NORCICS SFI Norwegian Centre for Cybersecurity in Critical Sectors

NTNU Center for Cyber and Information Security

Edinburgh Napier UNIVERSITY

“Can we trust AI?”

Dr. Nikolaos (Nick) Pitropakis
Associate Professor of Cybersecurity
n.pitropakis@napier.ac.uk

Presentation at: Norwegian Centre for Cybersecurity in Critical Sectors
Centre for Cyber and Information Security
(NORCICS CCIS 2024)

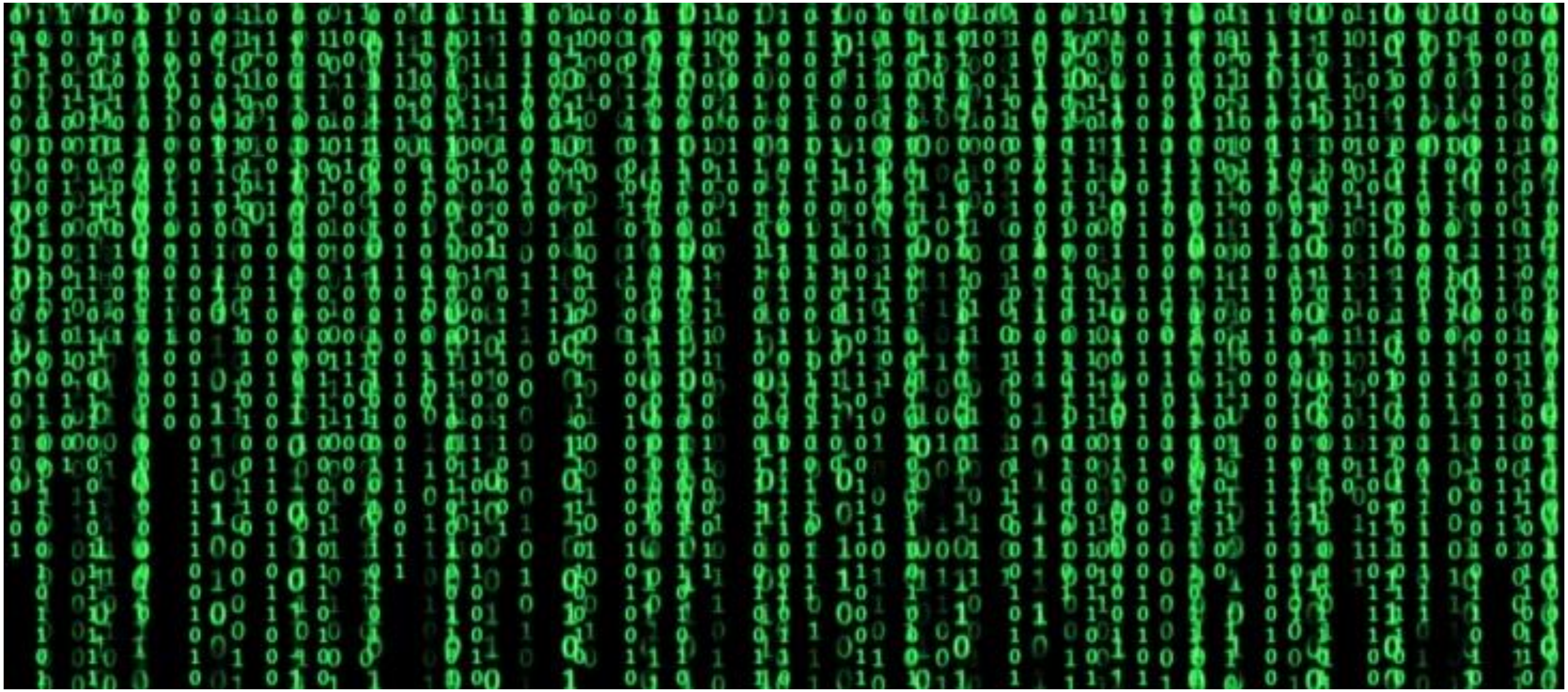
November 2024

Contents

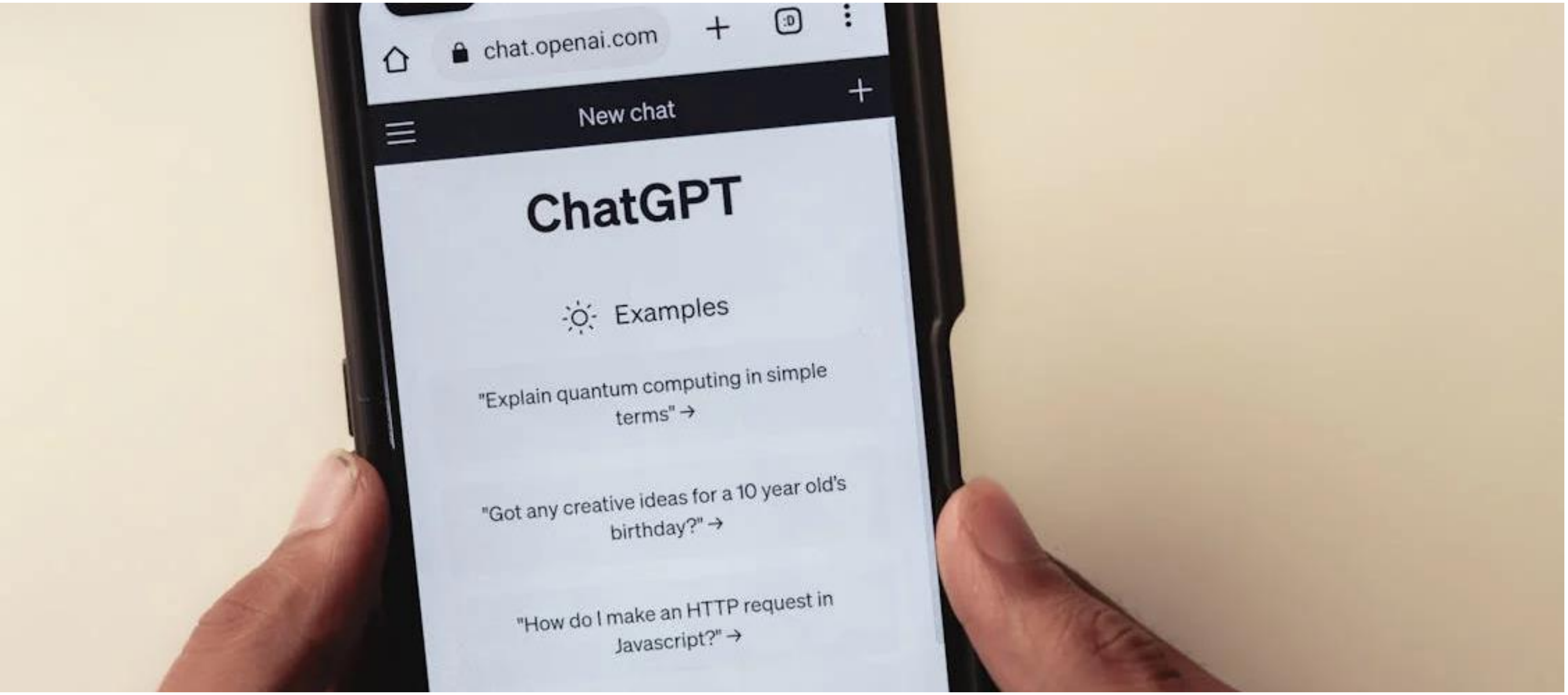
- What is AI/Machine Learning?
- Threat Landscape
- Use Case IoT IDS
- Use Case Malicious URL Detection
- Use Case Tweets Classification
- Use Case Financial Time Series Models
- Conclusions

What is AI?

