UNIVERSITÀ DEGLI STUDI DI PADOVA

FALL OF GIANTS: HOW POPULAR TEXT-BASED MLAAS FALL AGAINST A SIMPLE EVASION ATTACK

Authors: Luca Pajola, Mauro Conti

SPRITZ
SECURITY & PRIVACY RESEARCH GROUP

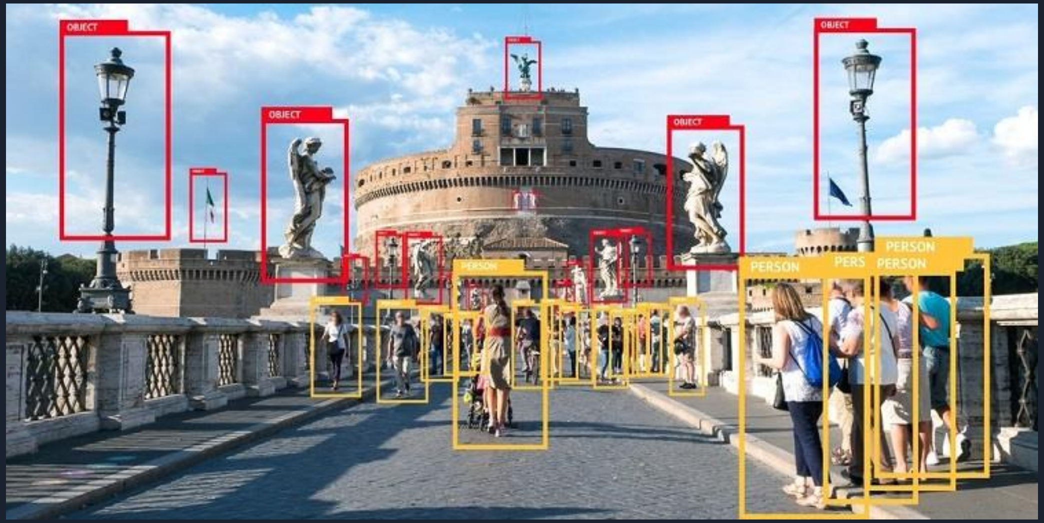
OUTLINE

1. Motivations
2. Zero-Width Attack (ZeW)
3. Results
   - Controlled Environment
   - Into the "wild"
4. Discussions

MOTIVATIONS

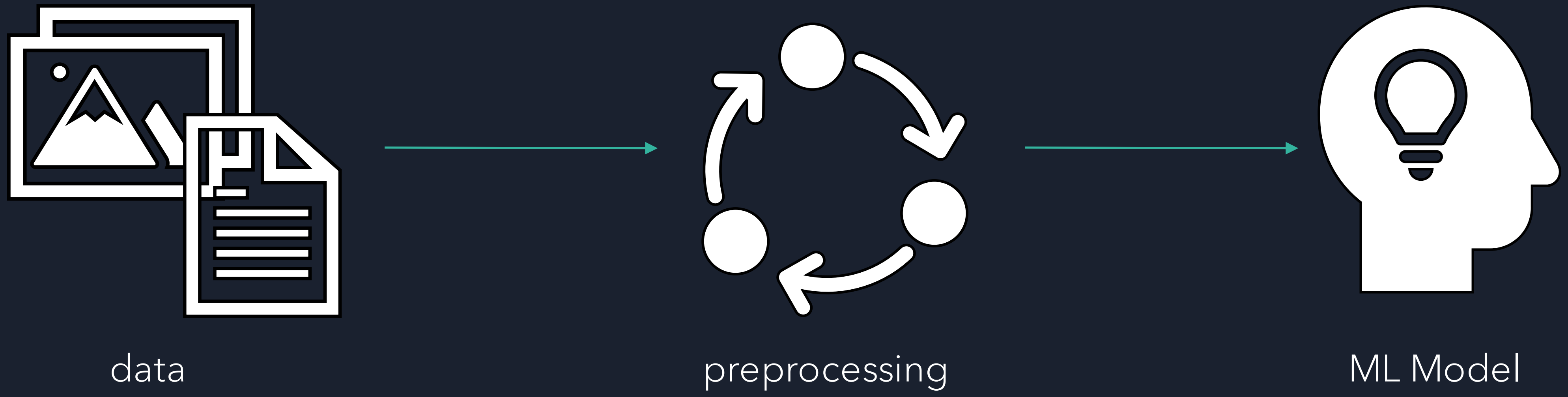
MOTIVATIONS

1. Machine Learning (ML) is here
   - Wide set of ML-based applications are already deployed
2. Several Commercial Usages
3. Gorgeous performance, but what about the ***security***?



