University of Pisa

CYBER SECURITY

www.ictriskassessment.com

University of Pisa

ENHANCING CYBERSECURITY:

AI STRATEGIES FOR PHISHING DETECTION AND PREVENTION

In this presentation, we explore advanced AI strategies for detecting and preventing phishing attacks. The study compares three different machine learning approaches, analyzing their effectiveness in combating the growing threat of phishing, which targets users by exploiting their trust through deceptive websites and communications

presentation by Matteo Giorgi

University of Pisa

LIST OF TOPICS

1 The phishing threat and common approaches

2 Useful metrics in AI

3 EPDB

4 PDGAN

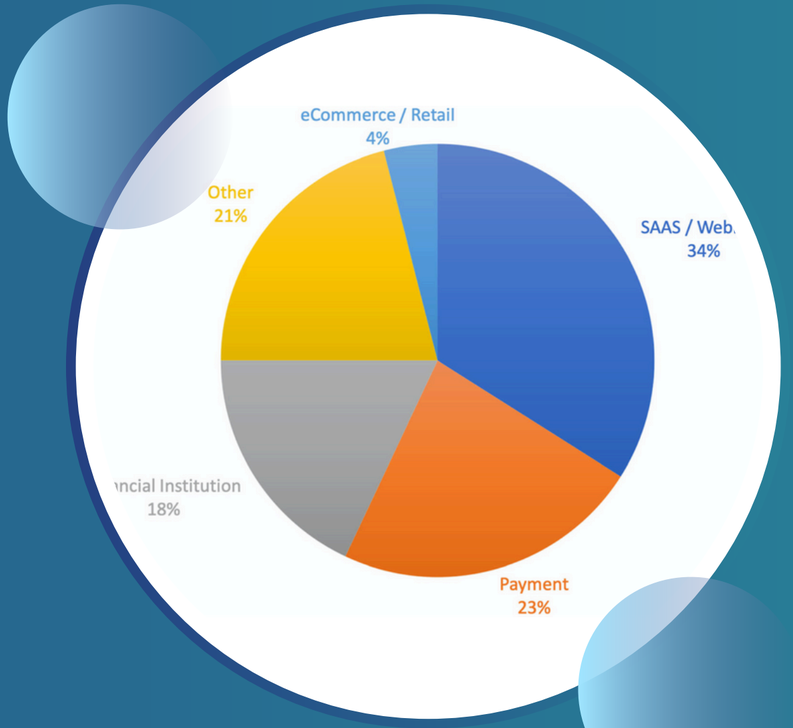
5 PhishingTransformer

6 Considerations and Conclusions

University of Pisa

THE PHISHING THREAT

- 2017 FBI Report : 25,344 phishing scams identified and $29,703,421 in losses
- 2021 Anti-Phishing Working Group : 611,877 phishing sites identified
- 2023 Security Magazine: phishing emails totaled 1.76 billion, the highest amount on record. This represents a 51% increase from 2022



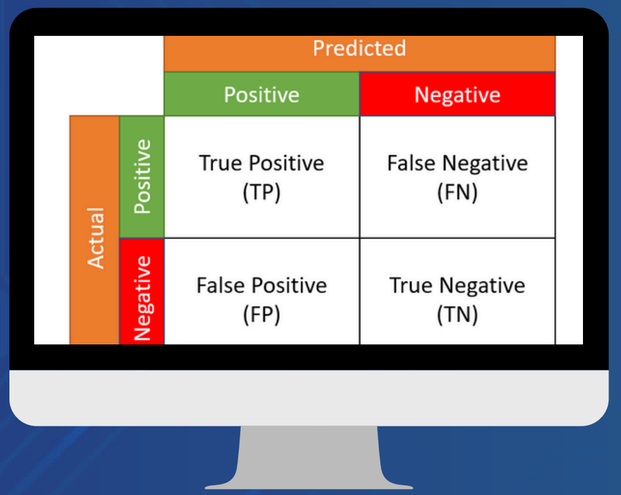
THE PHISHING THREAT

- **Browsers in the Browser (BiTB) Attacks:**
  - Malicious content running within legitimate sites
  - Exploits browser vulnerabilities using hidden iframes/windows
- **Watering Hole Attacks:**
  - Compromises legitimate sites frequented by targets
  - Redirects users to fraudulent pages
- **Clickjacking:**
  - Uses transparent overlays or hidden links
  - Deceives users into unintended actions, leading to malware installation or data theft

COMMON APPROACHES

- **Non-content based approach**
  - Blacklisting and Whitelisting
  - DNS Analysis
- **Content based approach**
  - URL Analysis and Visual Similarity
  - Spam Filters
- **Advanced Techniques Using AI**
  - Machine Learning
  - Deep Learning
  - Transformers