AI



AI



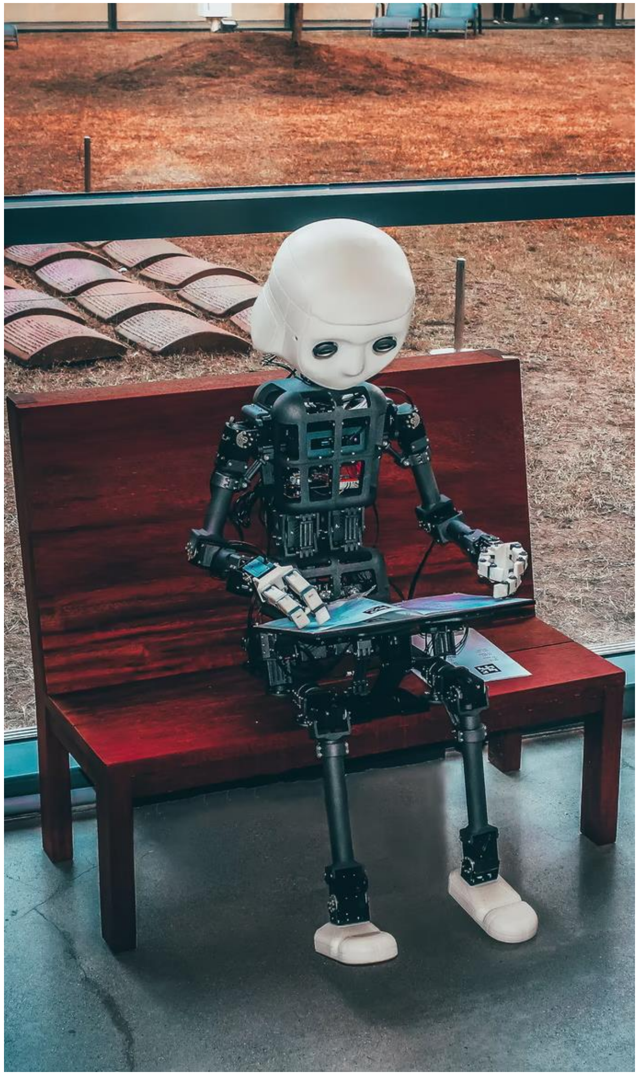
"Explain quantum computing in simple terms" →

"Got any creative ideas for a 10 year old's birthday?" →

"How do I make an HTTP request in Javascript?" →

AI

What is Machine Learning?

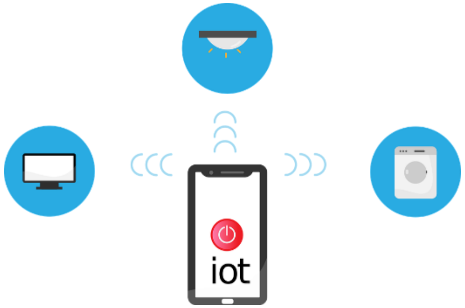


Machine Learning

Threat Landscape

- **The Assumption:**
  - All ML systems are trained using datasets that are assumed to be **representative** and **trustworthy** for the subject matter in question thus enabling the construction of a valid system perception of the phenomenon of interest.
- **Malicious actors** can impact the decision-making algorithms by:
  - either targeting the training data or the model (**poisoning**)
  - forcing the model to their desired output (**evasion**)
- **Adversaries** can significantly decrease overall performance, cause targeted misclassification or bad behaviour.
- **Adversarial Machine Learning (AML)** is the study of effective machine learning techniques against an adversarial opponent.

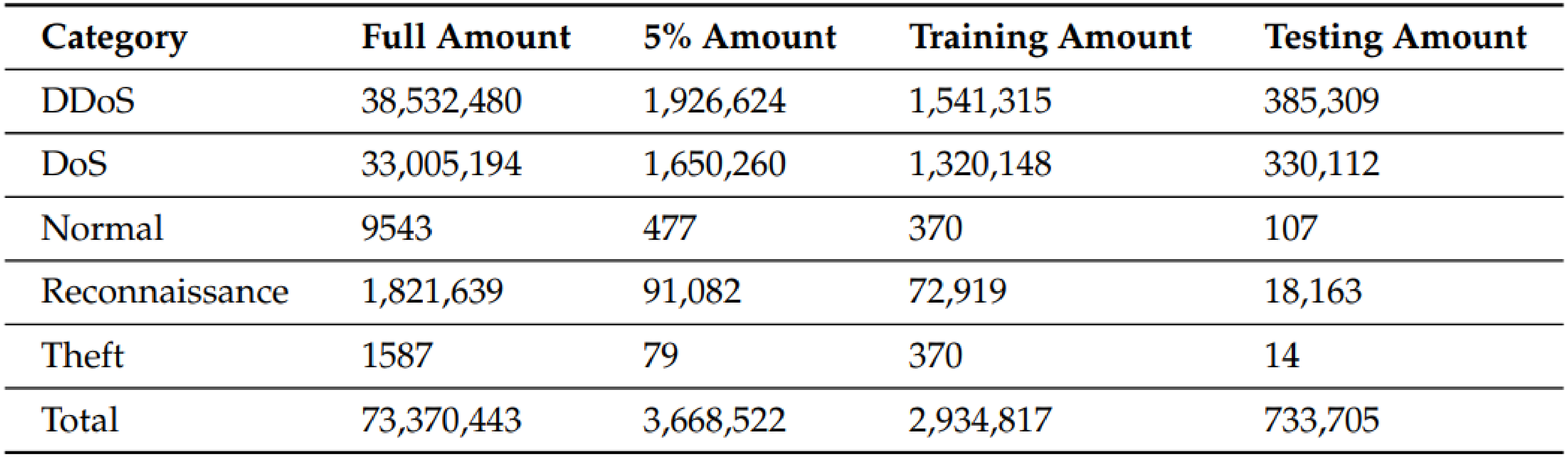
Use Case – IoT IDS



- The internet is shifting towards increasing the connectivity of physical devices, called the **Internet of Things (IoT).**
- The **security standards** for IoT devices are not defined and often become an after-thought in a profit-driven market.
- Gartner predicts that by 2023 there will be **40 billion IoT devices**.
- Due to the nature of limited resources within an IoT device, **IDS** are often placed at the network perimeter to detect adversarial activity.
- **Machine learning-based IDS** solutions are vulnerable when the model is targeted and exploited by adversaries.

Use Case – IoT IDS

- ❑ The **Bot-IoT dataset** was created in a simulated environment formed up of victim and attacking machines.
- ❑ It has not knowingly been created using adversarial examples towards the machine learning model.
- ❑ The different attacks are categorised into five classes; (1) Normal, (2) Reconnaissance, (3) DDoS, (4) DoS, (5) Information Theft.



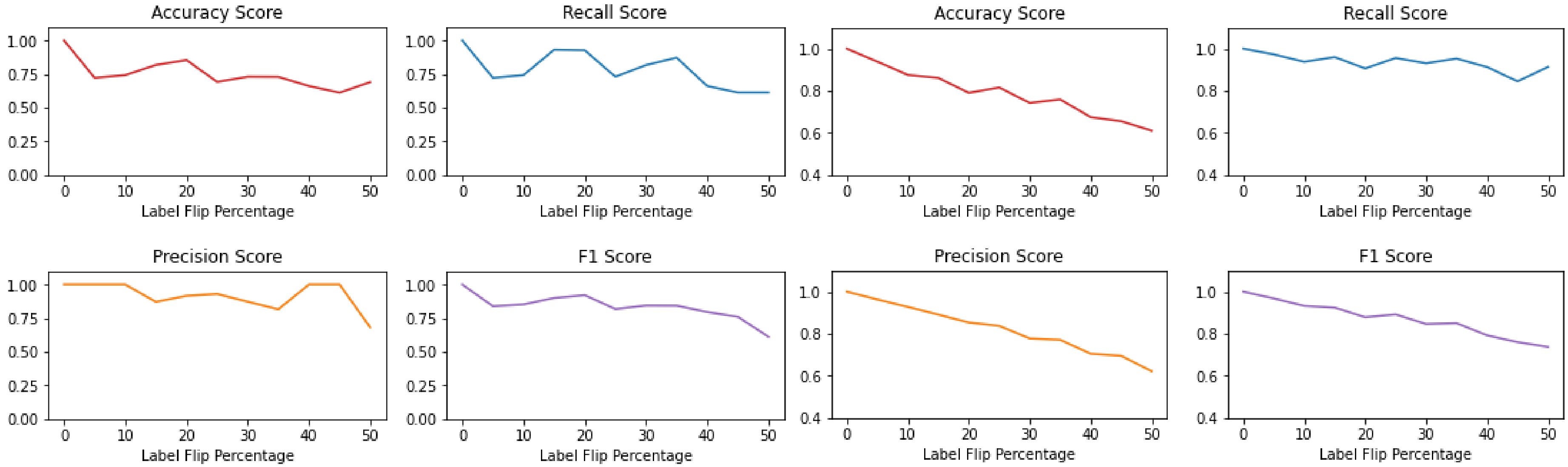
Category	**Full Amount**	**5% Amount**	**Training Amount**	**Testing Amount**
DDoS	38,532,480	1,926,624	1,541,315	385,309
DoS	33,005,194	1,650,260	1,320,148	330,112
Normal	9543	477	370	107
Reconnaissance	1,821,639	91,082	72,919	18,163
Theft	1587	79	370	14
Total	73,370,443	3,668,522	2,934,817	733,705

Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet ofthings for network forensic analytics: Bot-iot dataset. Future Gener. Comput. Syst. 2019, 100, 779–796.

