Detecting Phishing Websites: A Deep Learning Approach with MLP

Jonathan Garza, Ulysses Ochoa, Ahmad Kirkland

Dataset

The dataset used in this project was obtained from the UCI Machine Learning Repository and contains over 11,000 samples. Each sample is described by 30 features derived from website attributes, such as the presence of an IP address in the URL, URL length, usage of URL shortening services, SSL certificate validity, and favicon consistency. The dataset was originally structured with feature values ranging from -1 to 1, where -1 represents a negative feature, 0 indicates an absent or neutral feature, and 1 represents a positive feature. To ensure compatibility with the Multilayer Perceptron (MLP) model, the dataset was preprocessed by converting -1 values to 0, resulting in a range of [0,1]. This dataset was chosen for its well-documented structure, clear distinctions between phishing and legitimate websites, and minimal preprocessing requirements, making it ideal for developing and testing our phishing detection model.

Tools

Programming Libraries:

-NumPy and Pandas: Data manipulation and preprocessing.

-PyTorch: Model development and training.

- Sklearn: Division of dataset into training and testing

- SciPy: arff file type compatibility

- Matplot and Seaborn: Data Visualization

FrameWorks:

-Python for implementing the machine learning pipeline.

Methods

Model Architecture:

-Input Layer: 30 neurons (one for each feature).

-Hidden Layers: 64, 32, and 16 neurons with ReLU activation functions.

-Output Layer: 1 neuron with sigmoid activation for binary classification.

Training Process:

Optimizer: Adam, with a learning rate of 0.001.

Loss Function: Binary cross-entropy for classification tasks.

Learning Rate Scheduler: StepLR, reducing learning rate dynamically to prevent overshooting.

Data Preprocessing:

-Feature values adjusted from [-1, 1] to [0, 1] for compatibility with the model.

Implementation

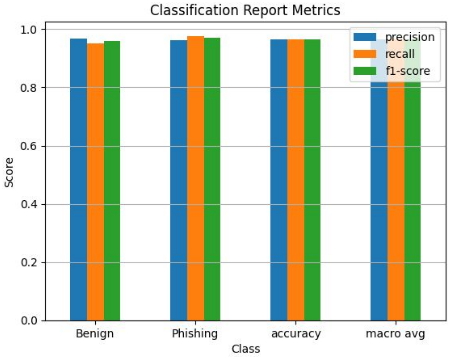
The implementation began with preprocessing the dataset by transforming feature values from the range `[-1, 1]` to `[0, 1]` to ensure compatibility with the Multilayer Perceptron (MLP) model. Redundant and unnecessary features were removed to enhance processing efficiency. The MLP model was built using PyTorch with an architecture comprising an input layer of 30 neurons (one for each dataset feature), three hidden layers with 64, 32, and 16 neurons respectively, all utilizing ReLU activation functions, and a single output neuron with a sigmoid activation function for binary classification. The training process used the Adam optimizer with an initial learning rate of 0.001, along with a binary cross-entropy loss function suitable for classification tasks. A StepLR learning rate scheduler was implemented to reduce the learning rate dynamically by half every 5 epochs, ensuring optimal convergence.

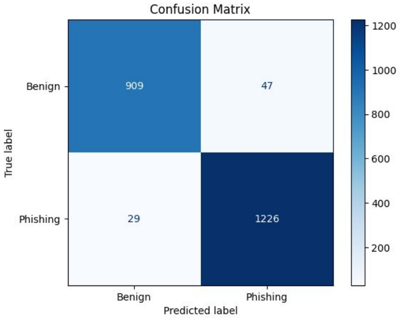
Hyperparameter tuning involved experimenting with the number of epochs, ranging from 25 to 30, and increasing the learning rate to 0.01 for better convergence. The model was trained on an 80/20 train-test split, with performance monitored using metrics such as precision, recall, F1-score, and accuracy. During evaluation, the model demonstrated strong performance with an accuracy of 95–97% and a balanced confusion matrix, highlighting minimal false positives and negatives.

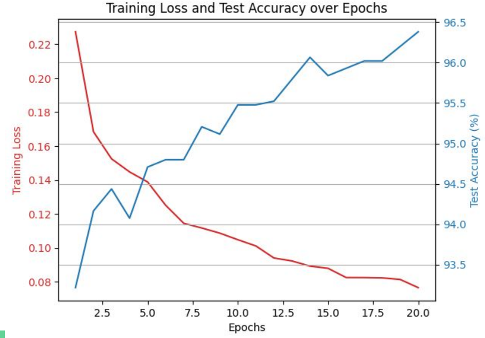
Results

The phishing detection model achieved exceptional performance in classifying phishing and legitimate websites. The overall accuracy ranged between 95% and 97%, with precision and recall values of 96% and 97%, respectively, for both phishing and benign classifications. The F1-score, which balances precision and recall, consistently reached 97%, reflecting robust overall performance. A confusion matrix analysis showed that the model correctly identified 1,226 phishing websites (true positives) and 909 legitimate websites (true negatives), while only misclassifying 47 legitimate websites as phishing (false positives) and 29 phishing websites as legitimate (false negatives). This balance demonstrates the model's ability to minimize security risks by reducing false negatives while maintaining a low false-positive rate.

The training and testing process also highlighted effective convergence, with the training loss steadily decreasing and the test accuracy increasing over epochs. These results indicate the model’s robustness and reliability, making it suitable for real-world deployment in phishing detection systems where both precision and recall are critical.







Examples

To demonstrate the model's effectiveness, we tested it with real-world examples of phishing and legitimate websites. For instance, a phishing website might contain suspicious features such as an unusually long URL, the presence of an IP address instead of a domain name, or the use of URL shortening services. The model correctly flagged these as phishing with high confidence. On the other hand, legitimate websites typically showed characteristics like valid SSL certificates, consistent favicon usage, and shorter URL lengths, which the model accurately classified as benign.

In one specific case, a phishing website attempting to mimic a banking portal with subtle URL variations was correctly identified, showcasing the model's ability to detect obfuscation strategies. Similarly, benign websites, such as e-commerce platforms with proper security measures, were consistently classified as safe. These examples highlight the model’s practical application in distinguishing between phishing and legitimate websites in diverse scenarios.

Conclusion

This project developed a Multilayer Perceptron (MLP) model that achieved 95–97% accuracy in detecting phishing websites. By leveraging structured data, effective preprocessing, and advanced training strategies, the model demonstrated strong performance with minimal false positives and negatives. These results highlight its potential for real-world deployment and its role in addressing cybersecurity threats. Future work can focus on expanding features and improving real-time detection capabilities.

References

Mohammad, R. M., Thabtah, F. A., & McCluskey, L. (2012). *An Assessment of Features Related to Phishing Websites Using an Automated Technique*. 2012 International Conference for Internet Technology and Secured Transactions (ICITST), 492–497.

Mamun, S. A., Rathore, M. M., Lashkari, A. H., Stakhanova, N., & Ghorbani, A. A. (2016). *Detecting Malicious URLs Using Lexical Analysis*. Network and System Security, Springer International Publishing, 467–482.

UCI Machine Learning Repository. *Phishing Websites Dataset*. Retrieved from https://archive.ics.uci.edu/ml/datasets/phishing+websites.

PyTorch Documentation. (2024). *PyTorch Library Overview*. Retrieved from https://pytorch.org.

Scikit-learn Developers. (2024). *Scikit-learn User Guide*. Retrieved from https://scikit-learn.org.

Contributions

- Jonathan
  - MLP Algorithm and Google Collaboration Adaptation
- Ahmad
  - Presentation Slides
- Ulysses
  - Written Report and Related Work (References)