University of Pisa



AI PERFORMANCE MEASURES

Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Recall

$$\text{Recall} = \frac{TP}{TP+FN}$$

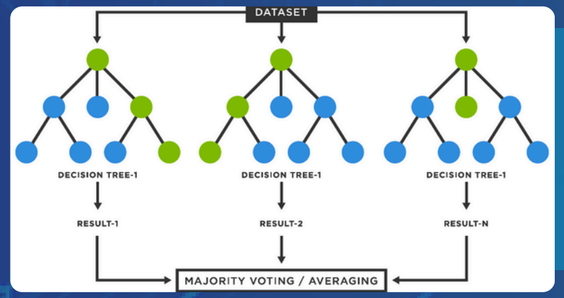
Precision

$$\text{Precision} = \frac{TP}{TP+FP}$$

F1-score or F-measure

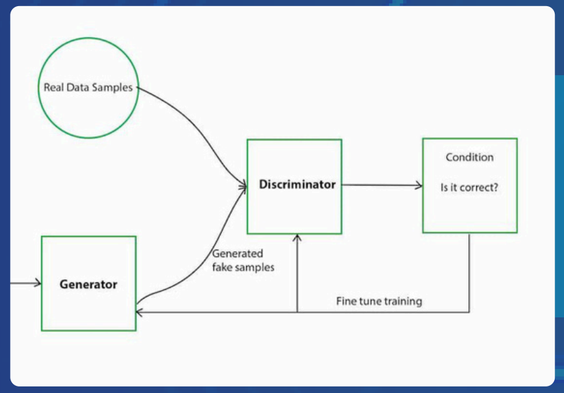
$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

LITERATURE OVERVIEW



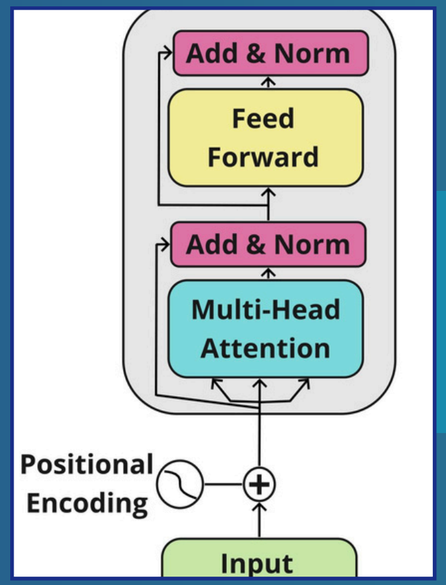
EPDB

An anti-phishing browser based on an ML model, specifically a Random Forest, which is an ensemble of decision trees