Use Case – IoT IDS

- We split the 5% extracted records into training and testing splits, sized **80% and 20%** respectively.
- We used:
  - A Support vector machine (**SVM**) model which is a Support Vector Classifier (SVC) using the linear kernel.
  - A trained Artificial Neural Network (**ANN**) that had one input layer, three intermittent layers and one output layer.
- We attacked:
  - SVM using **label flipping**.
  - ANN using Fast Gradient Sign Method (**FGSM**)

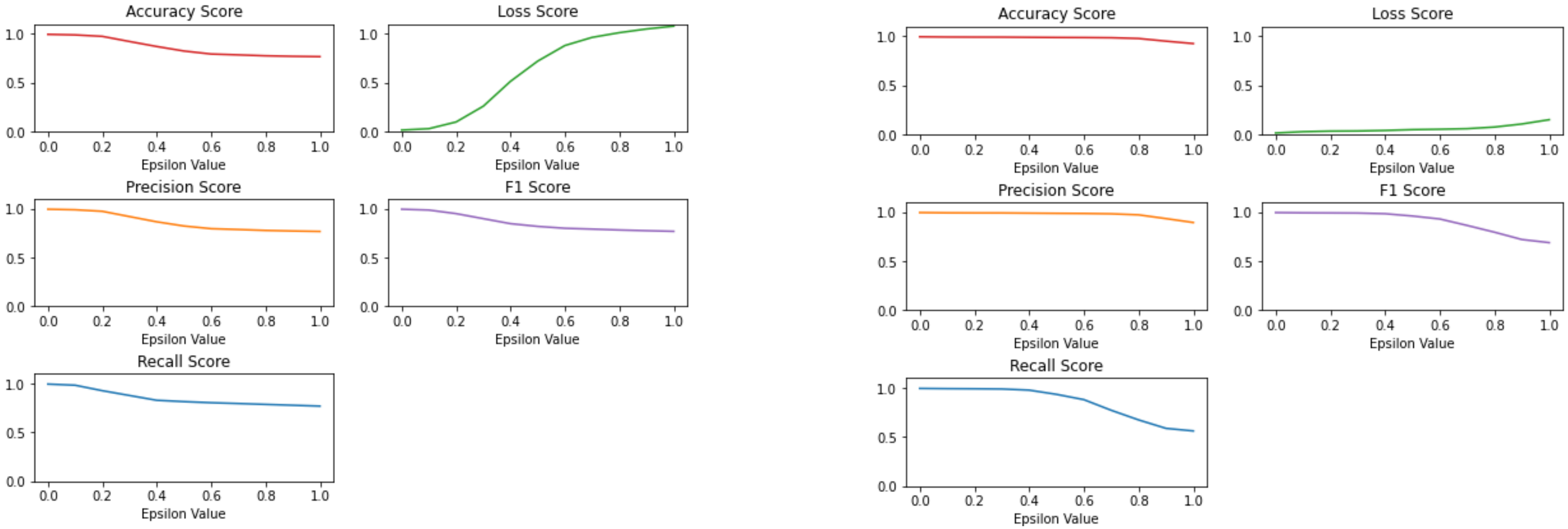
Use Case – IoT IDS



(a) Targeted Support Vector Machines flip metrics. (b) Non-targeted Support Vector Machines flip metrics.

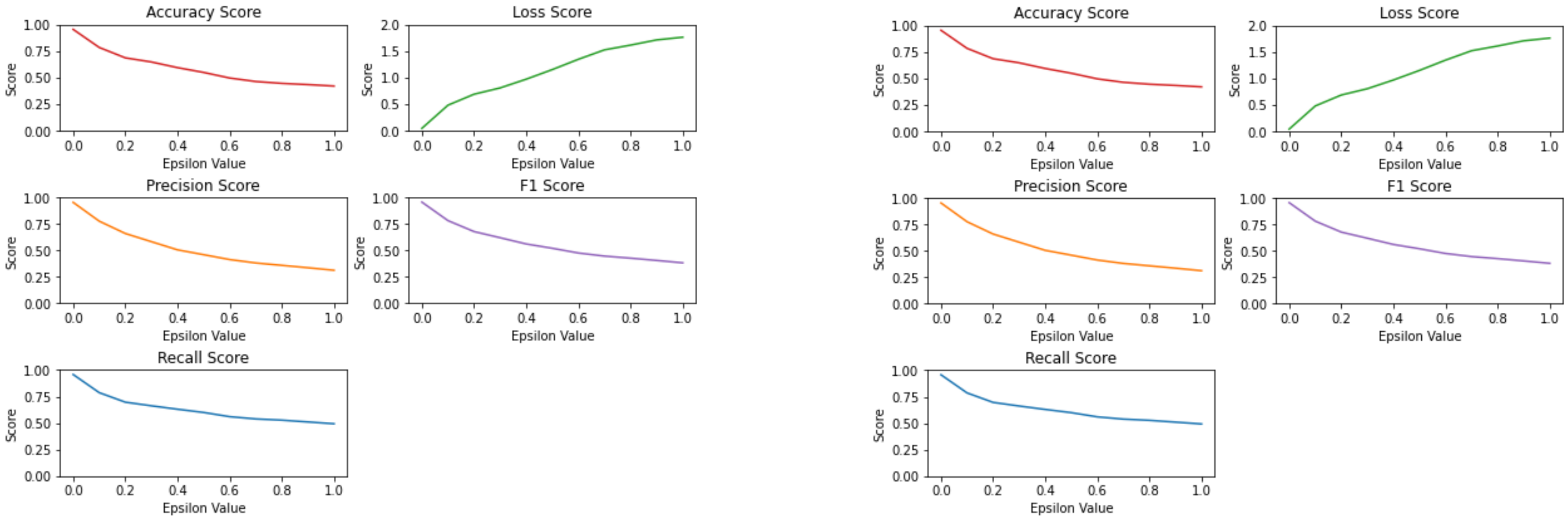
Use Case – IoT IDS

- The activities to create adversarial models using the FGSM are the same for the binary and the five-class multi-classification ANNs. The difference between the binary and five-class multi-classification lies in the feature column



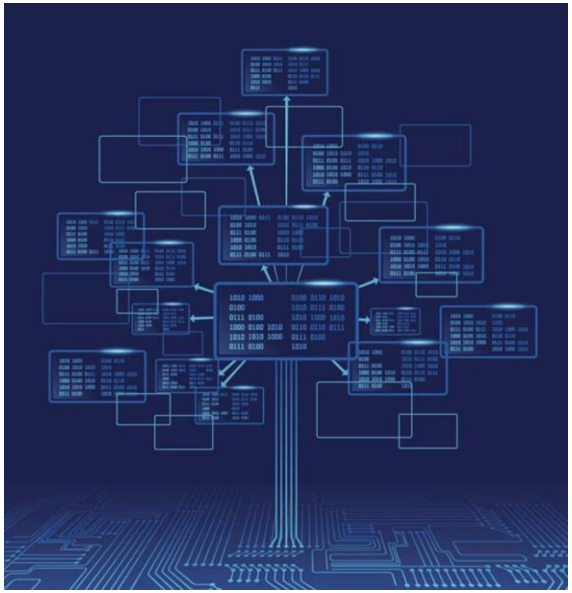
(a) Binary classification metrics, non-targeted Fast Gradient Sign Method. (b) Binary classification metrics, targeted Fast Gradient Sign Method.

Use Case – IoT IDS



(a) Multi-classification metrics, targeted Fast Gradient Sign Method. (b) Multiclassification metrics, non-targeted Fast Gradient Sign Method.

