decisão (floresta aleatória) para classificar dados. Ele é robusto e pode lidar com dados de alta dimensão e interações não lineares.

Progresso de desenvolvimento

from sklearn.metrics import accuracy_score

Esta função calcula a acurácia de um modelo, que é a proporção de previsões corretas em relação ao total de previsões feitas.

- Fórmula: $\text{Acurácia} = \frac{\text{Número de Previsões corretas}}{\text{Total de Previsões}}$

Progresso de desenvolvimento

from sklearn.preprocessing import LabelEncoder

LabelEncoder é uma classe que converte rótulos categóricos em números inteiros. Isso é útil para preparar dados antes de treinar modelos, já que muitos algoritmos de aprendizado de máquina não podem trabalhar diretamente com dados categóricos.

Progresso de desenvolvimento

from sklearn.feature_extraction.text import TfidfVectorizer

TfidfVectorizer é uma ferramenta para converter um conjunto de documentos de texto em uma matriz de características utilizando a técnica TF-IDF (Term Frequency-Inverse Document Frequency).

Progresso de desenvolvimento

from sklearn.feature_extraction.text import TfidfVectorizer

TfidfVectorizer é uma ferramenta para converter um conjunto de documentos de texto em uma matriz de características utilizando a técnica TF-IDF (Term Frequency-Inverse Document Frequency).

Common Weakness Enumeration

A community-developed list of SW & HW weaknesses that can become vulnerabilities

1425 - Weaknesses in the 2023 CWE Top 25 Most Dangerous Software Weaknesses

- Out-of-bounds Write - *(787)*
- Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') - *(79)*
- Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') - *(89)*
- Use After Free - *(416)*
- Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') - *(78)*
- Improper Input Validation - *(20)*
- Out-of-bounds Read - *(125)*
- Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') - *(22)*
- Cross-Site Request Forgery (CSRF) - *(352)*
- Unrestricted Upload of File with Dangerous Type - *(434)*
- Missing Authorization - *(862)*
- NULL Pointer Dereference - *(476)*
- Improper Authentication - *(287)*
- Integer Overflow or Wraparound - *(190)*
- Deserialization of Untrusted Data - *(502)*
- Improper Neutralization of Special Elements used in a Command ('Command Injection') - *(77)*
- Improper Restriction of Operations within the Bounds of a Memory Buffer - *(119)*
- Use of Hard-coded Credentials - *(798)*
- Server-Side Request Forgery (SSRF) - *(918)*
- Missing Authentication for Critical Function - *(306)*
- Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') - *(362)*
- Improper Privilege Management - *(269)*
- Improper Control of Generation of Code ('Code Injection') - *(94)*
- Incorrect Authorization - *(863)*
- Incorrect Default Permissions - *(276)*

Após classificar todas as 25 vulnerabilidades com o Dataset e aplicar as bibliotecas e ferramentas, o próximo passo está sendo calibrar os resultados variados devido a variabilidade de diversos códigos de linguagens diferentes contidas entre as vulnerabilidades.

Questões de Pesquisa e Resoluções

- Existe uma ou mais vulnerabilidades com 100% de eficácia de detecção pelo software? A que se deve esse fator?
- Existe uma ou mais vulnerabilidades com 0% de eficácia de detecção pelo software? A que se deve esse fator?
- Quantos caracteres ou linhas de código podem ser modificadas em código anteriormente analisado e detectado para que o software seja incapaz de reconhecer a vulnerabilidade?