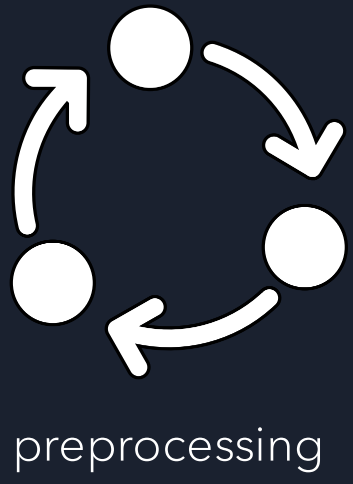
MOTIVATIONS

- Where should we focus?



MOTIVATIONS

- Most attacks are designed to leverage ***ML models weaknesses***
- But preprocessing algorithms plays a ***foundamental*** role in the pipeline
- They are the "foundaments" of our applications
- If an attacker affects these techniques ...



MOTIVATIONS

- Example of image scaling attack [1]
    - The attack affects image scaling techniques applied during the preprocessing
- What about NLP?

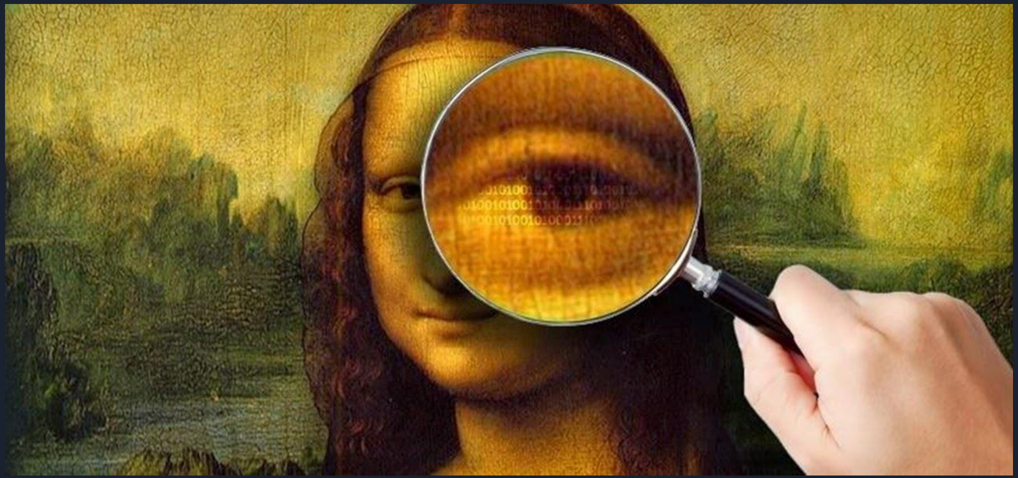
What you see

What your model actually sees

ZERO-WIDTH ATTACK

ZEW – THE IDEA

- Steganography leverages "unnoticeable" characters
  - Among these we find *non-printable characters*
- If inserted inside text, we might affect pre-processing techniques in several ways



ZEW – NLP CHALLENGES

- NLP challenges compared to CV
  1. Input domain
     - Different type of perturbation
     - i.e., in CV we add RGB masks, in NLP?
  2. Human perception
     - Perturbation are easier to spot
  3. Semantic
     - The perturbations should not alter the sentence meaning
     - e.g., I hate you -> I ate you

ZEW – EFFECT

- Word-based models
    - Words with ZeW chars becomes *unknown*
        - And maybe discarded
    - E.g., "I lo$ve you"
        - With unk: "I UNK you"
        - Without unk: "I you"
- Character-based models (more resistant)
    - ZeW characters becomes *unknown*
        - With unk: "I loUNKve you"
        - Without unk: "I love you"