Use Case Malicious URL Detection

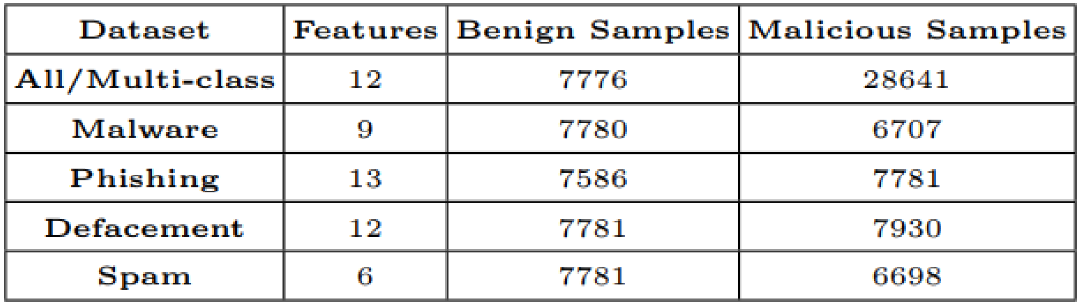


- **Traditional detection methods** are costly and reactive:
  - Human verification
  - Allow and block lists
- **Machine Learning (ML)** can have high detection accuracy
  - Trained models classify URLs
  - Performance is likely to be severely degraded by even mild perturbations
- **To be effective ML requires**
  - Voluminous high-quality data
  - Well engineered features

Use Case Malicious URL Detection

- Mamun, et al. (2016) trained the algorithms Random Forest, C4.5, and k-Nearest Neighbour (KNN) to classify URLs in a supervised learning scenario.

- Training and testing data was comprised of four single-class and one multi-class dataset:

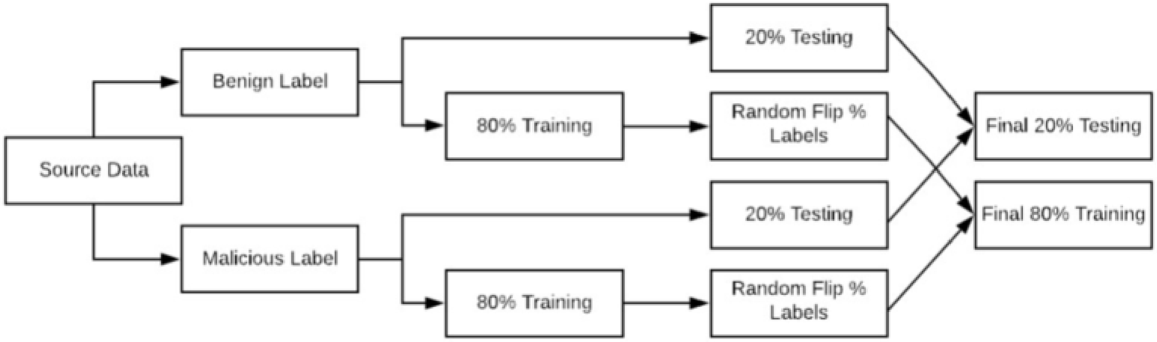


Dataset	Features	Benign Samples	Malicious Samples
All/Multi-class	12	7776	28641
Malware	9	7780	6707
Phishing	13	7586	7781
Defacement	12	7781	7930
Spam	6	7781	6698

- Although this work produced high-accuracy models it did not account for any adversarial conditions.

Bursztein, E. "Inside Mirai the infamous iot botnet: A retrospective analysis" (2017).

Use Case Malicious URL Detection

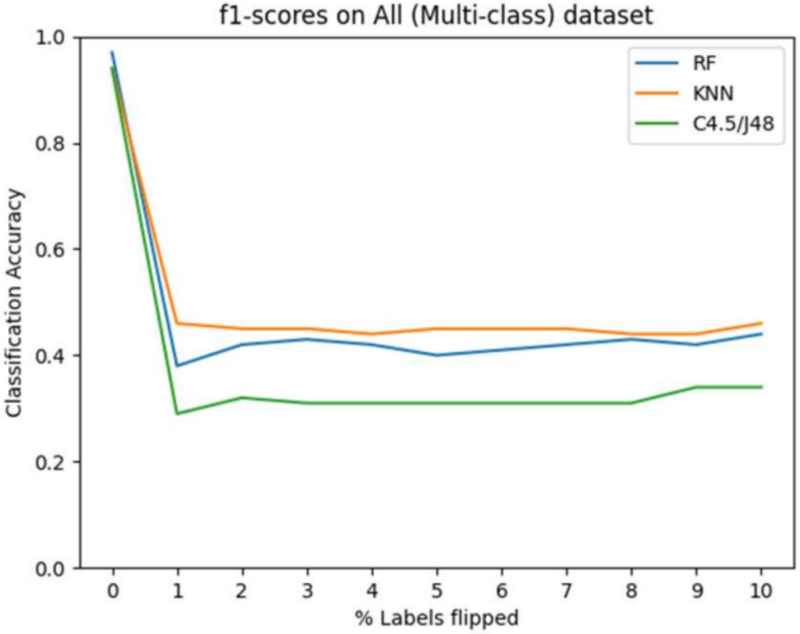


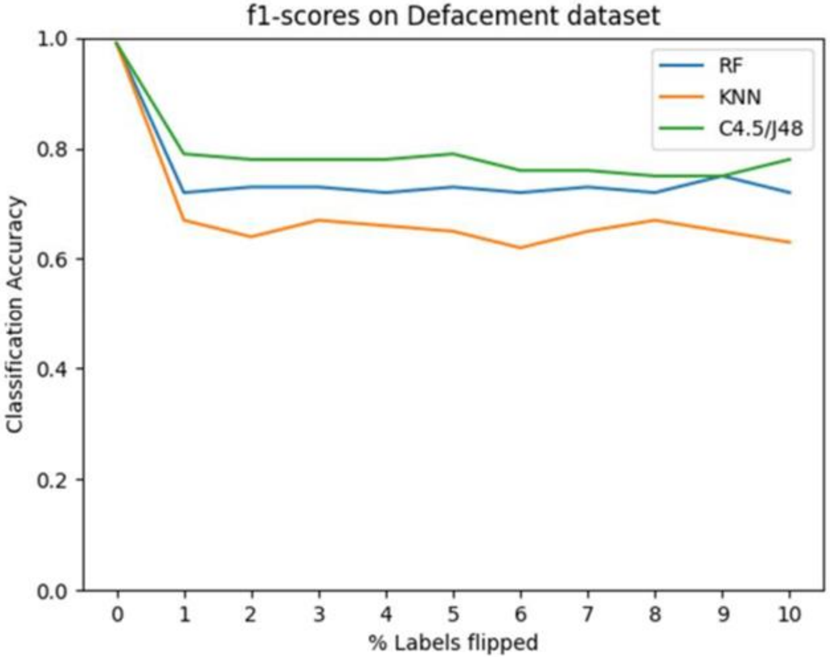
- ❑ **Stage 1:** Separate the data points according to whether they are labelled benign or malicious.
- ❑ **Stage 2:** Divide each of these sets again, this time into what will be training data (80%) and testing data (20%).
- ❑ **Stage 3:** Randomly flip a percentage of labels in both the malicious and benign training data. The testing data remains ‘clean’.
- ❑ **Stage 4:** Recombine the testing and the now perturbed training data.

Each target model was then trained on the poisoned training data at each percentage step and tested on the unperturbed test data.

