Machine Learning (In) Security: A Stream of Problems (Part 1) - Summary - Sidharth Anil

Presentation

The paper mainly deals with the problems that needs to be overcome when engineering a machine learning model in the context of cybersecurity. This includes problems in each of the phases that the process would go through from selecting a model, acquiring the data, training the model and evaluating the model.

1. In the case of selecting a model, the presenter made it clear that there is no golden model that will work in all instances. Instead careful reasoning with a combination of trial and error is required to identify the requirements expected as well as the model that can deliver it.
2. Next comes the stage of collecting data and the challenges associated with it.
    - Data leakage due to temporal inconsistency is one of the most common pitfalls in this stage. This is because, malware has a constantly adapting data distribution where new malwares are influenced by the past malwares. This implies that temporal information is important in case of malware detection, and if this is ignored it would cause data leakage and lead to models that do not perform well in the real world.
    - Another issue is data labelling, as different AV companies use different labelling styles. Moreover the labels might change over time too.
    - A solution to accomodating for class imbalances is during the data processing stage, where either undersampling or oversampling could be employed. However, care must be taken to preserve the proportion of different samples (at specific times too), in order to make sure that the modified dataset represents the original dataset accurately.
    - The dataset size should also be taken care of. A huge dataset does not necessarily equate to a better model. Instead, priority must be given to a representative dataset. A dataset that is filtered to only a few types of malware samples would only show good accuracy for similar datasets as it would be tuned to perform well for the type of malware it was trained upon. This would mean that if an existing model is to be employed in a different region of the malware space, the model would have to be retrained using new data that is representative of the new region. Also, it can be observed that after a point increasing the amount of data would only increase the performance of the model marginally.
3. From raw data different kinds of attributes can be obtained, the major categorization being static attributes and dyamic attributes. Dynamic attributes would come at a higher cost of extraction, but at the same time could contain more information. The effects of different types of attributes on the model would depend majorly on the type of model that is used. As an example, it was shown that while dynamic attributes outperformed static attributes by a substantial margin in case of SVC RBF model, for a Random Forest model they were comparable.

4. Feature Extraction - While the numerical features and categorical features could be easily encoded and normalized to be fed into a machine learning model, there are other features like the log data that cannot be so easily used. For that different types of feature extractors can be used such as Bag of Words, TF-IDF, Word2Vec, BERT, AutoEncoders etc.
    - Even the feature extracters are trained using a dataset and hence due to the ever-changing nature of the dataset, the feature extracters too must be retrained and updated when the situation calls for it. This is done using Drift Detectors.
    - While selecting features, care must be given to selecting robust features that cannot be easily utilized by the attackers to make evasive malware.

Discussion

- The different metrics that can be utilized to evaluate a model was a point of discussion. It was mentioned that the False Positive Rate is considered a criteria, and hence after the model is optimized enough to perform under a required FPR, the other metrics could be used for comparison.
- The necessity for fixing class imbalance was also discussed. Without taking measures to accomodate for this factor, the model would be biased towards the majority. But at the same time, the measures taken for this should not destroy the representative nature of the dataset.
- A key point of discussion was that, in the case of oversampling, new samples were created not in the malware space but in the feature space. This would mean that new feature vectors were synthetically created to fix the class imbalance, but this does not mean that a corresponding malware file would or could be created.
- Temporal information is important in the context of malware detection and hence it should be preserved
- The point was raised that more data would lead to better performance, however marginal the difference might be. However, as a counterpoint, it was mentioned that there would be models that would have to be run on the client side machine which wouldn't have the resources to utilize such huge data. It would be an engineering problem to use the minimal amount of resources available to obtain the maximum performance, and hence it becomes essential to compare the trade-offs and make engineering decisions.