Evasive PDF Samples

Inteligência Artificial 2023/2024

Group A2_38:
Pedro Filipe Vale Gomes (202108825)
Simão Queirós Rodrigues (202005700)

Specification of the Work to be Performed

- This project focuses on enhancing malware detection in PDF files by addressing evasion attacks, which are attempts to bypass existing detection mechanisms.

- The dataset used consists of evasive PDF samples, labeled as either malicious (1) or benign (0).

- By using machine learning algorithms, this project's main objective is to develop a malware detector in PDF files, capable of resisting evasion attacks.We'll also test the results with other detectors, and evaluate the robustness of each algorithm in comparison with ours.

Related Work and References

- Exploration of existing studies and technologies focused on detecting malware in PDF files, with a special emphasis on evasion attacks.

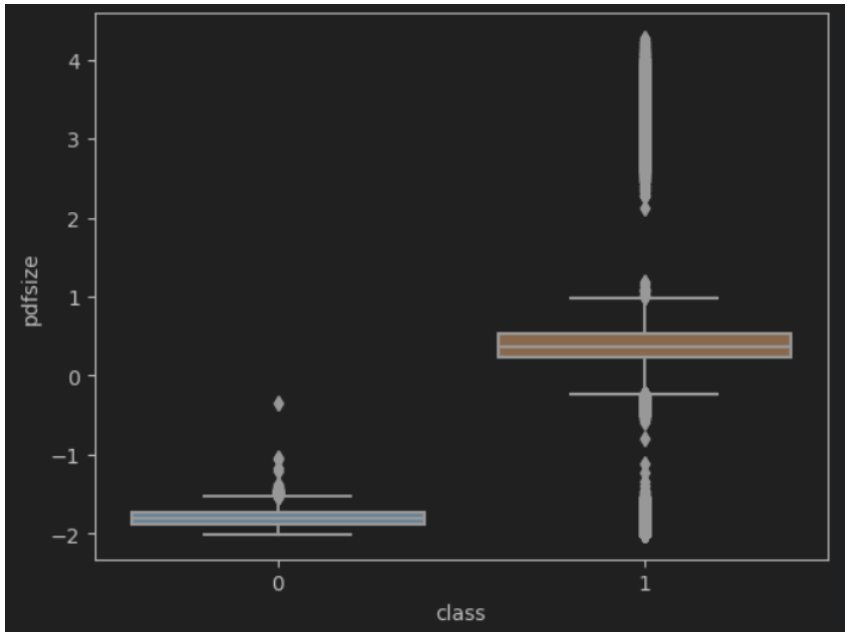
- https://www.kaggle.com/datasets/fouadtrad2/evasive-pdf-samples This is where the dataset was downloaded from, and from which the initial understanding of the project was obtained, as well as the description of the dataset's columns.
- https://doi.org/10.3390/app13063472 - Trad, F.; Hussein, A.; Chehab, A. Leveraging Adversarial Samples for Enhanced Classification of Malicious and Evasive PDF Files. Appl. Sci. 2023, 13, 3472. This is the paper cited by the reference above. It provides a thorough study of the problem and information about the different approaches that can be taken.

