Classifying Computer Processes in the DARPA OpTC dataset

Final Paper for XCS229ii - 003

Andrew Veal
Final Report
XCS229ii
Cohort 003
aveal@acm.org

FirstName Surname
Department Name
Institution/University Name
City State Country
email@email.com

FirstName Surname
Department Name
Institution/University Name
City State Country
email@email.com

ABSTRACT – 5 points

The detection of malware and malicious activity in enterprise networks is an ongoing challenge in cybersecurity. Our objective is to build a system that classifies activity associated with a computer process as benign or malicious, using host-based logging data from the computer. By focusing on host activity patterns rather than specific malware signatures, we hope to discover behavioral traits that distinguish malicious processes from benign processes. We used supervised learning and 'ground truth' labelled data to build a classification model for the DARPA Operationally Transparent (OpTC) dataset. This is challenging for a number of reasons: the dataset has over 17 billion events; only 0.0016% of events are malicious; the attacks used related *modus operandi*, so classifiers trained on this dataset may not generalize well to other attack scenarios. Our core hypothesis is that we can distinguish between malicious and benign processes using the frequency count of (object, action) events associated with each process as a feature vector. We show that a simple and interpretable Random Forest classifier can achieve 94% precision with a recall of 51% on the test set, which suggests our hypothesis is reasonable. We pay special attention to choosing training, validation and test sets from distributions that reflect the data we expect to get in the future. We also show that the DARPA OpTC dataset has the requisite scale, richness and class imbalance to become a new benchmark dataset for cybersecurity researchers.

CCS CONCEPTS

• Intrusion detection • Machine Learning • Big data analytics for security • Techniques and methods for generating training and test sets

AISec'2021, November 13, 2021, Seoul, South Korea

KEYWORDS

Intrusion detection, Machine Learning, Host Based Capability, Cybersecurity dataset

ACM Reference format:

Andrew Veal, FirstName Surname and FirstName Surname. 2021. Classifying Computer Processes in the DARPA OpTC dataset: Final Paper for XCS229ii - 003. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security (AISec 2021).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1234567890

INTRODUCTION – 10 points

The past twelve months has seen a spate of high-profile cyber-attacks on government, commercial and critical national infrastructure [4,6,11,15,18,20,28,32,34-37]. As the New York Times reported on the SolarWinds hack [35]:

> "Those behind the widespread intrusion into government and corporate networks exploited seams in U.S. defenses and gave away nothing to American monitoring of their systems."

The detection, classification and attribution of malware and malicious activity in enterprise networks is a cyber weapons arms race. State and non-state hackers develop new attacks faster than IT Security teams can deploy signature-based methods to detect new malicious Tactics, Techniques and Procedures (TTP). The SolarWinds compromise in 2020 was attributed to SVR [2] cyber actors, known and tracked in open source as APT29, Cozy Bear and The Dukes [27]. The National Cyber Security Centre (NCSC), alongside the US Department for Homeland Security's Cybersecurity Infrastructure Security Agency (CISA), Federal Bureau of Investigation (FBI) and the National Security Agency (NSA), published advice [27] on detection and mitigation of SVR activity, advising enterprises to

> "Set up a security monitoring capability so you are collecting the data that will be needed to analyse network intrusions."

Intrusion detection systems leveraging data collected on host computers are well established – see the review [3] – but the data collected on government, corporate or academic networks is not available to researchers outside these organizations. For example, the NCSC's Host Based Capability (HBC) service aims to strengthen central (UK) government's cyber security [25]. HBC is an IT monitoring and analysis service for (UK) government departments [26] – but the data is not available outside government. As noted in [38]:

> "The lack of diverse and useful data sets for cyber security research continues to play a profound and limiting role within the relevant research communities and their resulting published research."

For example, the utility of many available open-source cyber security data sets is compromised by anonymization and a lack of labelled malicious activity – for example, the LANL 2017 data release [21,38]. Many researchers use sandbox environments to measure machine activity [33] and software system calls [39], but this is not representative of large-scale enterprise networks. Machine Learning (ML) and Artificial Intelligence (AI) researchers need decent truth datasets with two fundamental properties:

- They are a large enough, real, representative sample to provide the necessary background noise.
- They need to have representative, and current, malicious activity in it.

The DARPA OpTC dataset [1,8,9,10] is a relatively new open-source simulated dataset that appears to have the requisite scale, richness and extreme class imbalance to drive AI/ML research. It is challenging to work with for a number of reasons:

- **Scale** – the DARPA OpTC dataset has over 17 billion events: significant data engineering is required to create the feature vectors for the Machine Learning (ML) experiments.
- **Extreme class imbalance** – Only 0.0016% of the events in the complete dataset [1] are malicious, so it is marginal whether there are sufficient examples for supervised learning.
- **Limited variability of attacks** –the attacks used related modus operandi [10]; therefore, classifiers trained on this dataset may not generalize well to other attack scenarios.