PDGAN

A GAN consisting of an LSTM generator and a CNN discriminator



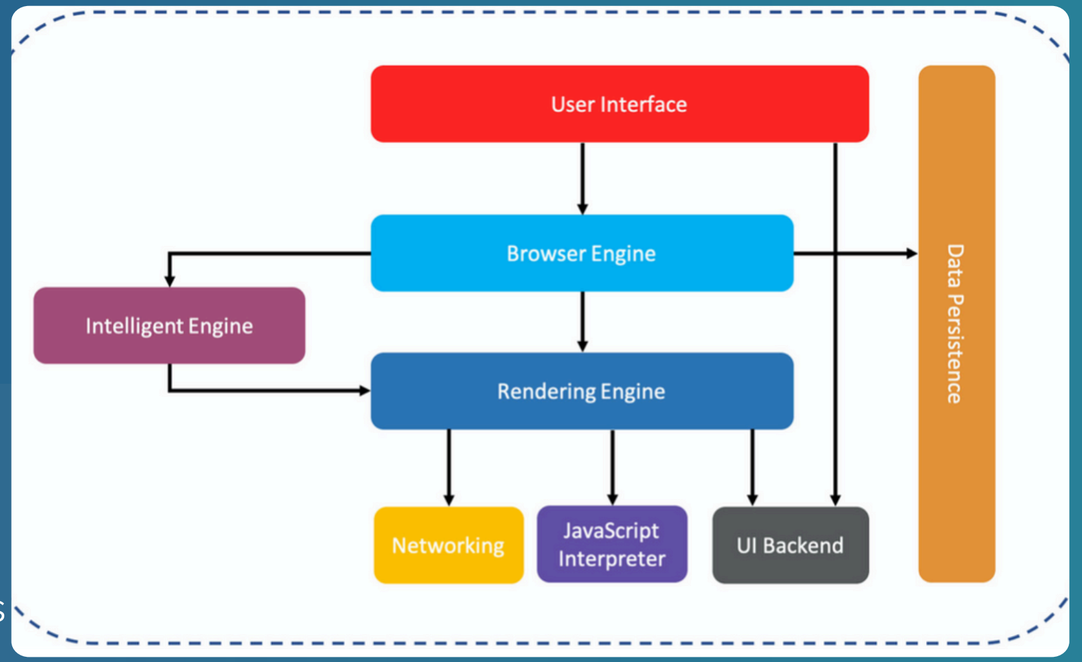
PHISHINGTRASFORMER

A DL model composed of a CNN and a Transformer encoder

University of Pisa

EPDB SYSTEM

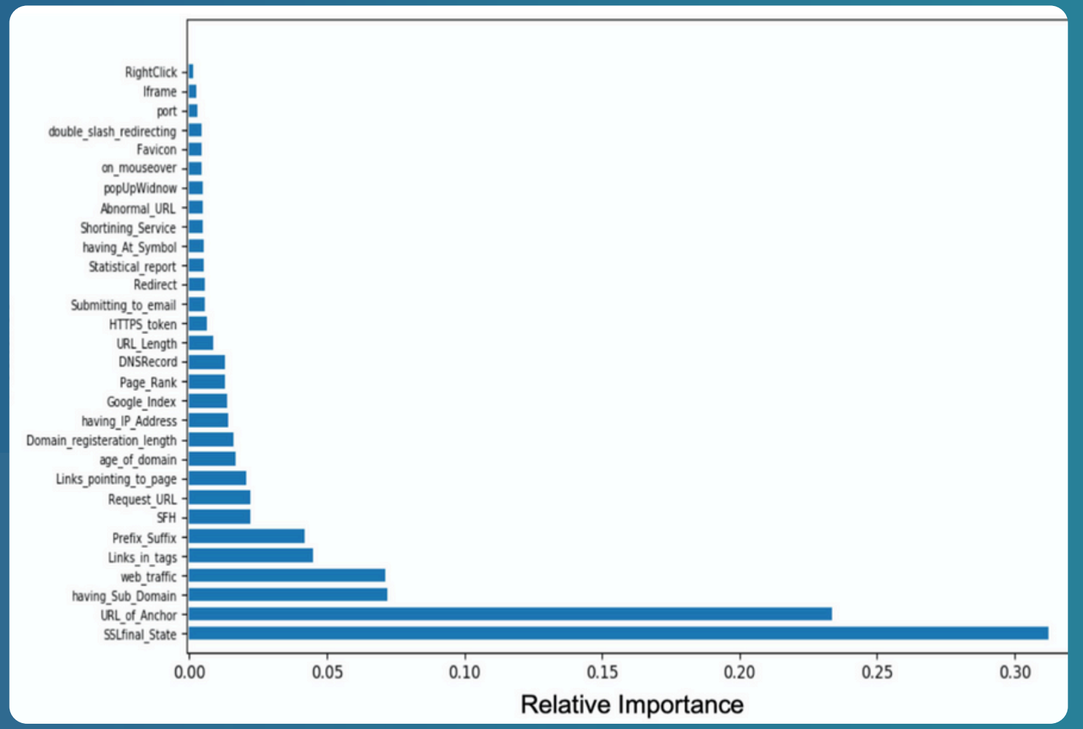
- The system consists of a part common to all browsers and an intelligent component, implemented by the RF model
- When a page needs to be loaded, the browser and the intelligent engine work simultaneously
- If the intelligent engine detects the page as fraudulent, it alerts the user with a pop-up and stops the execution of the JavaScript code.



University of Pisa

EPDB DATASET & MODEL

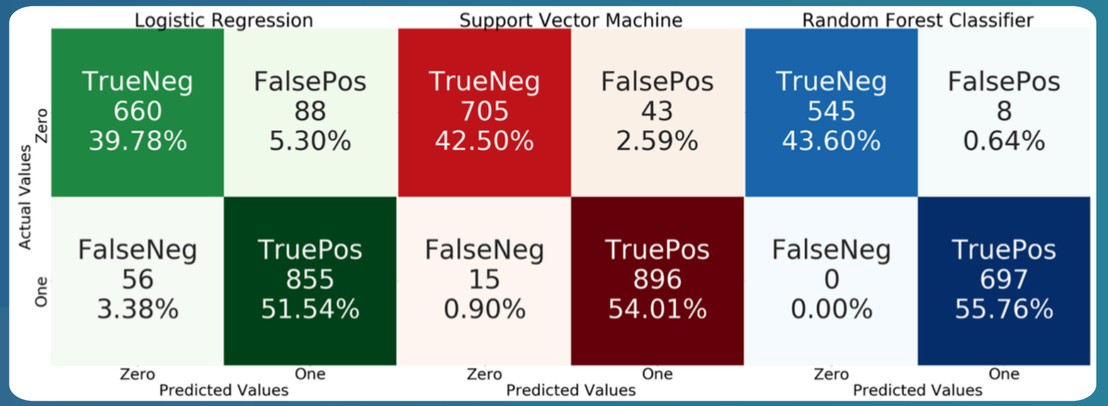
- The dataset used comes from PhishTank and MillerSmiles and includes 11,055 records, with each record consisting of 30 features associated with the URL
- The RF model is trained using 5-fold cross-validation, and the best parameters are found using GridSearchCV