Use Case Malicious URL Detection

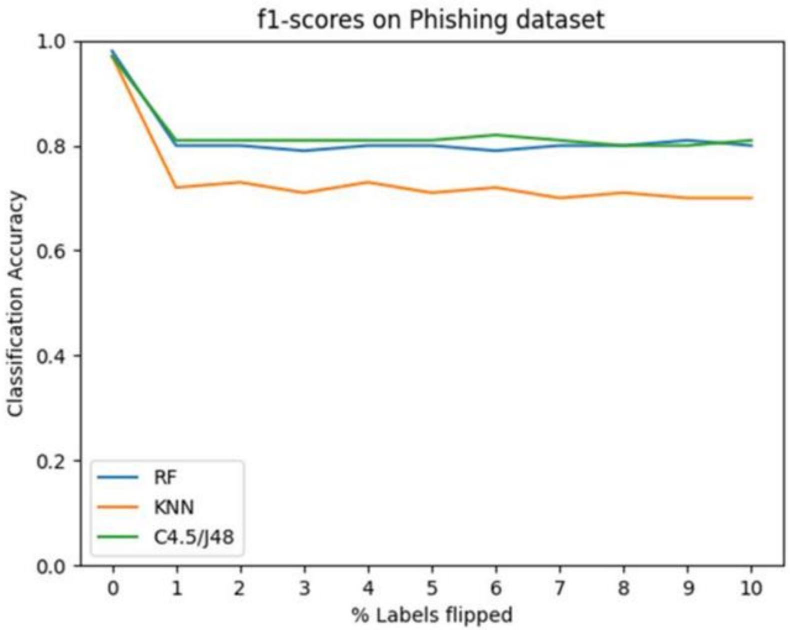
- **Attack observations:**

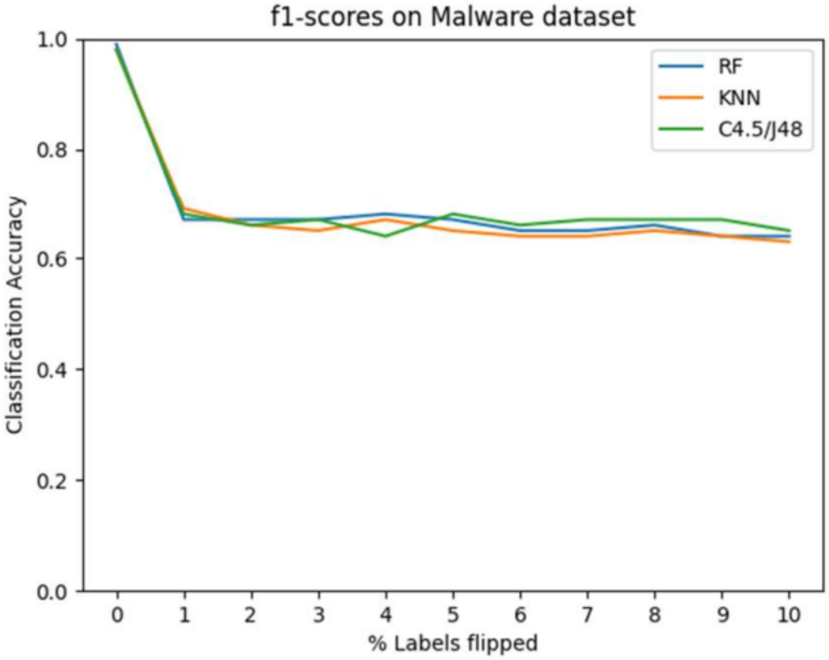




Use Case Malicious URL Detection

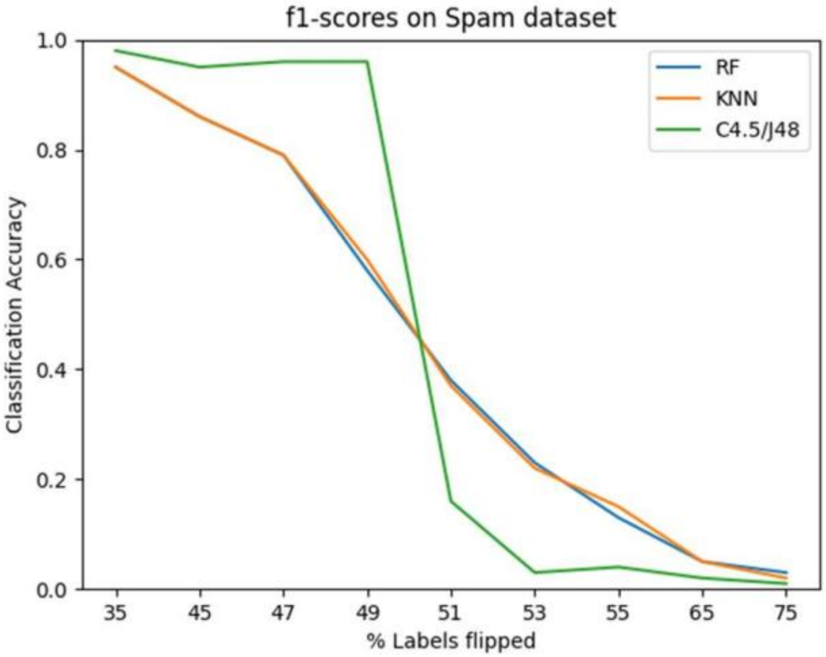
- **Attack observations:**





Use Case Malicious URL Detection

- **Attack observations:**



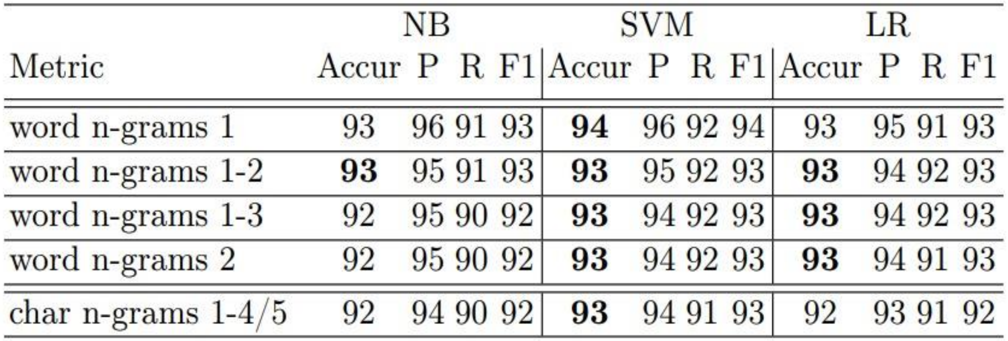
Use Case Tweets Classification

Homes for Ukraine Policy

- ❑ Russia's invasion to Ukraine resulted in the fastest growing refugee crisis since World War II as 2.5 million fled the country in the subsequent two weeks. The government's policy encouraging people to house displaced refugees was being discussed on **X (Twitter)** and trailed in the mainstream press.
- ❑ The initiative encouraged sponsors - organisations and also members of the public - to accommodate Ukrainian refugees for a minimum of six months. In addition, £10,500 would be allocated to local councils (per refugee) to support the resettlement.
- ❑ The scheme appeared to originate from a position of both empathy and compassion. The reasoning adopted here was that tweets approving of the policy would have displayed empathy towards those forced to flee their homes, and also compassion by showing either a willingness to help **directly** (by becoming a sponsor themselves) or **indirectly** (by being in favour of the appropriation of public funds to the scheme).
- ❑ Conversely, **tweets** disapproving of the policy would have displayed either a lack of empathy (i.e. an inability to put oneself in the position of having to flee their home), a lack of compassion or both.
- ❑ We crawled Twitter for messages related to the Homes for Ukraine scheme, a resulting corpus of 32,000 and carefully annotated the resulting tweet data.

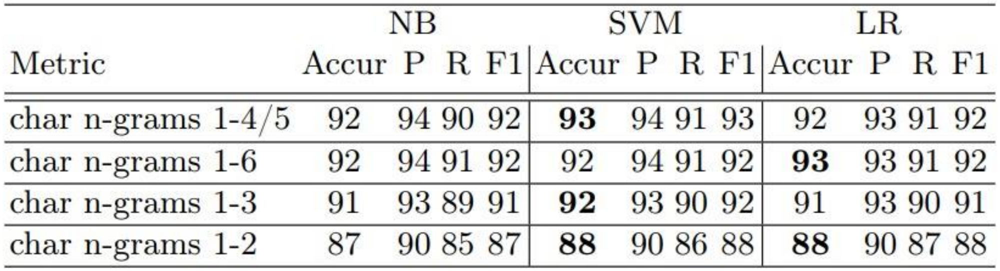
Use Case Tweets Classification

- ✓ **In bold,** the highest accuracies achieved by **NB, SVM and LR** classification algorithms for the different variations of n-grams.
- ✓ Word n-grams returned **higher accuracies** than character n-grams, with unigrams achieving the max accuracy value (i.e., 94).
- ✓ Adding bigrams and trigrams only **degraded** the performance of the model, but metrics were still marginally higher than for bigrams alone.
- ✓ In terms of classification algorithms, **SVM attained the top accuracy** for all six variations of text representation, LR in three of these categories and NB in one. Other flavours of word n-grams such as trigrams and bigram combinations returned slightly lower accuracy metrics.