Many papers in the literature do not adequately describe the algorithms used, show a naïve application of Machine Learning (ML) or have poor experimental methodologies. For example,

- [39] indiscriminately apply 8 different ML algorithms and use accuracy as their success metric in settings with class imbalance.
- [40] simply apply 10-fold cross validation and use Area Under the ROC (AUROC) as the success metric in a paper titled "*Tackling Class Imbalance in Cyber Security Datasets*" – this (as should be well known – see [16,17]) is completely inappropriate.
- [7] develop a new classification algorithm 'SK-Tree' for detecting malware in computer process event logs. It is applied to data from one computer in the DARPA OpTC dataset (the host computer with the highest number of malicious events). It is hard to interpret the results as they simply apply cross validation and report the AUROC (again, see [16,17]).

In this paper, we establish a simple and interpretable baseline model for the classification of computer process activity in the DARPA OpTC dataset. For each process, we aggregate counts for the observed set of (object, action) events. Feature vectors for Machine Learning (ML) are created by joining the frequency counts with 'ground truth' labels [10]. We apply a small number of Machine Learning (ML) algorithms, taking care to partition the data into training, validation and test sets using stratified sampling and 5-fold cross validation. We use simple techniques to mitigate class imbalance and pay special attention to choosing training, validation and test sets from distributions that reflect the data we expect to get in the future [29].

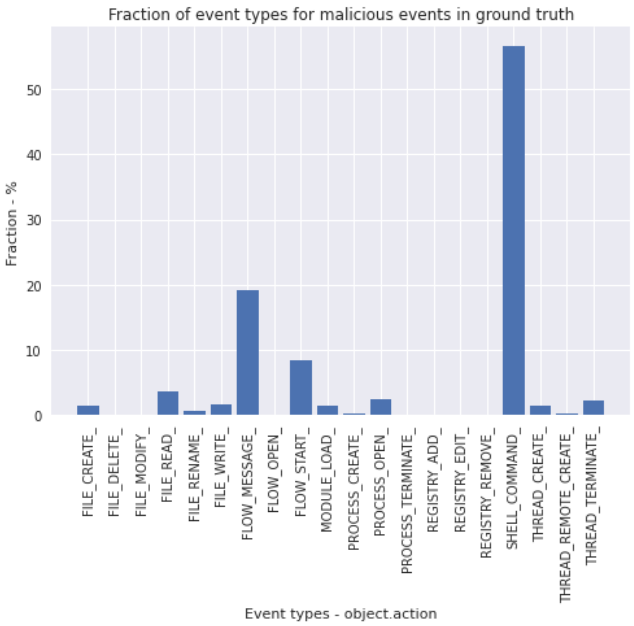

Figure 1: The fraction of event types for malicious events in the ground truth table.

METHODS – 12 points

The DARPA OpTC dataset contains events from an isolated enterprise network of 1000 host computers. Sensors logged 'action' events associated with 11 'object' types over 14 days. On 3 consecutive days, a 'red team' introduced malware and malicious activity on 29 computers [1,8,9,10].

The dataset is hosted on a Google drive [9] and documentation is available in a Github repository [8]. A ground truth document [10] provides details of the malicious events. Figure 1 shows the fraction of event types for malicious events in the ground truth table. Note, that we define a malicious process to be a process which is linked to a malicious event detailed in the ground truth [10]. The starting point for the project was the raw json data [9], ground truth table [10] and a table of 'process create' event data (extracted from properties in the raw json PROCESS_CREATE_ events).

Extract, Transform & Load (ETL) data engineering pipeline to create Machine Learning (ML) data

The raw json data was downloaded to AWS S3 and partitioned into 4554 part files, each o(500MB).

In the first ETL stage, we selected the data for the 29 computers attacked on the 3 'red team' days, dropping detailed meta-data contained in 'properties' tables [1]. The resulting 228 'summary' files, each o(400MB), contained the following fields:

- object – one of {FILE, FLOW, HOST, MODULE, PROCESS, REGISTRY, SERVICE, SHELL, TASK, THREAD, USER_SESSION}
- action – one of a limited set associated with a given object: see Figure 1 for the set of types
- actorID – the unique identifier (UID) for the parent process
- objectID – the unique identifier (UID) for the object (child process if object = PROCESS)
- hostname – the name of the computer (for example, SysClient0501.systemia.com)

In the next stage, we created aggregate counts for every object.action pair observed for each actorID. The resulting 212 'aggregate' files, each o(400KB), contained following additional field:

- object.action counts – dictionary of counts for the observed object.action pairs

In the final ETL stage, we join reference data from the ground truth table [10] and the 'process create' table to the 'aggregate' data to create a dataframe of feature vectors for Machine Learning (ML). The resulting ML dataframe has shape (349841, 37). The first 4 columns in the ML dataframe are

['actorID', 'hostname', 'process_name', 'label']

where process_name is the executable name, for example Powershell.exe, and label is 0 for benign and 1 for malicious. The remaining 33 columns contain the object.action counts for actorID. Figure 2 shows the fraction of event types for benign and malicious processes in the ML dataframe and Table 1 gives the number and proportion of benign and malicious processes in the ML dataframe.