University of Pisa

EPDB PERFORMANCE

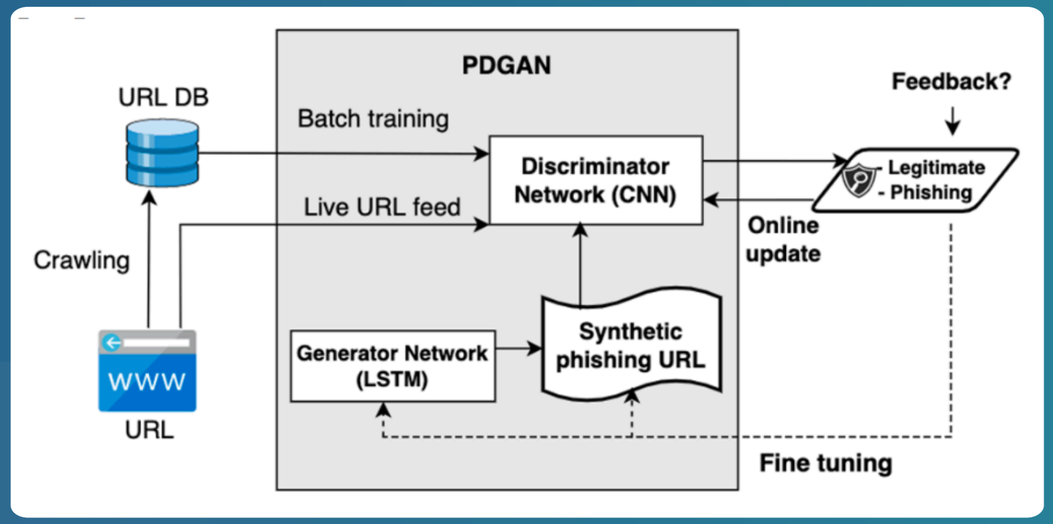
- The best model turned out to be the Random Forest, with an accuracy of 99.36% and an F1-score of 99.43%
- On average the model was able to analyze a site in 4 seconds compared to 6 seconds for the Chrome extension, making this client-side solution more efficient



University of Pisa

PDGAN SYSTEM

- The generator is trained to maximize the error made by the discriminator in recognizing the URL as synthetic, which indicates that it is improving in generating artificial URLs
- The discriminator is trained to recognize the generated URLs as accurately as possible, hence the term "adversarial neural networks."

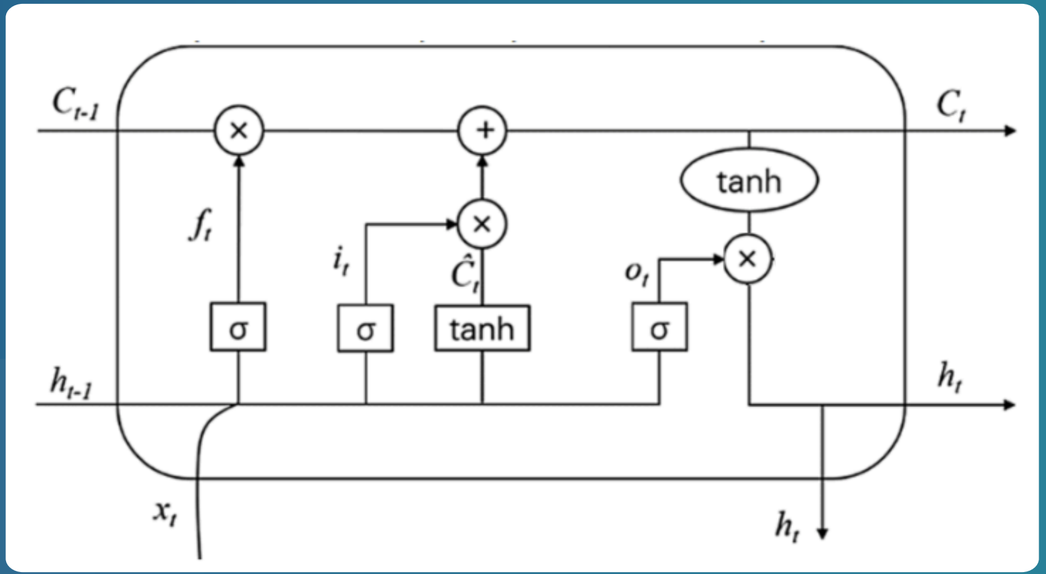


University of Pisa

PDGAN GENERATOR

- The Generator is an LSTM, composed of cells with the structure shown in the figure. A cell is a type of neural network that remembers important information for long periods and forgets less important details, allowing it to handle long sequences effectively. This is possible thanks to a memory that it updates over time using three gates:

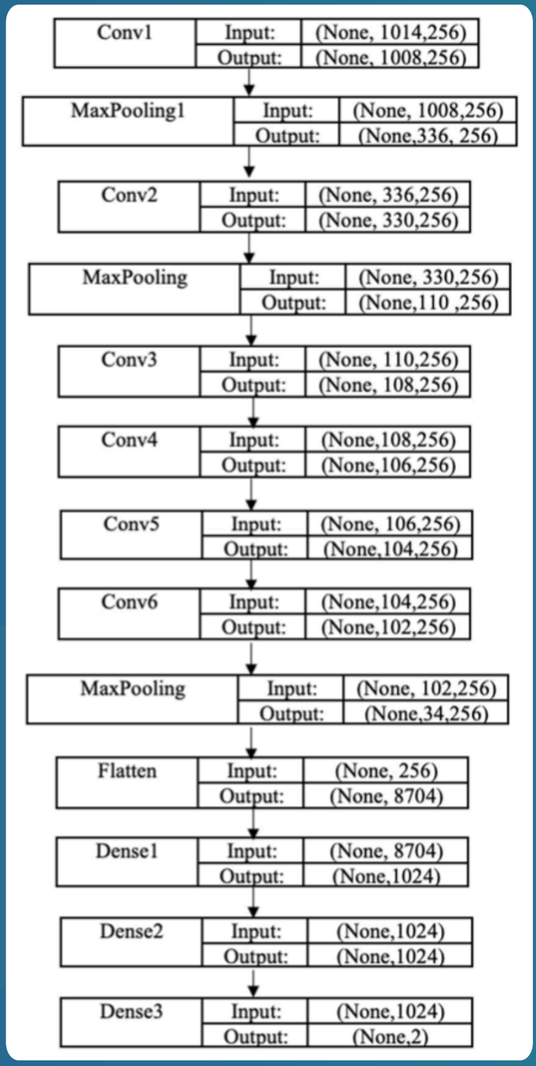
1. Forget Gate (f_t): Decides what information to discard from the memory.
2. Input Gate (i_t): Decides what new information to add to the memory.
3. Output Gate (o_t): Decides what to output based on the updated memory.



University of Pisa

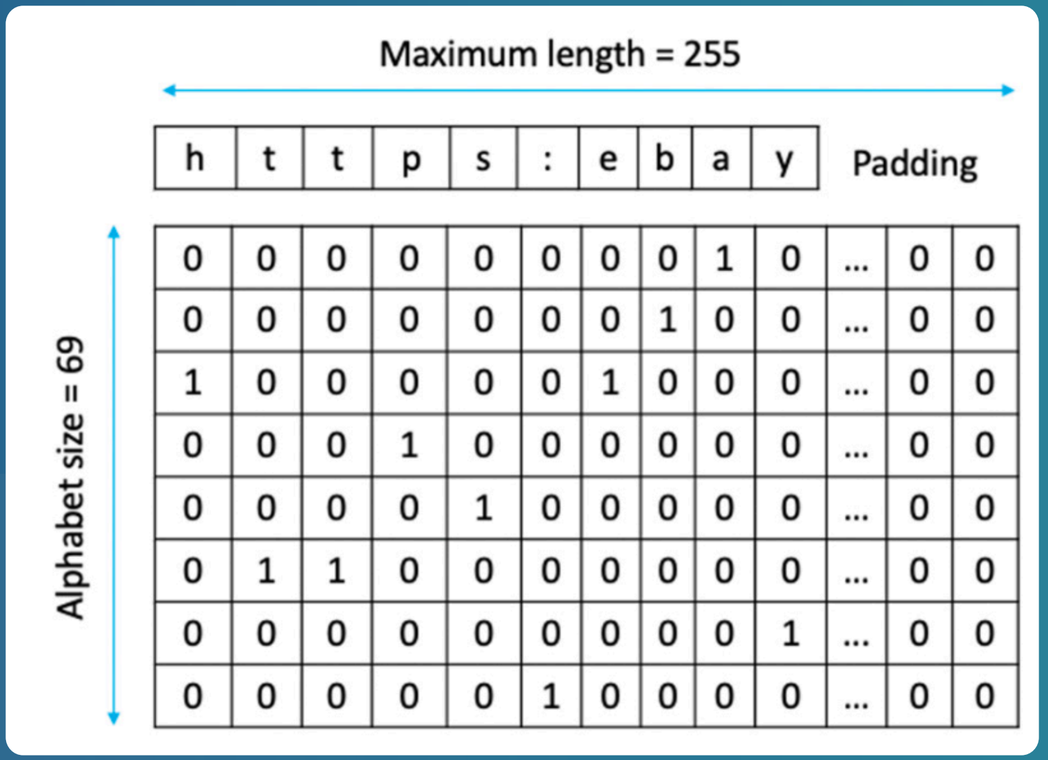
PDGAN DISCRIMINATOR

- The discriminator is a CNN consisting of 9 layers in depth, with 6 convolutional layers and 3 fully connected dense layers. One-dimensional filters were used
- Max-pooling layers were employed to extract the most relevant features and reduce the dimensionality
- Dropout layers were used to decrease the number of network parameters, thereby reducing the likelihood of overfitting.