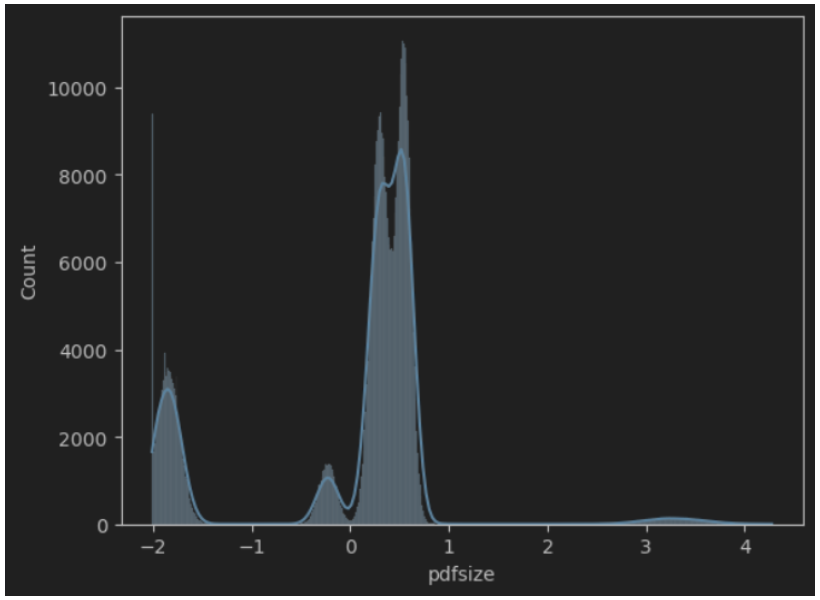
Tools and Algorithms

- **Python:** For all back-end algorithms and data manipulation.
- **Scikit-Learn:** Machine learning library for implementing various classification algorithms and data pre-processing.
- **Pandas:** Data structures and data analysis tools for reading and manipulating data.
- **NumPy:** Numerical computing with support for large, multi-dimensional arrays and matrices.
- **Matplotlib/Seaborn:** Plotting libraries for visualizing the dataset and results.
- **Jupyter Notebook / DataSpell:** Interactive computing environments where the project is being developed and documented.

Implementation Work Already Carried Out

- **Data Acquisition:** The dataset comprising evasive PDF samples was sourced and loaded for analysis.
- **Exploratory Data Analysis (EDA):** Conducted an initial exploration to understand data characteristics, including the distribution of features and checking for missing values.
- **Data Preprocessing:** Normalized data using StandardScaler to ensure feature scaling and variance homogeneity.
- **Data Splitting:** Segregated the dataset into training (80%) and test (20%) sets to prepare for unbiased model evaluation.





Chosen Algorithms

- **Decision Tree**

Description: Splits data into branches forming a tree structure.

Justification: Good performance and highly interpretable, making the decision process easy to understand.

- **K-Nearest Neighbors (K-NN)**

Description: Classifies based on the closest K neighbors.

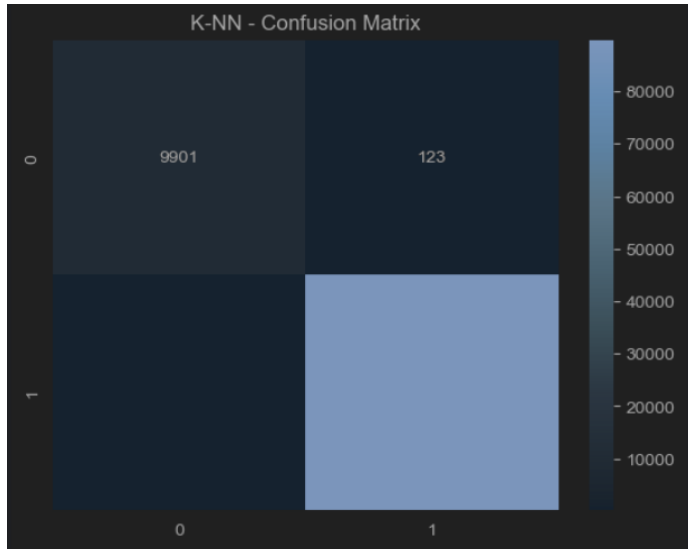
Justification: Simple to understand and implement, relying on similar known examples.