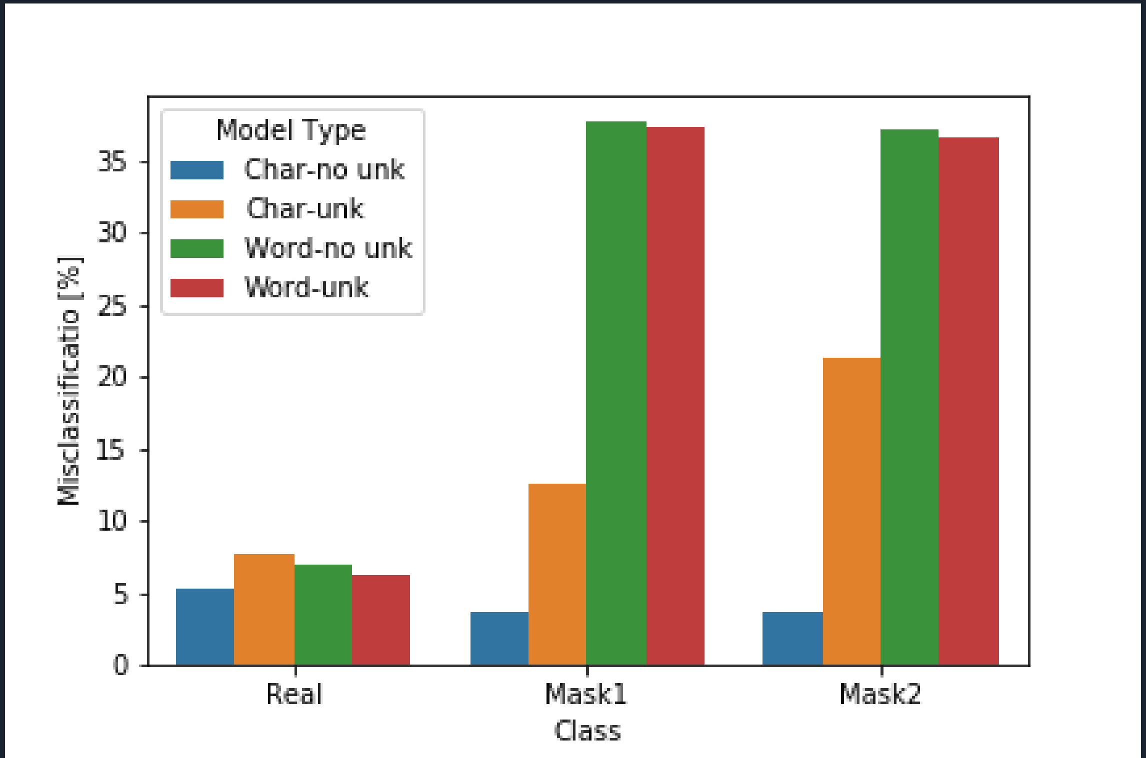
RESULTS

RESULTS – ALGORITHM

- Case Study: Hate Speech Evasion
- Algorithm
  - Identification of negative words in a given sentence
  - Add ZeW characters inside the words
- Two injection strategies
  - *Mask1*: insertion on the middle of the word
    - Hate -> ha$te
  - *Mask2:* insertion in between each word
    - Hate -> hate$

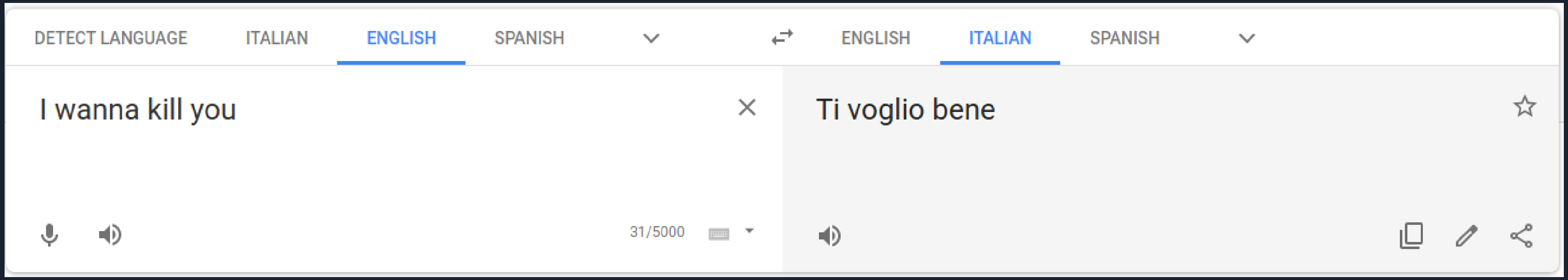
RESULTS – CONTROLLED ENVIRONMENT

- RNN model: GRU
- Representation type: char and word
- With and without UNK tokens
- Dataset: Sentiment140 dataset [3]
- Goal: evasion of negative sentences



RESULTS – INTO THE WILD

- Tested 12 API
  - Developed by Amazon, Google, Microsoft, and IBM
  - Different type of services (e.g., translators, sentiment analyzers)
- Goal: manipulate outcomes of hate-speech analyses



RESULTS – INTO THE WILD

(a) Amazon Comprehend.

(b) Google Cloud Natural Language.

(c) IBM Watson Natural Language Understanding.

(d) Microsoft Text Analytics.

DISCUSSIONS

DISCUSSIONS

- A simple sanitification techniques might prevent ZeW
    - First rule in cybersecurity: don't trust the input!
    - UNICODE contains a lot of characters
- Preprocessing techniques are perfect attack vectors
    - ML applciations do not only contain ML models!
- The attack works in real-life applications
    - We should be more carefull on what we deploy

THANK YOU

REFERENCES

[1] Xiao, Qixue, et al. "Seeing is not believing: Camouflage attacks on image scaling algorithms." USENIX Security (2019).

[2] Hutto, Clayton, and Eric Gilbert. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." Proceedings of the International AAAI Conference on Web and Social Media. Vol. 8. No. 1. 2014.

[3] A. Go, R. Bhayani, and L. Huang. (2009) Twitter sentiment classification using distant supervision.