	NB				SVM				LR			
Metric	Accur	P	R	F1	Accur	P	R	F1	Accur	P	R	F1
word n-grams 1	93	96	91	93	**94**	96	92	94	93	95	91	93
word n-grams 1-2	**93**	95	91	93	**93**	95	92	93	**93**	94	92	93
word n-grams 1-3	92	95	90	92	**93**	94	92	93	**93**	94	92	93
word n-grams 2	92	95	90	92	**93**	94	92	93	**93**	94	91	93
char n-grams 1-4/5	92	94	90	92	**93**	94	91	93	92	93	91	92

Top accuracy, precision, recall and F1 metrics for NB, SVM & LR classifiers

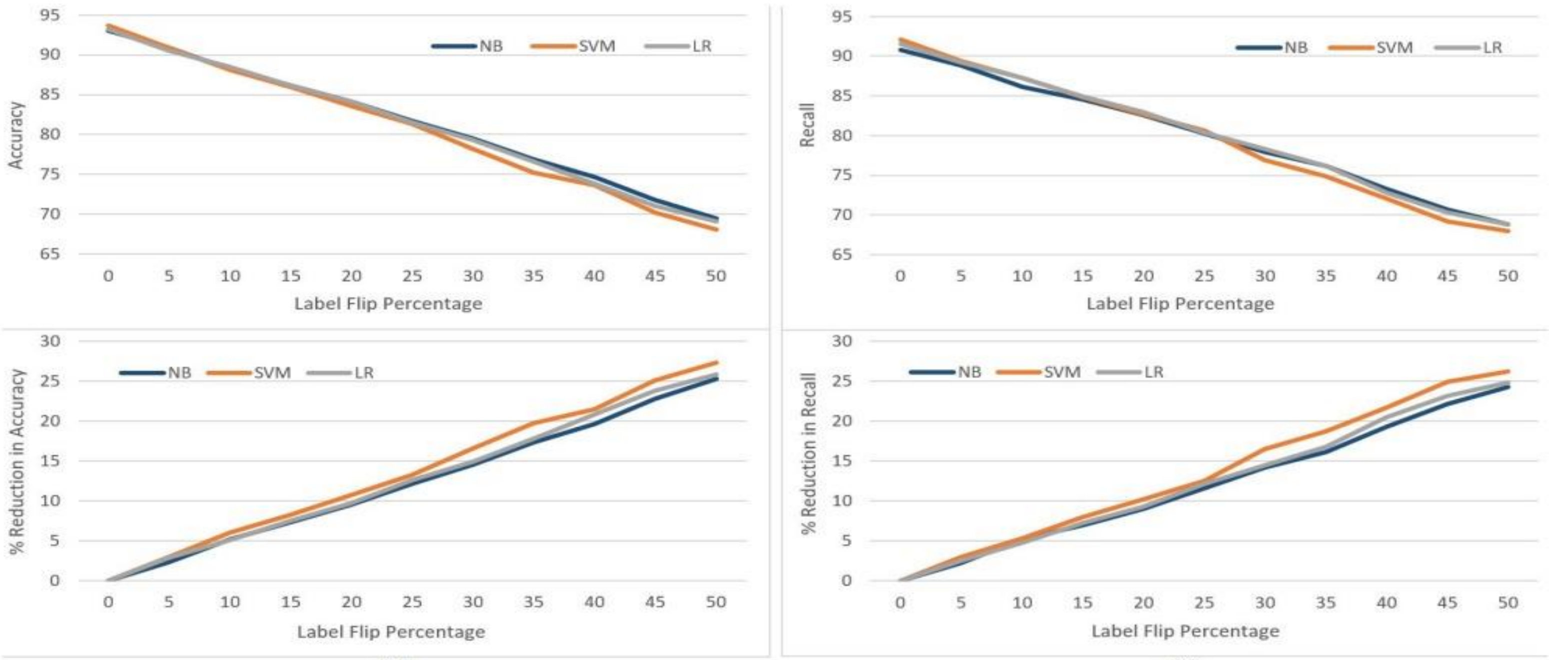


	NB				SVM				LR			
Metric	Accur	P	R	F1	Accur	P	R	F1	Accur	P	R	F1
char n-grams 1-4/5	92	94	90	92	**93**	94	91	93	92	93	91	92
char n-grams 1-6	92	94	91	92	92	94	91	92	**93**	93	91	92
char n-grams 1-3	91	93	89	91	**92**	93	90	92	91	93	90	91
char n-grams 1-2	87	90	85	87	**88**	90	86	88	**88**	90	87	88

Character n-gram metrics for NB, SVM & LR classifiers

Use Case Tweets Classification

- ✓ **(a)** The reduction in accuracy metrics for the three classification algorithms as the percentage of randomly-poisoned observation labels was increased. Their relationship is linear, with each percentage increase in flipped labels equating to a half-point reduction in accuracy. The SVM classifier proved the most sensitive to label poisoning, starting as the most accurate before poisoning, but dropping to the least accurate after 10% of observations are poisoned. By contrast, **the NB model**, starting as the least accurate, is seen to be the **most robust** to the adversarial attack.
- ✓ **(b)** A similar pattern in terms of recall; low recall values correspond to high instances of false negatives. SVM shows greater degradation, dropping from the highest to the lowest recall figure after 30% of poisoned data.



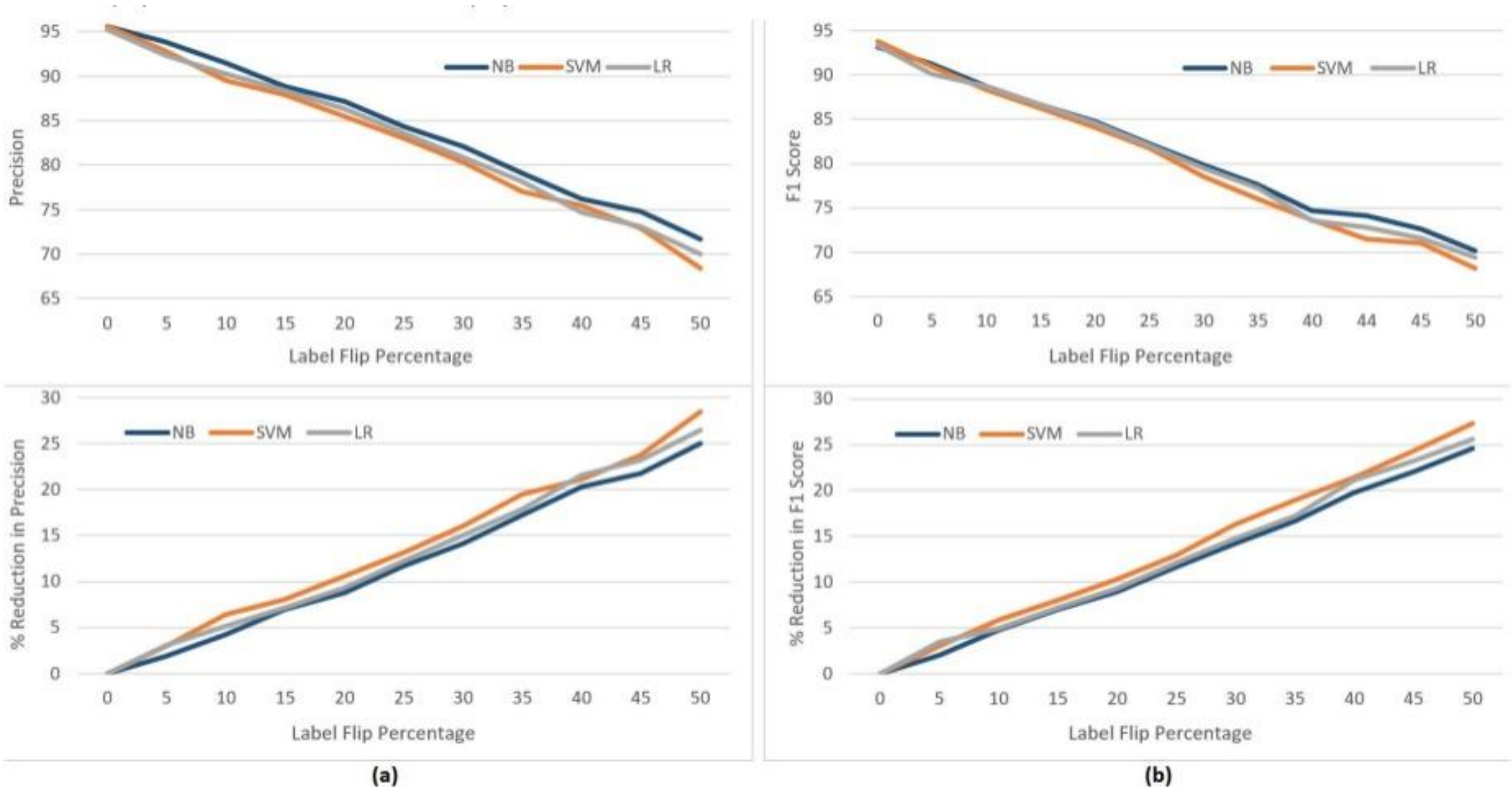
(a)

(b)

(a) Accuracy and (b) recall metrics for increasing levels of label poisoning

Use Case Tweets Classification

- ✓ The precision metrics in **(a)** indicate that the NB model reacts better to adversarial attacks than the other classifiers, while the SVM model is affected the most.
- ✓ Not surprisingly, since F1 is a balanced metric of precision and recall, results in **(b)** are similar.



