Machine Learning (In) Security: A Stream of Problems (Part 2) - Summary - Sidharth Anil

Presentation

This is a continuation of the topics presented in the previous session. The last session was over the different challenges and pitfalls faced during Model Selection, Data collection, Attribute Extraction and Feature Extraction in. This summary would be its continuation going into the actual Model, its Evaluation and an Overview of the bigger picture.

Model

1. The feature extractor utilized in a machine learning model is also dependent on the dataset and hence needs to be updated occasionally. The condition for initiating an update to the feature extractor is when a concept drift or a concept evolution occurs. A concept drift is when the relation between the input data and the target value changes while concept evolution is when the concepts are refined by researchers leading to new concepts and labels. This effect can be detected using drift detectors such as DDM, EDDM, ADWIN etc.
2. Adversial attacks are of two types - White Box attacks and Black Box attacks. In a white box attack, an attacker would have complete access to the machine learning model and hence can analyze the model to customize and create adversial attacks to bypass the model. On the other hand, a black box attack is when the attacker has no information about the model and hence has to either try out random modifications or try and recreate the model locally (essentially converting it into a white box attack).
   One of the main defensive strategy employed against dealing with adversial attacks, is for the malware detection side to come up with as many adversial variants of a malware for the detection model to train on. One implementation of this is through Generative Adversial Networks (GANs), where one neural networks generates inputs (adversial variants to the malware samples provided), and the other neural network classifies it. Both these neural networks work together to maximize their performance.
3. Another key point of the presentation was tackling the class imbalance problem at the model level (tackling it at the data level was discussed before). Cost sensitive learning is a solution at the model level, that imposes penalties on the model for incorrect predictions on minority class (falling prey to the majority bias). Ensemble learning is another example where multiple classifiers work together to achieve this.
4. At the model level, a potential avenue that is being explored for cybersecurity is transfer learning. A huge amount of research is being conducted in many different fields where cybersecurity applies such as the image detection problem. Transferring the knowledge from these areas into cybersecuirty leads to minimizing resources you have to put into solving the malware detection

task. Microsot and Intel has implemented this with good performance metrics. However, there are pros and cons to this idea, one con being that since the baseline model used for this purpose is most likley open source, there is a higher risk of white box attacks.

Metrics

In the context of cybersecurity, accuracy might not a good enough metric to measure the performance of a model. Accuracy misses a lot of important information like false positives and false negatives, each of which have significant consequences for a malware detection system. In this case, a confusion matrix would provide a more detailed and hence better picture of how the model is performing. Newly proposed metrics like Conformal Evaluator and Tesseract can also be considered to evaluate models effectively.
Furthermore, comparing different models meant for different usecases using these metrics can be misleading. An AV company's offline model and a local machine's online model would have different priorities and hence should be evaluated using different metrics.

Overview and Conclusions

A claim always made by AV companies is about how their model is 0-day resistent. This would imply that their model is effective against new malware that has never before been seen. However, this is simply a myth and as you understand how a machine learning model relies on data for detection, it is clear why this claim is a myth.
Secondly, for a malware detection system explainability is more important than it would be for other machine learning models. If a security engineer or researcher is able to identify why a model failed to detect a specific type of malware, he can take measures to rectify this and patch the security solutions. Having an explainable model would greatly help in this process.

Discussion

- The implementation of the gradient based white box attacks would be modifying malware files and not the feature vectors directly. The malware files would be modified incrementally towards the direction that leads to maximum evasive behaviour.
- In the case of transfer knowledge, there are cons that must be considered along with the pros that comes with it. For example, images with random pixels attached to the bottom is not a case that is considered in image processing whereas for malwares that could be an effective way of evasion.
- In the case of black box attacks, it was reiterated that the primary method of attack would be to try and recreate the model locally. From an attacker's point of view, submitting all the potential evasive malware files that you generate, against the actual model is not wise as the model would be utilizing all of these attempts to learn and be better. Instead, what is normally done is that all of the trial and error attempts would be against the locally recreated detection model. Only the final malware file that you have confidence in would be used against a real world system.
- The previous point led to a discussion on the ethics and legal complications of attacking your own system and generating adversial samples. It was discussed that without researchers engaging in

red-team behavioural activities such as these, it would be hard to create security solutions against real world attacks.

- Comparing between models, it is normally observed that Random Forest performs really well compared to other models. This might be because of the fact that the randomness in the model has the capability to handle non-linearity which is essential for drawing patterns between files and their labels. Also, once the data distribution changes (which is regular for malwares) it is easier to retrain a random forest, since only the part of the model affected by this change need to be modified with the rest of the forest remaining same. A counterpoint was mentioned that a neural network might have better performance than a random forest in the case of highly imbalanced dataset.
- One-class classifier and their application was also discussed. They can be used for detecting a specific kind of malware, for example when a company is trying to create a patch against a new malware released and researched. They can be easily added to a pipeline of detection models too.