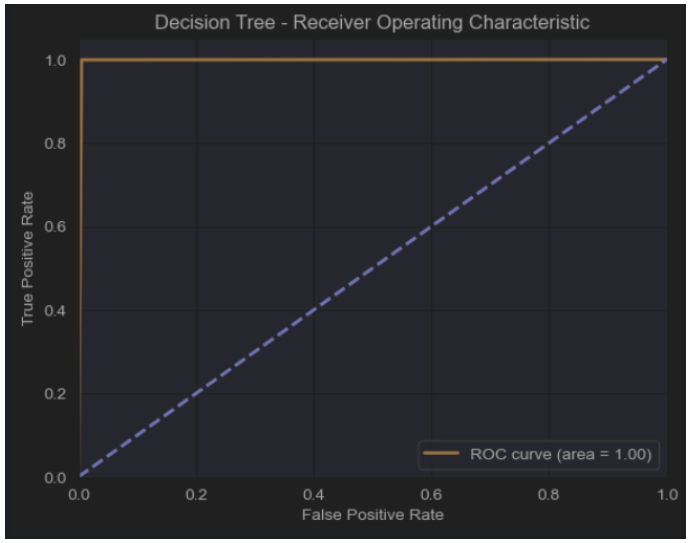
- **Neural Network (MLPClassifier)**

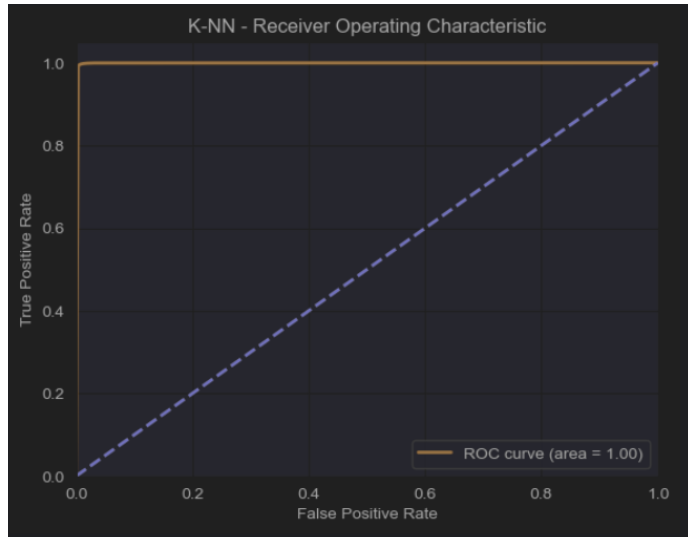
Description: Layers of neurons that learn to recognize patterns.

Justification: Models complex relationships and achieves high accuracy, suitable for complex tasks.