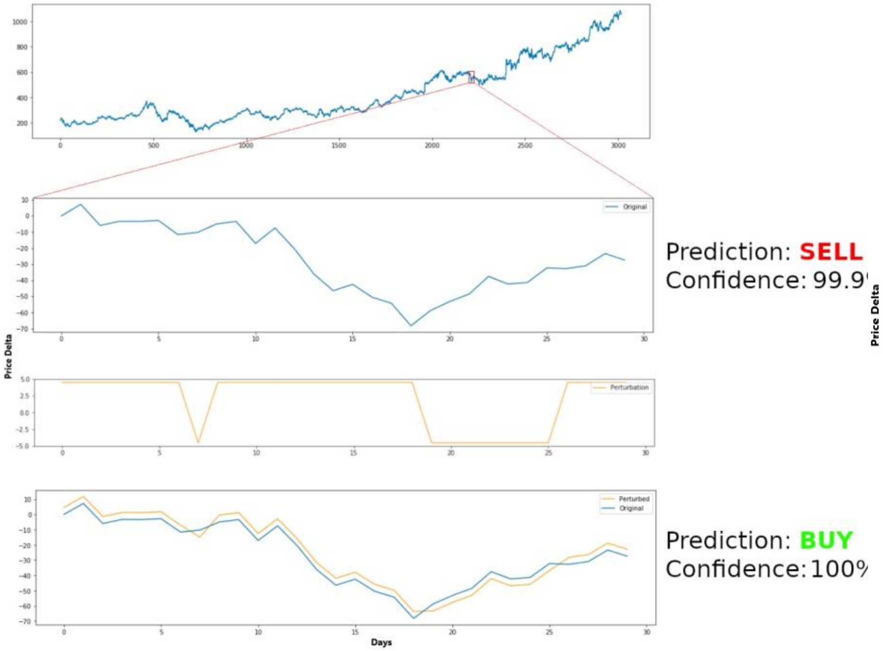
(a) Precision and (b) F1 score for increasing levels of label poisoning

Use Case Financial Time Series Models

- **Financial time series models are known for:**
  - being a major influential factor in the global economy
  - their non-linear, nonstationary and noisy natureTo be effective ML requires

- Financial stock data was obtained from **Kaggle (2017)**, and daily stock price data for Google stock was used. The time range was 2006–2018.
- We used 1-Dimensional Convolutional Neural Network model, whose training period was 2006–2014 and the remaining data was used for validation.
- The data was processed with a sliding window fixed to 30 days, and the future price prediction offset was fixed to 14 days.
- The data was normalised so that each entry in the time series was the price difference from the previous entry. This is known as the price delta or Δ.