PDGAN DATASET

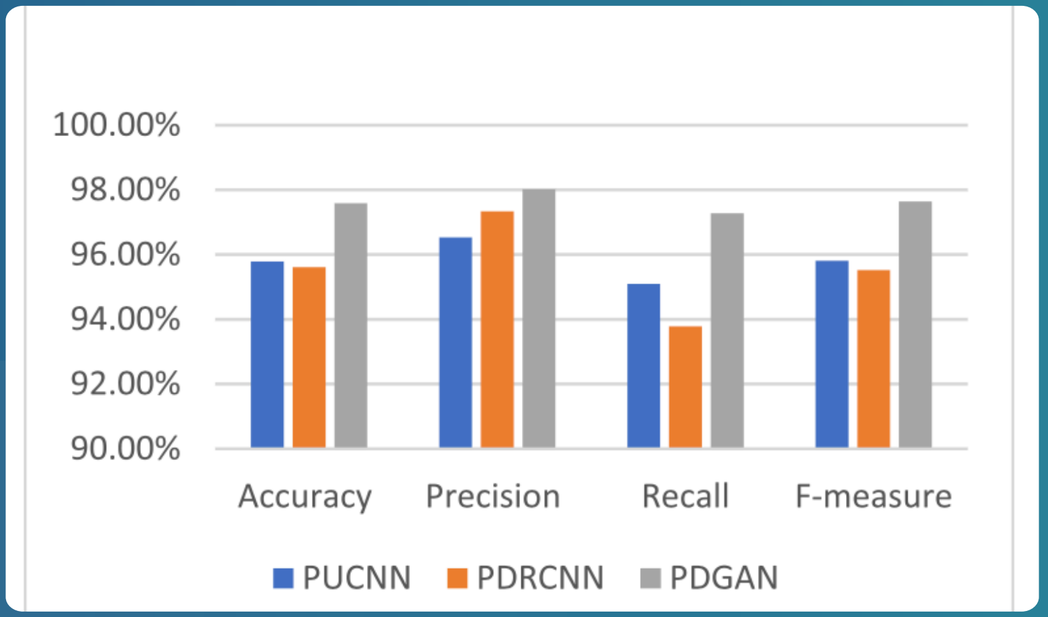
- The dataset used comes from PhishTank, specifically MUPD, a balanced collection reduced to 2.3 million URLs after preprocessing, randomly split into 60% training, 20% validation, and 20% test sets
- Each character in the URL is transformed in a one-hot vector, with a value of 1 corresponding to the represented character and all other characters set to 0, as shown in the figure.



University of Pisa

PDGAN PERFORMANCE

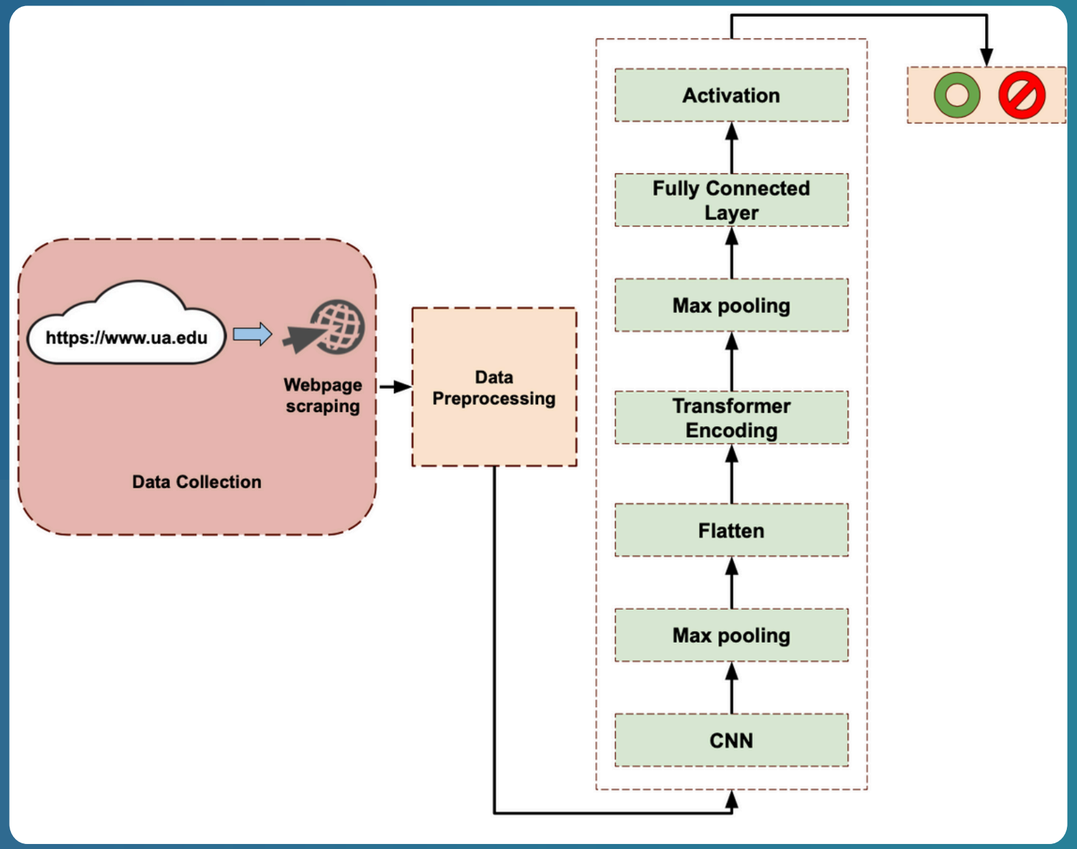
- The developed model was compared with other models that are similar in terms of dataset and model structure
- From the graph, we can see that the PDGAN model outperforms all the others across all evaluation metrics and benefits from the data augmentation provided by the generator