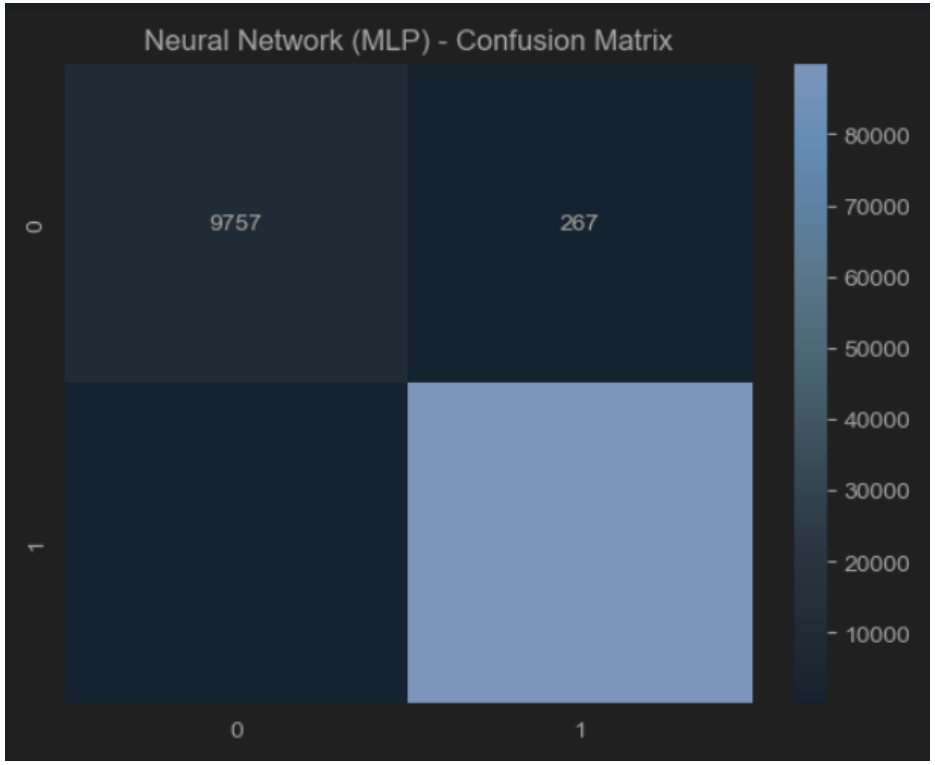
Decision Tree and K-NN - Results

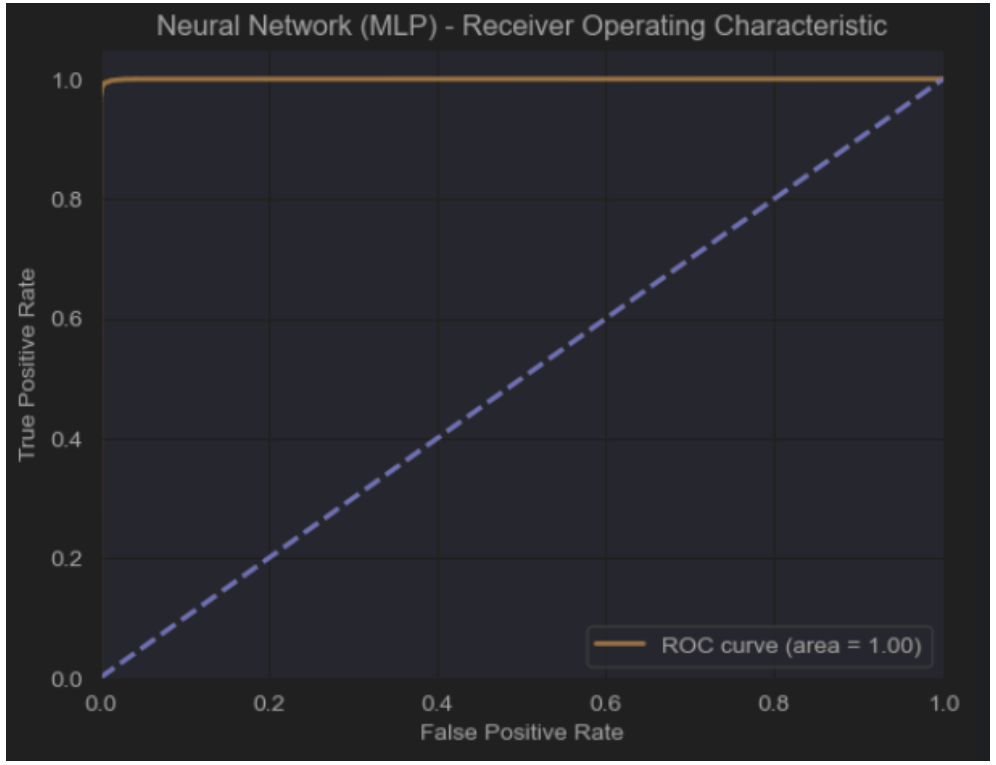






Neural Network (MLP) - Results





Model comparison

	Model	Accuracy	Precision	Recall	F1-Score
0	Decision Tree	0.99869	0.998691	0.99869	0.998690
1	K-NN	0.99703	0.997038	0.99703	0.997033
2	Neural Network	0.99657	0.996567	0.99657	0.996550

Conclusion

Overall, all three algorithms demonstrated high accuracy and effectiveness for the given classification task. The Decision Tree algorithm slightly outperformed the others in this instance, making it a highly recommended choice for tasks requiring interpretability and strong performance. However, K-NN and Neural Network remain valuable options depending on the specific needs and complexity of the problem.