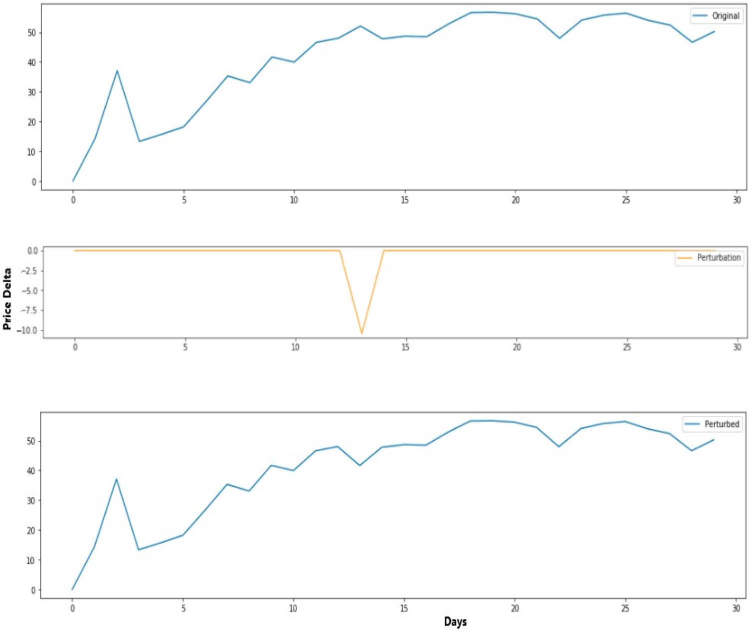
Use Case Financial Time Series Models



Prediction: **SELL**
Confidence: 99.9

Prediction: **BUY**
Confidence: 100%

Sliding Window input and FGSM attack

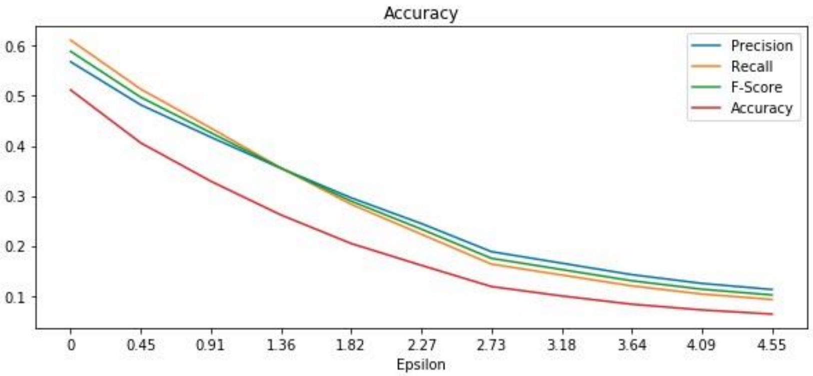


Prediction: **SELL**
Confidence: 92%

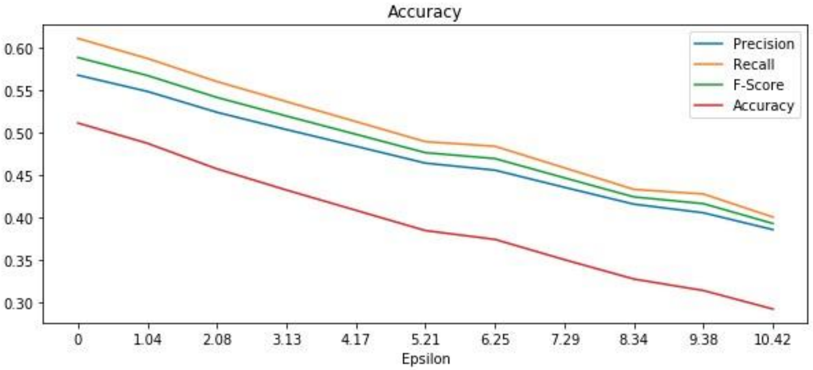
Prediction: **BUY**
Confidence: 97%

Single Value attack example

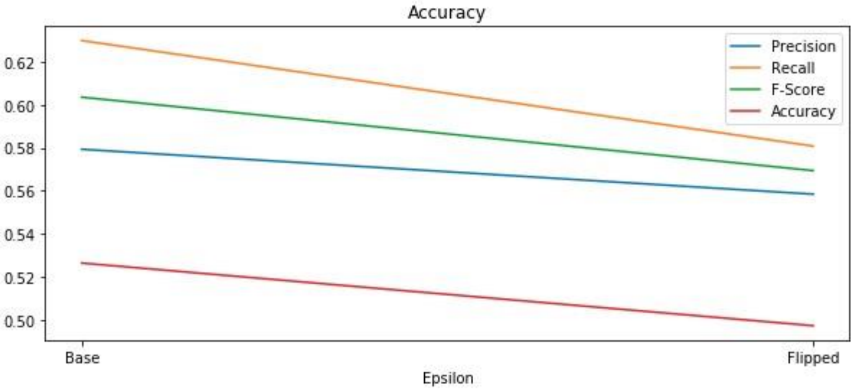
Use Case Financial Time Series Models



FGSM Accuracy

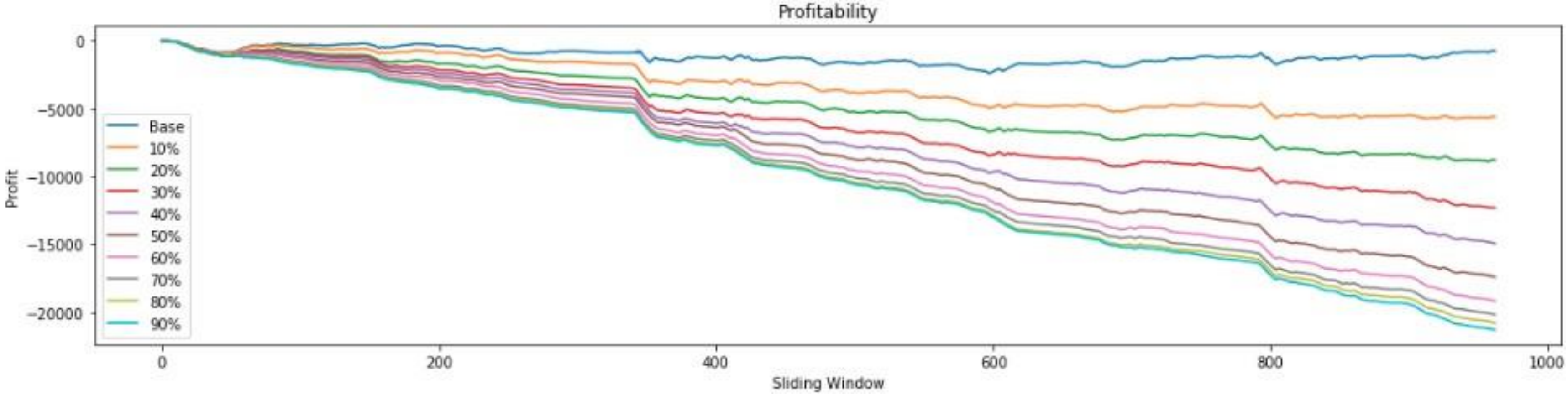


Single Value Accuracy

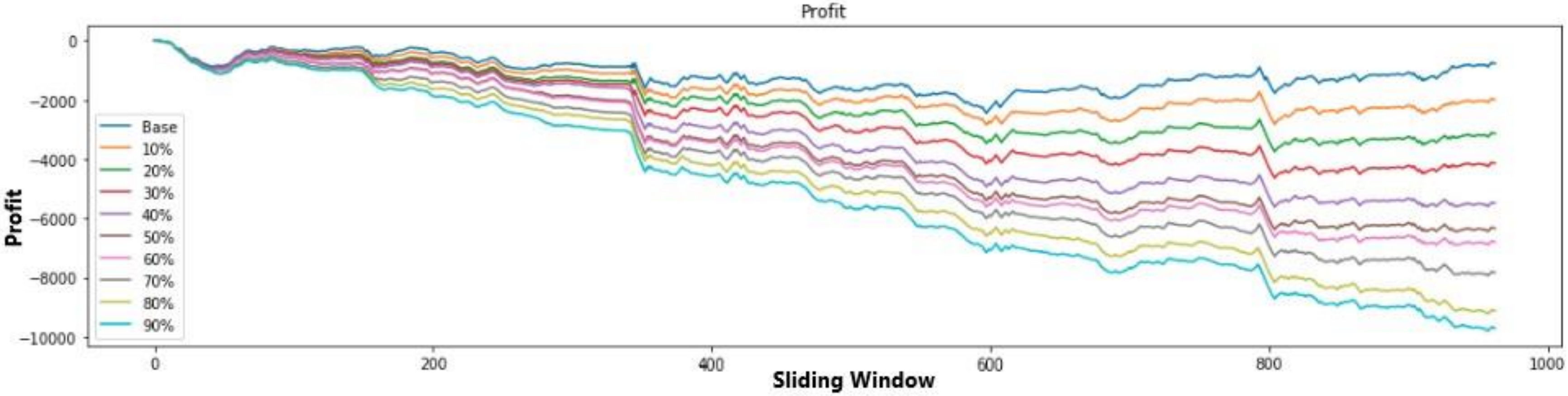


Label Flip Accuracy

Use Case Financial Time Series Models

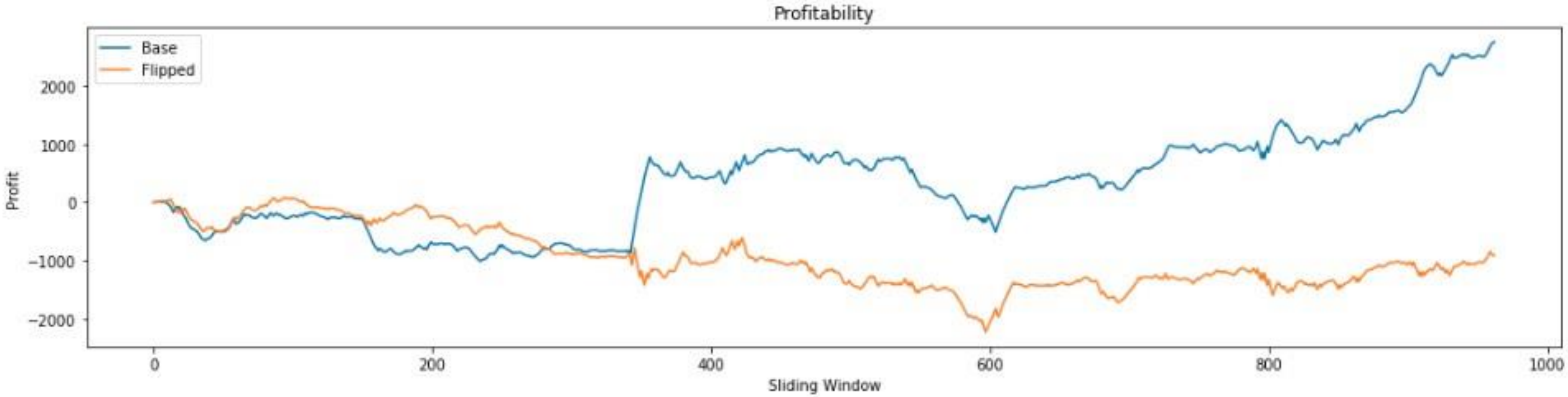


FGSM simulated profit



Single Value Attack simulated profit

Use Case Financial Time Series Models



Label Flip simulated trading performance

- The results demonstrated a significant performance loss in each attack and how this can lead to financial loss when used in financial trading simulations.

- As these are white-box attacks, an attacker would need access to the internals of the neural network to perform them.

- A more finely tuned model could be even more affected by attacks.

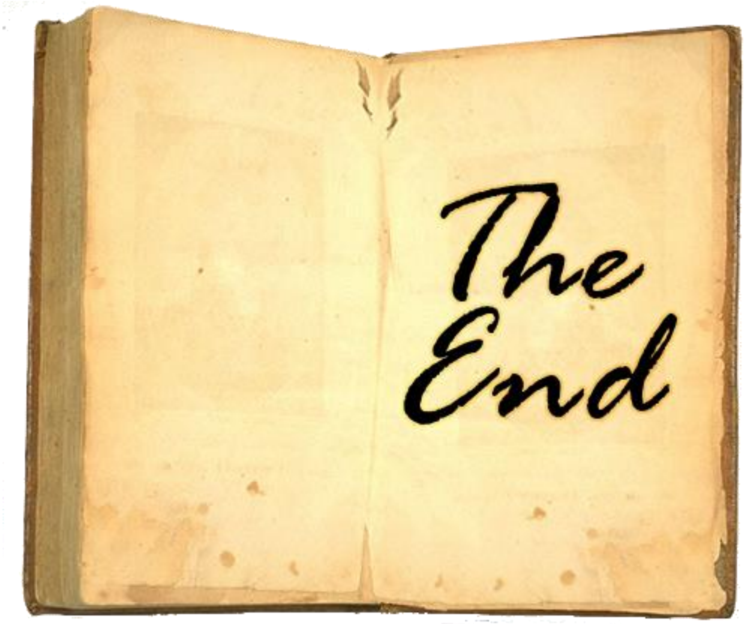
Conclusions

- Machine Learning algorithms rely on the quality of the training data to produce good results.

- A privileged entity can change the perception of the machine about reality and create misclassifications.

- There is a need for mitigation against poisoning and thus a need of defensive techniques.

- **In the future:**
  - we aim to perform more attacks in different datasets and classifiers.
  - to investigate the effects of defensive measures.



Thanks for your attention!