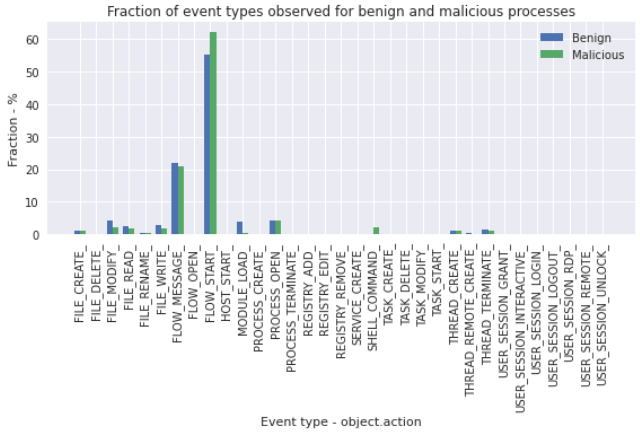

Figure 2: The fraction of event types for benign and malicious processes in the Machine Learning (ML) dataframe.

Class	Number of Processes	Fraction of Processes
Benign	348868	0.997219
Malicious	973	0.002781

Table 1: Number of benign and malicious processes in the Machine Learning (ML) dataframe

Our baseline model is the Random Forest (RF) classifier. It offers bagging and balanced class weights to mitigate the class imbalance and gives an out-of-bag estimate for accuracy. It has hyper-parameters we can tune and feature importance metrics to aid feature selection and interpretability. We used the scikit-learn [31] implementation RandomForestClassifier and set class_weight = balanced. We did some hyper-parameter tuning and used validation curves to set an optimized value for max_depth, which is the critical hyperparameter if you want to avoid over-fitting to the training set. We denote the Random Forest model with tuned hyper-parameters as RF*.

Exploratory Data Analysis (EDA) of the data suggests that decision trees could do well and they offer the additional benefit of explanatory results – for example, the following 'rule' detects only malicious processes in the ML dataframe df:

```
df[df['SHELL_COMMAND_']>2800][['label']].app
ly(lambda x: Counter(x))

{1: 12}
```

We also did experiments with Logistic Regression (LR) models using the scikit-learn [31] implementation LogisticRegression, setting class_weight = balanced (to mitigate class imbalance) and solver = sag (so the calculations scaled). The last model we tried was a Linear Support Vector Machine (SVM) using the scikit-learn [31] implementation SGDClassifier with class_weight = balanced (to mitigate class imbalance) and loss = hinge (to give Linear SVM).

Our core hypothesis is that we can distinguish between malicious and benign processes using the frequency count of the (object, action) events associated with each process as a feature vector. For each model, {LR, SVM, RF, RF*}, we calculated the performance metrics on training and test sets, created using stratified sampling. For the initial experiments, we used scikit-learn's [31] train_test_split function with train:test ratio 70:30, random sampling and stratify=y, where y contained the labels. We then used 5-fold stratified cross validation on the training set to calculate the Precision-Recall curve, the probability threshold value that maximized the F1 score and the area under the Precision-Recall curve. For information only, we also report AUROC. Finally, we recalculate the test set results using the probability threshold calculated using cross-validation.

If we look ahead to Table 2, we see that raising the probability thresholds drives down the false positive errors. We still need to examine misclassification errors in detail. By looking at off-diagonal cases in the confusion matrix, we investigated whether the hard to classify cases were associated with low event counts or specific executables (is PING.EXE predominantly benign or malicious?).

Coming up with an unbiased and effective data selection and partition strategy for train/validation/test sets is important. We looked at two different strategies:

- Simple stratified train/test split and stratified k-fold cross validation over the training set using scikit-learn. This is a naïve method that does not take account of the hostname or process_name or date when we split the dataframe. This is the method we used for model selection but there is a suspicion that we are causing data leakage.
- Custom stratified train/validation/test split on hostname, which ensures we don't place hosts with the same name in both the training, validation and test sets. This reflects the way malware propagates and the modus operandi of malicious attackers, who tend to step through hosts in some order. It also prevents data from the same host being placed in more than one partition.

RESULTS – 8 points

We shall use *precision*, *recall* and the *F1 score* as success metrics for the classification of malicious examples. In a setting where malicious examples are rare, it is important to account for the impact of false positives: the *precision* of the classifier is the fraction of processes in the validation (or test) set it labelled as malicious that really are malicious; *recall* is the fraction of all malicious processes in the validation (or test) set it correctly labelled. There is a trade-off between having a high *precision* and high *recall*. As we really care about both *precision* and *recall* we shall compute the *F1 score* and use it as our evaluation metric of success for optimizing thresholds [19,29].