University of Pisa

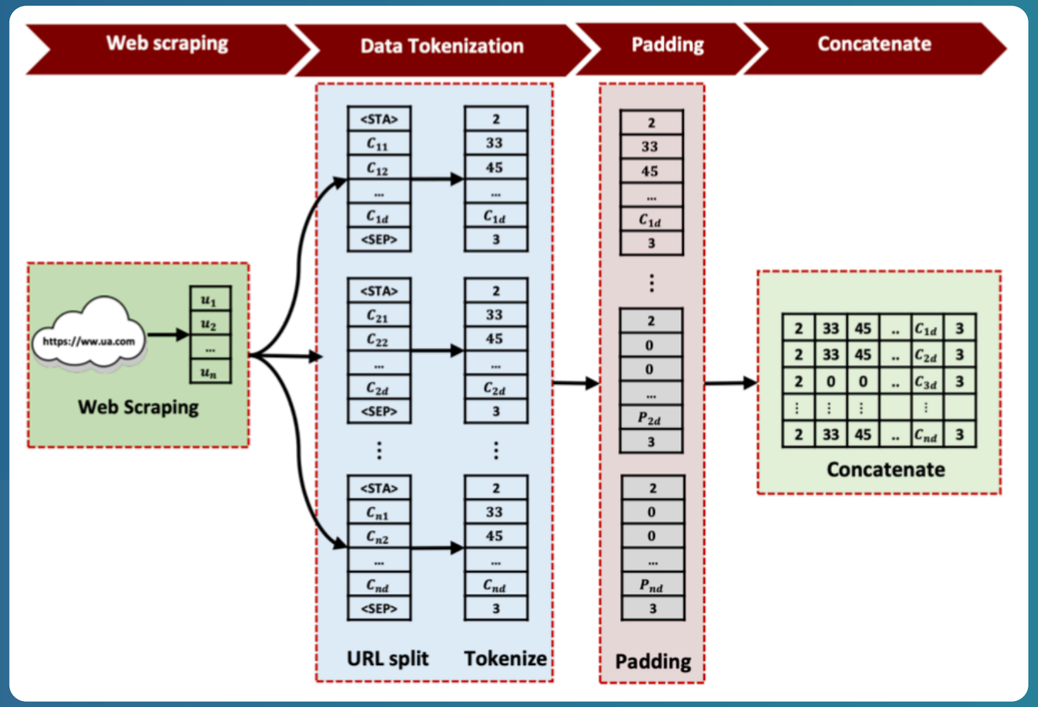
PHISHINGTRANSFORMER SYSTEM

- The system implements a hybrid approach that combines URL analysis with content analysis of the page. For each URL, the web page is visited, and the URLs contained within it are extracted through a scraping process
- The model's structure includes a CNN that extracts the local features from the URL and a Transformer encoder that identifies long-range dependencies among the various encodings



PHISHINGTRANSFORMER DATASET

- The dataset consists of a collection of 50,000 malicious URLs from Phishtank and PhishArmy and a collection of 50,000 benign URLs from Alexa. Half of them were found offline, so they were not analyzed
- The final dataset after the scraping phase takes the form D = [X1, X2, ..., Xn], where Xi is the list of URLs contained within the Xi URL, in the form Xi = [u1, u2, ..., um]
- The balanced dataset was partitioned into 70% training, 20% test, and 10% validation.

