On Botnet Detection with Genetic Programming under Streaming Data, Label Budgets and Class Imbalance

Sara Khanchi
Dalhousie University
Halifax, NS, Canada

Ali Vahdat
Huawei Technologies, Noah's Ark Lab.
Montreal, QC, Canada

Malcolm I. Heywood
Dalhousie University
Halifax, NS, Canada

A. Nur Zincir-Heywood
Dalhousie University
Halifax, NS, Canada

ABSTRACT

Botnets represent a widely deployed framework for remotely infecting and controlling hundreds of networked computing devices for malicious ends. Traditionally, detection of Botnets from network data using machine learning approaches is framed as an offline, supervised learning activity. However, in practice both normal behaviours and Botnet behaviours represent non-stationary processes in which there are continuous developments to both as new services/applications and malicious behaviours appear. This work formulates the task of Botnet detection as a streaming data task in which finite label budgets, class imbalance and incremental/online learning predominate. We demonstrate that effective Botnet detection is possible for label budgets as low as 0.5% when an active learning approach is adopted for genetic programming (GP) streaming data analysis. The full article appears as S. Khanchi et al., (2018) "On Botnet Detection with Genetic Programming under Streaming Data, Label Budgets and Class Imbalance" in Swarm and Evolutionary Computation, 39:139-140. https://doi.org/10.1016/j.swevo.2017.09.008

CCS CONCEPTS

• **Information systems → Data streaming**; • **Computing methodologies → Genetic programming**;

KEYWORDS

Botnet detection, Active learning

ACM Reference Format:
Sara Khanchi, Ali Vahdat, Malcolm I. Heywood, and A. Nur Zincir-Heywood. 2018. On Botnet Detection with Genetic Programming under Streaming Data, Label Budgets and Class Imbalance. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3205651.3208206

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3208206

1 INTRODUCTION

Botnets represent a collection of networked devices that at some point have had their security compromised (bots), so letting a bot master remotely control them. Unknown to the original users, the bot master is then free to use the compromised devices to perform malicious behaviours (e.g. spam, click fraud, distributed denial of service, identify theft). Detection of Botnets is non-trivial because: 1) malicious behaviours are mixed in with legitimate normal behaviours; 2) users have a wide range of 'normal' behaviours; 3) network load and application mix are time varying parameters; 4) many applications dynamically switch between different modes of operation in unpredictable ways (e.g., services such as Skype and Tor explicitly attempt to hide their communication protocols); 5) new applications/updates to current applications (whether malicious or not) coexist with both old versions of the same application and, 6) the ratio of data pertaining to malicious versus non-malicious behaviour is very low.

Typically, Botnet detection is framed as an off-line activity in which either prior rules are used to detect Botnet activity, or detectors are trained on a prior dataset using supervised learning (e.g. see [3]). In this work an incremental approach is adopted in which data is viewed as a continuous stream. Specifically, the task is framed as follows. We cannot predict a priori when Botnet behaviours will appear in the stream, as network data represents a mixture of normal and malicious data. Normal network data is also non-stationary, implying that it is also not feasible to pre-train models off-line and then deploy. Human expert(s) are available for providing true labels for a *small* subset of the stream data (i.e. label budget) on a continuous basis. This is necessary because an attacker can manipulate stream data content leading to attacks against the machine learning algorithm itself [1]. A champion GP individual must always be available for label prediction, before any label querying can take place (real-time anytime operation). The GP framework therefore operates interactively with the stream providing predictions about the content (normal or Botnet) and directs the human labelling of the stream under a finite label budget. In framing the task this way, the proposed system has the ability to operate under a wide range of network devices including servers and client devices.

2 STREAMING GP

The development of machine learning algorithms for streaming data has to address multiple properties [4], including but not limited to: change detection (non-stationary processes generating the data), anytime operation, incremental or online updates, and small

amounts of label information. Multiple mechanisms have been proposed/developed for addressing these issues, although comparatively little from the perspective of evolutionary computation (reviewed in [4]).

The form of GP deployed in this work cooperatively coevolves teams of programs, where this is synonymous with the development of an ensemble of classifiers, and has previously been shown to be much more effective under streaming data scenarios than solutions taking the form of single monolithic programs [7]. Thus, a teaming approach to GP enables better selectivity for incrementally removing specific components of a model, hence reacting to change that might be specific to particular class(es) or sub-class(es).

The specific emphasis of this work is with regards to a set of design decisions for enabling streaming operation under: limited label budgets, incremental improvement to the classifier, and resilience to high degrees of class imbalance. To do so, an active learning framework is adopted in which GP fitness evaluation is only performed relative to the content of a small sample of labeled exemplars. Two policies then need defining, a sampling policy and an archiving policy. The sampling policy determines under what conditions labels are requested for exemplars within the current window (interface to the stream) and enforces the label budget. The archiving policy determines what exemplars from the data subset to replace (with the most recently labeled data). Two sampling policies are considered: uniform random sampling and biased sampling in which the current champion GP classifier is used to prioritize instances from the current window location for labelling. Two archiving policies are considered: uniform random identification of exemplars currently in the data subset, or biased replacement. The biased replacement model prioritizes exemplars in proportion to: 1) how overrepresented their class is in the data subset, and 2) when a class is selected, replace the older instances.