As noted already, there is a trade-off between *precision* and *recall* – in Figure 3 we see the Precision-Recall (PR) curve as we vary the Random Forest probability threshold for classifying processes as malicious (calculated using 5-fold stratified cross-validation). The Area Under Curve (AUC) for the Precision-Recall (PR) curve is a useful measure of classifier performance when dealing with imbalanced classes [19] (see Table 2). In Figures 3 and 4, we show the point on the PR curve where the *F1 score* is a maximum and we report the values of the F1 score, precision, recall and the threshold at that point. Using the probability threshold value for the classifier, we can rerun our test set experiments to get the results at the bottom of Table 2.

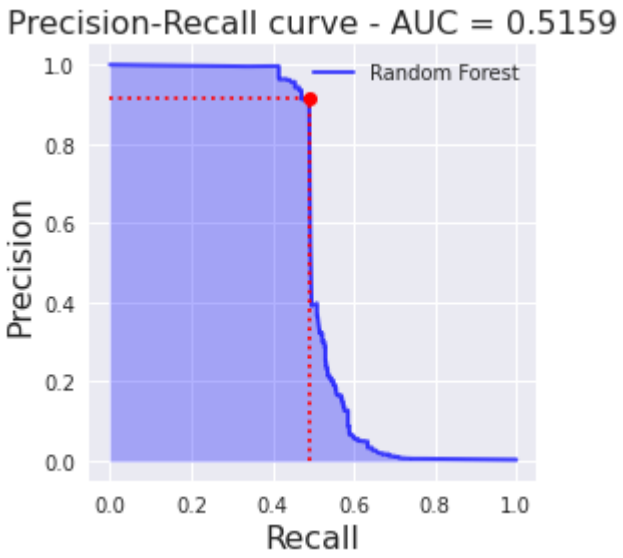

Figure 3: Precision-Recall curve for baseline Random Forest (RF) classifier, showing the point on the curve where the F1 score is maximized.

With extreme class imbalance it is important that we analyse bias-variance diagnostics at each stage of model selection and optimization. Following [29], we can interpret the bias as the error rate of the algorithm on the training set. In the model selection experiments we report in Table 2 we can see that the simple linear classifiers (Logistic Regression (LR) and linear Support Vector Machine (SVM))

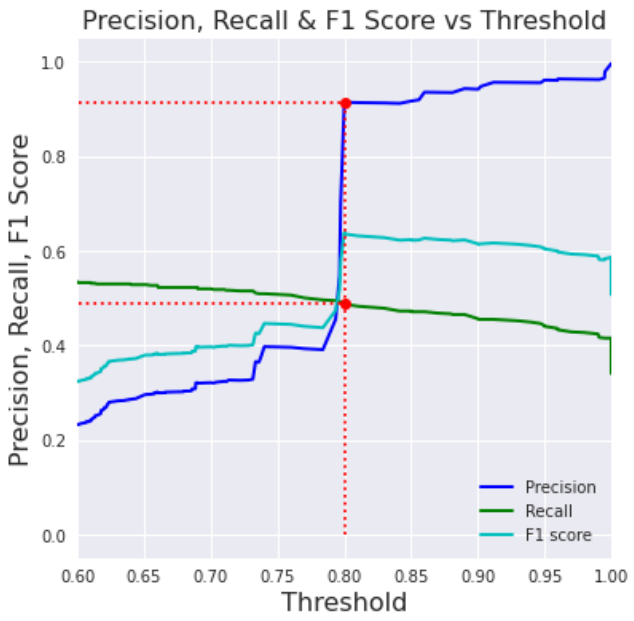

Figure 4: Precision, Recall and F1 Score as a function of the Threshold for the baseline Random Forest (RF) classifier, showing the probability value that maximizes the F1 Score.

Metric	LR	SVM	RF	RF*
Training set results				
Precision	0.0074	0.0072	0.2432	0.2835
Recall	0.4082	0.4332	0.9413	0.9060
F1 score	0.0145	0.0142	0.3865	0.4319
Test set results				
Precision	0.0081	0.0080	0.1551	0.1966
Recall	0.4452	0.4760	0.5582	0.6027
F1 score	0.0159	0.0157	0.2427	0.2965
Cross-Valid results				
Area Under PR curve	0.0238	0.0163	0.5159	0.5593
F1 score maximum	0.0818	0.0653	0.6386	0.6596
Precision at max (F1)	0.0672	0.0735	0.9151	0.9554
Recall at max (F1)	0.1043	0.0587	0.4905	0.5037
AUROC (for info only)	0.6834	0.6327	0.8106	0.8828
Test set – thresholded				
Threshold value	0.8780	142.16	0.8000	0.7844
Precision	0.0740	0.0747	0.8810	0.9430
Recall	0.1199	0.0616	0.5068	0.5103
F1 score	0.0915	0.0675	0.6435	0.6622

Table 2 Model selection - results for three types of model: Logistic Regression (LR), Linear Support Vector Machine (SVM) and baseline Random Forest (RF) classifier. We also give results for the Random Forest (RF*) with tuned hyper-parameters.

underfit the data and have a high bias. Again, following [29], we can interpret the variance as how much worse you do on the test set compared to the training set.