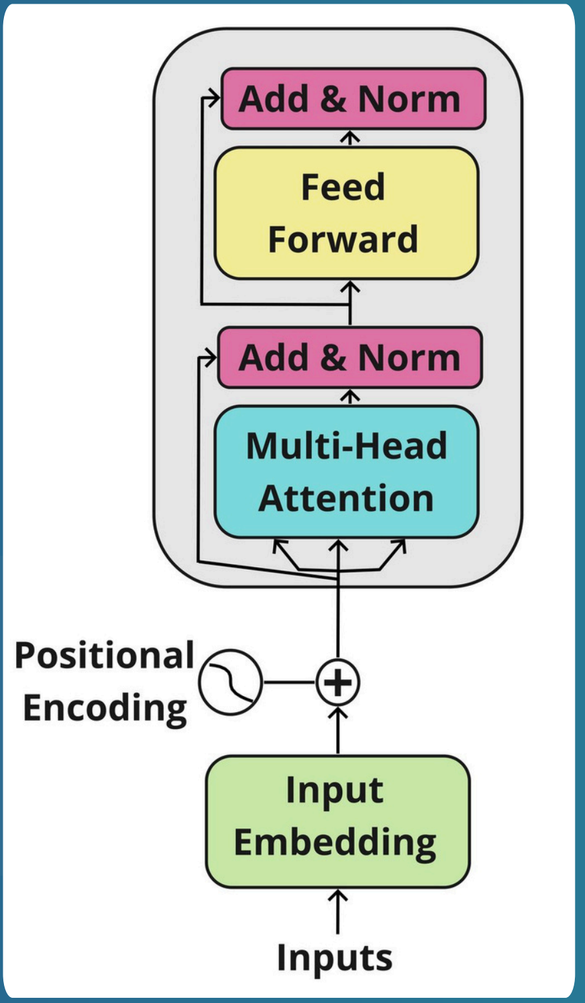


PHISHINGTRANSFORMER

ENCODER

- Each character is associated with its position through a positional encoder.
- Each vector embedding is multiplied by three randomly initialized weights, called Query (Q), Key (K), and Value (V).
- The self-attention layer calculates a score for each character relative to the current character using the following formula

$$Z(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



PHISHINGTRANSFORMER PERFORMANCE

- The model was compared with ML and DL models, as well as with models similar to it. URLNET is a CNN based on feature extraction at three different levels, while MPURNN is a biLSTM that focuses on character-level features.

Model	Precision	Recall	F1 Score
SVM	0.84	0.85	0.84
CNN	0.98	0.97	0.98
BiLSTM	0.95	0.96	0.95
Our model	0.99	0.99	0.99

Model	Precision	Recall	F1 Score
URLNET [14]	0.93	0.85	0.87
LSTM + one hot encoding [29]	0.80	0.82	0.80
Our model	0.99	0.99	0.99

University of Pisa

MODEL COMPARISON

- All the proposed models achieve excellent results, with EPDB and the Transformer coming close to perfection, while the GAN performs slightly less well but still delivers excellent results
- However, it's important to note the significant differences in the sizes of the three datasets

Model name	Accuracy	Precision	Recall	F1 score
EPDB	99.36	98.87	100.00	99.43
PDGAN	97.58	98.02	97.27	97.64
PhishingTransformer	98.90	98.80	99.00	98.80

CONSIDERATIONS

Difference in dataset dimensions

- EPDB : 11,055 records
- PhishingTransformer : 50,000 records
- PDGAN : 2.3 million records

Execution times

- EPDB : 4 seconds
- PhishingTransformer ?
- PDGAN ?

Hyperparameter Optimization

- EPDB : GridSearchCV
- PhishingTransformer : defaut values
- PDGAN : parameter variations

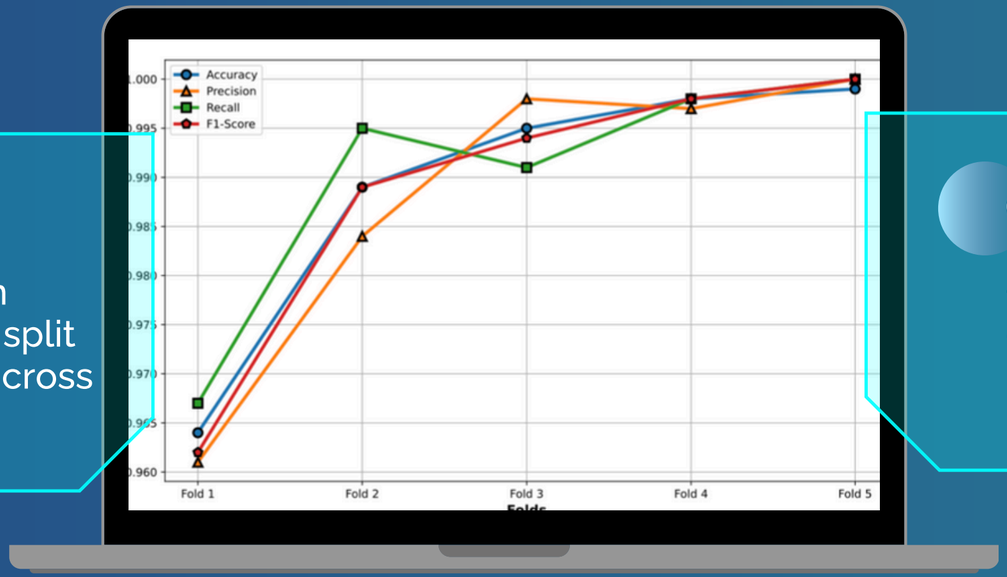
Approach

- EPDB : URL based
- PhishingTransformer : hybrid
- PDGAN : URL based

PERSONAL TIPS

Dataset Splitting

EPDB : 5-fold cross validation
PDGAN : Train-Validate-Test split
PhishingTransformer : 5-fold cross validation ?



Generator usage in PDGAN

+Data augmentation
-Possible overfitting

University of Pisa

FUTURE IMPROVEMENTS

- Increase dataset size for EPDB and PhishingTransformer
- Explore continuous or federated learning techniques
- Create an ensemble method with all the 3 models
- Add the ensemble model to EPDB browser

University of Pisa

THANK YOU!

matteogiorgi196@gmail.com

https://github.com/mgiorgi13/AI-strategies-against-phishing