This results in a total of 4 configurations for Streaming GP: Rnd (both policies assume uniform random selection), Sample (Biased sampling, Uniform archiving), Archive (Uniform sampling, Biased archiving) and Both (Biased sampling and archiving).

3 EVALUATION

The CTU dataset [2] represents a state-of-the-art collection of 13 different datasets describing multiple Botnets. Labels define one of four general categories: background, normal, Botnet, command and control (C&C). The majority of the data present in the data set takes the form of 'background' traffic, where this represents network traffic collected from a real-world network. Filters were then used to characterize definitively known examples of normal behaviour [5]. Any data from the background traffic labelled by the 'normal' filters are labelled as normal, the remainder is labeled as background. Finally, attack data is explicitly created using (Botnet) attack tools from specific IP addresses on a virtual network. This means that any data explicitly labeled as attack is definitely attack data, although some amount of the background traffic data could also be so. Moreover, data associated with the operation of the Botnet master is explicitly distinguished from that of data associated with Botnet slaves (labelled as C&C and Botnet respectively).

The data is described by 12 'flow' statistics obtained by the Argus network flow generator. However, out of these 12 features, IP addresses and port numbers are *not employed* as many recent network applications (Voice over IP, social media and network based games) can dynamically change their port addresses based on the blocked/unblocked port combinations. Moreover, IP addresses can be spoofed by attackers for malicious intentions or can be hidden by proxies for legitimate reasons to protect privacy of users. Thus, any classifier relying on these attributes may not generalize well in real world applications.

The resulting dataset is particularly challenging for streaming machine learning algorithms because of the non-stationary properties present in both normal and Botnet behaviour. Moreover, the datasets are extremely unbalanced, with the C&C class appearing at a rate of less than 0.5% of stream content.

Comparator algorithms are adopted that also operate with label budgets care of the MOA toolset for machine learning under streaming data [6]. Three label budget limits are assumed: 5%, 1% and 0.5%. The 5% budget would be prohibitively high in practice, but provides an indication of how much might be gained if more labels could be provided.

4 RESULTS

A clear preference for 'Archive' and 'Both' configurations of Streaming GP is demonstrated across all 13 CTU datasets with Naive Bayes (with the variable active learning policy from [6]) typically ranked third. This was true across all label budgets. In addition, analysis of the the dynamic properties of the Streaming GP framework indicated that it was particularly effective at detecting the two Botnet classes both early and reacting to new instances. We also demonstrate how the Archive and Both configurations result in effective balancing of the Data Subset during the course of the stream. Finally, the anytime nature of Stream GP is documented, demonstrating that the evolutionary step is completed in $\approx$ 2.7 seconds and predictions made at the rate of 4.1μ seconds.

ACKNOWLEDGMENTS

This research is supported by the Canadian Safety and Security Program(CSSP) E-Security grant. The CSSP is led by the Defense Research and Development Canada, Centre for Security Science (CSS) on behalf of the Government of Canada and its partners across all levels of government.

REFERENCES

[1] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar. 2010. The security of machine learning. *Machine Learning* 81, 2 (2010), 121–148.
[2] S. García, M. Grill, J. Stiborek, and A. Zunino. 2014. An empirical comparison of botnet detection methods. *Computers & Security* 45 (2014), 100–123.
[3] Fariba Haddadi and A. Nur Zincir-Heywood. 2017. Botnet behaviour analysis: How would a data analytics-based system with minimum a priori information perform? *International Journal of Network Management* 27, 4 (2017).
[4] Malcolm I. Heywood. 2015. Evolutionary model building under streaming data for classification tasks: opportunities and challenges. *Genetic Programming and Evolvable Machines* 16, 3 (2015), 283–326.
[5] C. Rossow, C. J. Dietrich, C. Grier, C. Kreibich, V. Paxson, N. Pohlmann, H. Bos, and M. van Steen. 2012. Prudent practices for designing malware experiments: Status quo and outlook. In *IEEE Symposium on Security and Privacy*. 65–79.
[6] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes. 2014. Active Learning With Drifting Streaming Data. *IEEE Transactions on Neural Networks and Learning Systems* 25, 1 (2014), 27–54.
[7] Ali Vahdat, Jillian Morgan, Andrew R. McIntyre, Malcolm I. Heywood, and A. Nur Zincir-Heywood. 2015. Evolving GP Classifiers for Streaming Data Tasks with Concept Change and Label Budgets: A Benchmarking Study. Springer, 451–480.