In fact, the results for the Random Forest are not bad:

- Training set accuracy for Random Forest was 0.9917 – see Table 2 for rest of metrics
- Out of bag estimate (validation set) accuracy for Random Forest was 0.9908
- Test set accuracy for Random Forest was 0.9903

The Random Forest already looks (in context) quite a good classifier: train, validation and test accuracy are all good and the out of bag estimate is close to test set accuracy. The key metrics for the baseline Random Forest (RF) classifier are the precision of 88% with a recall of 51% for the finished model with raised probability threshold on the test set (bottom of Table 2).

Another technique to reduce variance of the Random Forest is to do feature selection to decrease the number of input features: looking at Figure 5, where we rank feature importance, it appears that the top 20 features will capture the key discriminating factors (and the ground truth only used 20 features).

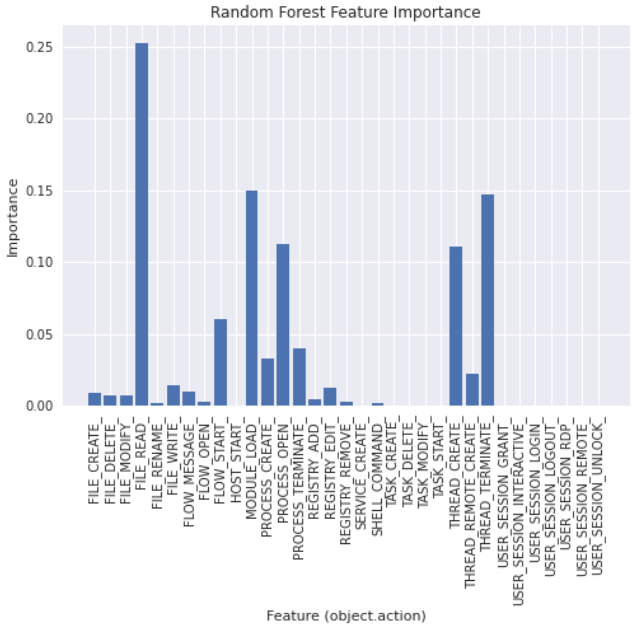

Figure 5: Feature importance for each of the (object, action) events for the baseline Random Forest (RF) classifier.

As noted previously, AUROC should never be used to measure performance [16,17] but as it is prevalent in the literature [7,39,40] we have given AUROC values in Table 2 for information only.

Having selected the Random Forest (RF) as our chosen Model, we did some tuning of the hyper-parameters max_depth and n_estimators. Figure 6 shows the validation curve for optimizing the max_depth hyper-parameter: we chose to set max_depth = 15. The n_estimators hyper-parameter did not show much variation so we kept the default value (100). The key metrics for the tuned Random Forest (RF*) are the precision of 0.94% with a recall of 51% for the finished model with raised probability threshold on the test set (bottom of Table 2).

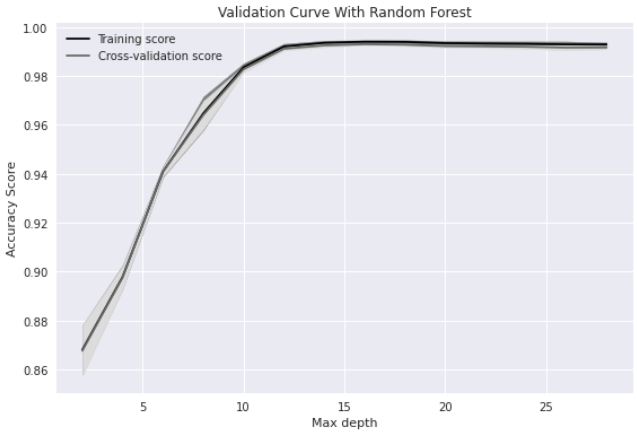

Figure 6: Validation curve for tuning the max_depth hyper-parameter for the Random Forest classifier.

The results in Table 2 are an example of a 'Model-centric' view [30] – we hold the data fixed and vary the models (LR, SVM, RF) and then iteratively improve the selected model (RF*) using the same benchmark dataset. When doing error analysis, we took a 'Data-centric' view [30] – we examined the false positive and false negative cases to see if we could identify executable files (process_name) associated with hard to classify computer processes and we then removed some of the rows of the dataframe to understand which examples the models performed well on. This is an example of holding the model fixed (RF*) and iteratively changing the data to understand performance.

We trained the tuned Random Forest (RF*) and got predictions for the validation dataset. Table 3 shows frequency counts for the top 5 executables associated with computer processes in the four quadrants of the confusion matrix for classification for data drawn from the train-validation partition of the dataframe. This shows that:

- The most common benign computer processes (true negative) are cmd.exe and conhost.exe
- The most commonly mis-classified executable is PING.EXE and it is the highest ranked false positive (misidentified as malicious)
- Unknown computer processes are likely long running processes that were created before logging was active
- The false negative cases are malicious – it would be interesting to ask a domain expert if these are well-known exploited executables
- The true positive counts show us that our relatively good classification results are due to identifying PING.EXE correctly, where it is being used maliciously

Top 5	Count
True Negative	
cmd.exe	114570
conhost.exe	101817
unknown	34039
schtasks	11574
PING.EXE	9684
False Positive	
PING.EXE	95
unknown	55
conhost.exe	48
svchost.exe	42
cmd.exe	38
False Negative	
unknown	54
fontdrvhost.exe	18
ShellExperienceHost.exe	16
svchost.exe	3
sihost.exe	2
True Positive	
PING.EXE	412
powershell.exe	20
wmiprvse.exe	16

Table 3: Frequency counts for the Top 5 executables for computer processes in the four quadrants of the confusion matrix arising from applying the tuned Random Forest (RF*) to the validation data under 5-fold cross validation.

As an experiment we removed PING.EXE from the Machine Learning dataframe, trained (fit) and predicted labels using the tuned Random Forest (RF*) classifier and reviewed the performance of the classifier. As would be expected, the performance of the classifier was degraded. The Area Under Precision Recall was 0.1602 compared to 0.5118 when PING.EXE is included.

DISCUSSION – 12 points

The discovery of malicious computer processes in an enterprise network is a key component of intrusion detection systems (IDS). The importance of IDS was underlined as I was writing the final draft of this report, as there was another news report of a new cyber-attack on the U.S. State Department [36].

Our hypothesis in this paper has been that we can distinguish between malicious and benign computer processes using the frequency counts of the (object, action) events associated with each process as a feature vector. Our results give some support to the hypothesis – we were able to train a Random Forest classifier to achieve 94% precision with a recall of 51%. But we should also recognize that our relatively good results are due to the ability to distinguish malicious use of the PING.EXE executable.

We are taking a summary view of process activity, aggregating frequency counts of high-level events only. We are not using the full richness of the dataset: each (object, action) event has detailed properties [1]. For example, PROCESS.CREATE event table contains meta-data for timestamp, user, image path and command line.

We showed results for three types of models – the Random Forest performed better than the linear models and also has the benefit that the Random Forest feature importance can identify the event-types which are more important for the classification of malicious processes.

We did not have time to look at Gradient Boosted Trees or Neural Networks. We did manage to get partial results for K-Nearest Neighbor (KNN) using the PyNNDescent package [12, 14] – the Precision Recall curve is shown in Figure 7 below.

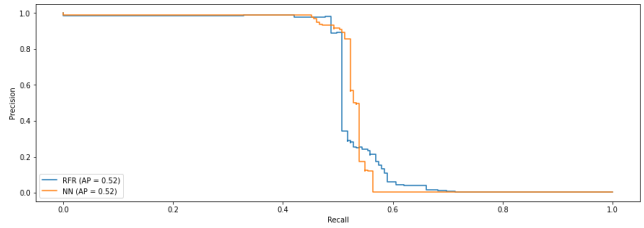

Figure 7: Precision-Recall curves for Random Forest and K-Nearest Neighbor (KNN), using PyNNDescent [12, 14].

Our work has shown that the DARPA OpTC dataset has the requisite scale, richness and class imbalance to drive AI research – it is large enough to be representative of the benign background 'noise' in an enterprise network [1]. What is less clear is whether it has sufficient representative malicious activity in it. Although there are 17 billion events in the dataset, this is really a small data problem because we are interested in rare events in the long tails of the distribution of the data [30].

We had hoped to better understand the work reported in [7]. The authors apply a new classification algorithm 'SK-Tree' to data from one computer in the DARPA OpTC dataset. They use a powerful streaming tree data structure and the methods appear to show great promise – but it is difficult to interpret the results in their paper. As we have shown, there is very little malicious data and it is not that diverse so supervised machine learning is probably not the best technique to use – unsupervised anomaly detection techniques would probably be more appropriate.

We didn't have time to investigate techniques for imbalanced classes [13,23] beyond using balanced weights to mitigate the class imbalance. For example, the imbalanced-learn package [22] has methods to under-sample the majority (benign) class.

We expect to use the DARPA OpTC dataset to prototype data engineering (MLOps) work flows and to scale system testing with representative data that is not sensitive.

ACKNOWLEDGMENTS

My employer enabled me to access IEEE publications during the Literature Review phase of the project. My employer also allowed me to use our corporate AWS account to do the Extract Transform and Load (ETL) pipeline to reduce the full DARPA dataset down to a scale at which the Machine Learning (ML) project could begin. We did experiments on the ML dataset on Amazon SageMaker – but the ML dataset can be processed on a laptop with Anaconda installed.

REFERENCES – 3 points

[1] Md. Monowar Anjum, Shahrear Iqbal and Benoit Hamelin. 2021. Analysing the Usefulness of the DARPA OpTC Dataset in Cyber Threat Detection Research. arXiv:2103.03080v2. Retrieved from https://arxiv.org/abs/2103.03080
Accepted for *ACM Symposium on Access Control Models and Technologies (SACMAT)*, 16-18 June, 2021, Barcelona, Spain [virtual event]. ACM Inc., New York, NY. DOI: https://doi.org/10.1145/3450569.3463573

[2] BBC. 2010. Profile: Russia's SVR intelligence agency. (June 29, 2010) Retrieved May 12 2021 from https://www.bbc.co.uk/news/10447308

[3] Robert A. Bridges, Tarrah R. Glass-Vanderlan, Michael D. Iannacone, Maria S. Vincent, and Qian (Guenevere) Chen. 2019. A Survey of Intrusion Detection Systems Leveraging Host Data. *ACM Comput. Surv.* 52, 6, Article 128 (November 2019), 35 pages. DOI: https://doi.org/10.1145/3344382

[4] Ben Buchanan. 2020. *The Hacker and the State: Cyber Attacks andthe New Normal of Geopolitics.* Harvard University Press, Cambridge, MA

[5] Anthony Burke. 2020. Robust artificial intelligence for active cyber defence. The Alan Turing Institute. Defence and Security Programme. (March 2020) Retrieved May 12, 2021 from https://www.turing.ac.uk/research/publications/robust-artificial-intelligence-active-cyber-defence

[6] Richard A. Clarke and Robert K. Knake. 2019. *The Fifth Domain: Defending Our Country, Our Companies, and Ourselves in the Age of Cyber Threats.* Penguin Press, New York, NY.

[7] Thomas Cochrane, Peter Foster, Varun Chhabra, Maud Lemercier, Terry Lyons and Cristopher Salvi. 2021. SK-Tree: a systematic malware detection algorithm on streaming trees via the signature kernel. arXiv:2102.07904v3. Retrieved from https://arxiv.org/abs/2102.07904

[8] DARPA. 2020. Operationally Transparent Cyber (OpTC) Data Release. README. Retrieved from http://github.com/FiveDirections/OpTC-data

[9] DARPA. 2020. Operationally Transparent Cyber (OpTC) Data Release. Retrieved from https://drive.google.com/drive/u/0/folders/1n3kkS3KR31KUegn42yk3-e6JkZvf0Caa

[10] DARPA. 2020. OpTC Red Team Ground Truth. Retrieved April 7, 2021 from https://github.com/FiveDirections/OpTC-data/blob/master/OpTCRedTeamGroundTruth.pdf

[11] Neil Daswani and Moudy Elbayadi. 2021. Big Breaches: Cybersecurity Lessons for Everyone. (1st Edition). Apress, Berkeley, CA. DOI: https://doi.org/10.1007/978-1-4842-6655-7

[12] Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web (WWW 2011).* March 28-April 1, 2011, Hyderabad, India. ACM. Retrieved from https://www.cs.princeton.edu/cass/papers/www11.pdf

[13] Alberto Fernandez, Salvador Garcia, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk and Francisco Herrera. 2028. *Learning from Imbalanced Data Sets.* Springer. Switzerland. DOI: https://doi.org/10.1007/978-3-319-98074-4

[14] GitHub. PyNNDescent. Retrieved from https://github.com/lmcinnes/pynndescent

[15] Sue Halpern. 2021. After the SolarWinds Hack, we have no idea what Cyber dangers we face. *The New Yorker – Daily Comment* (January 25, 2021) Retrieved January 26, 2021 from https://www.newyorker.com/news/daily-comment/after-the-solarwinds-hack-we-have-no-idea-what-cyber-dangers-we-face

[16] David J. Hand. 2009. Mismatched Models, Wrong Results, and Dreadful Decisions. Keynote at *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, June, 2009 Paris, France recorded June 2009, published September 14, 2009 http://videolectures.net/kdd09_hand_mmwrdd/ (video) http://videolectures.net/site/normal_dl/tag=45840/kdd09_hand_mmwrdd_01.pdf (slides)

[17] David J. Hand. 2009. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Mach Learn*, 77 (2009), 103–123. DOI: https://doi.org/10.1007/s10994-009-5119-5

[18] Robert Hannigan. 2020. SolarWinds hack exploited weaknesses we continue to tolerate. *The Financial Times (December 20, 2020).* Retrieved January 12, 2021 from https://www.ft.com/content/2bed3013-b21f-4b2c-8572-b2da016d1b4e

[19] Haibo He and Edwardo A. Garcia. 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, Vol.21, No. 9, (September 2009), 1263-1284. DOI: https://doi.org/10.1109/TKDE.2008.239

[20] Max Heinemeyer. 2021. Dissecting the SolarWinds hack without the use of signatures. Retrieved from https://www.darktrace.com/en/blog/dissecting-the-solar-winds-hack-without-the-use-of-signatures/

[21] LANL. 2017. Unified Host and Network Data Set. Retrieved from https://csr.lanl.gov/data/2017/

[22] Guillaume Lemaitre, Fernando Nogueira and Christos K. Aridas. 2016. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research,* 7 (2016) 1-5 DOI: https://dl.acm.org/doi/pdf/10.5555/3122009.3122026

[23] Bartosz Krawczyk. 2016. Learning from imbalanced data: open challenges and future directions. *Prog. Artif. Intell.* 5 (2016) 221-232. DOI: https://doi.org/10.1007/s13748-016-0094-0

[24] MIT Technology Review Insights. 2021. Preparing for AI-enabled cyberattacks. Retrieved from https://www.technologyreview.com/2021/04/08/1021696/preparing-for-ai-enabled-cyberattacks/

[25] National Cyber Security Centre. 2020. Host Based Capability. (May 4, 2020) Retrieved May 11, 2021 from https://www.ncsc.gov.uk/pdfs/information/host-based-capability.pdf

[26] National Cyber Security Centre. 2020. Introducing Host Based Capability (HBC). (November 6, 2020) Retrieved May 11, 2021 from https://www.ncsc.gov.uk/pdfs/blog-post/introducing-host-based-capability-hbc.pdf

[27] National Cyber Security Centre. 2021. Joint advisory: Further TTPs associated with SVR cyber actors. Published May 7, 2021. Retrieved from https://www.ncsc.gov.uk/news/joint-advisory-further-ttps-associated-with-svr-cyber-actors

[28] National Cyber Security Centre. CyberUK Online 2021. Oh that was clever! When even jaded incident responders are impressed. YouTube https://youtu.be/ppXOt8f5H8Q

[29] Andrew Ng. 2018. Machine Learning Yearning: Technical Strategy for AI Engineers, In the Era of Deep Learning. Draft Version. Retrieved May 6, 2021 from https://www.deeplearning.ai/programs/

[30] Andrew Ng. 2021. A chat with Andrew on MLOps: From Model-centric to Data-centric AI. You Tube https://youtu.be/06-AZXmwHjo

[31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research.* 12(85) (2011) 2825−2830. DOI: https://dl.acm.org/doi/10.5555/1953048.2078195

[32] Nicole Perlroth. 2021. *This is how they tell me the World ends: The Cyber Weapons Arms Race.* Bloomsbury Publishing, London.

[33] Matilda Rhode, Pete Burnap and Kevin Jones. 2018. Early-stage malware prediction using recurrent neural networks. *Computers & Security,* 77 (August 2018), 578-594. DOI: https://doi.org/10.1016/j.cose.2018.05.010

[34] David E. Sanger. 2018. *The Perfect Weapon: War, Sabotage, and Fear in the Cyber Age.* Scribe, London

[35] David E. Sanger, Nicole Perlroth and Julian E. Barnes. 2021. As Understanding of Russian Hacking Grows, So Does Alarm. *The New York* (January 2, 2021) Retrieved January 5, 2021 from https://www.nytimes.com/2021/01/02/us/politics/russian-hacking-government.html

[36] David E. Sanger and Nicole Perlroth. 2021. Russia Appears to Carry Out Hack Through System Used by U.S. Aid Agency. *The New York Times* (May 28, 2021) Retrieved May 28, 2021 from https://www.nytimes.com/2021/05/28/us/politics/russia-hack-usaid.html

[37] Bruce Schneier. 2018. *Click Here to Kill Everybody: Security and Survival in a Hyper-Connected World.* W.W.Norton & Company. New York, NY.

[38] Melissa J. M. Turcotte, Alexander D. Kent and Curtis Hash. 2018. Unified Host and Network Data Set. In *Data Science for Cyber-Security*, Chapter 1 (November 2018), 1-22. World Scientific DOI: https://doi.org/10.1142/9781786345646_001

[39] Aaron Walker and Shamik Sengupta. 2019. Insights into Malware Detection via Behavioral Frequency Analysis using Machine Learning. In *Proceedings of the 2019 IEEE Military Communications Conference (MILCOM)*, 12-14 November, 2019, Norfolk, VA, USA. IEEE *Explore,* 1-6. https://doi.org/10.1109/MILCOM47813.2019.9021034

[40] Charles Wheelus, Elias Bou-Harb and Xingquan Zhu. 2018. Tackling Class Imbalance in Cyber Security Datasets. In *Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration (IRI)*, 6-9 July, 2018, Salt Lake City, UT, USA. IEEE *Xplore*, 229-232. https://doi.org/10.1109/IRI.2018.00041