Swarm and Evolutionary Computation xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

On botnet detection with genetic programming under streaming data label budgets and class imbalance

Sara Khanchi, Ali Vahdat, Malcolm I. Heywood*, A. Nur Zincir-Heywood

Faculty of Computer Science, Dalhousie University, 6050 University Av., Halifax, NS, Canada

ARTICLE INFO

Keywords:
Non-stationary data
Streaming data
Botnet detection
Class imbalance
Genetic programming

ABSTRACT

Algorithms for constructing models of classification under streaming data scenarios are becoming increasingly important. In order for such algorithms to be applicable under 'real-world' contexts we adopt the following objectives: 1) operate under label budgets, 2) make label requests without recourse to true label information, and 3) robustness to class imbalance. Specifically, we assume that model building is only performed using the content of a Data Subset (as in active learning). Thus, the principle design decisions are with regard to the definitions employed for sampling and archiving policies. Moreover, these policies should operate without prior information regarding the distribution of classes, as this varies over the course of the stream. A team formulation for genetic programming (GP) is assumed as the generic model for classification in order to support incremental changes to classifier content. Benchmarking is conducted with thirteen real-world Botnet datasets with label budgets of the order of 0.5–5% and significant amounts of class imbalance. Specific recommendations are made for detecting the costly minor classes under these conditions. Comparison with current approaches to streaming data under label budgets supports the significance of these findings.

1. Introduction

Streaming data applications represent an environment in which data arrives on a continuous basis and exhibits non-stationary properties such as concept drift [1–4]. Thus, records $(\vec{x})$ appear sequentially at discrete points in time, t, and are described by a joint probability distribution $p_t(\vec{x}, d)$, where in this work d represents the record's *unknown* true label. If for two points in time, t and $t+1$ there exists an $\vec{x}$ such that $p_t(\vec{x}, d) \neq p_{t+1}(\vec{x}, d)$, then concept drift has occurred. Such drift might be slow or abrupt, subject to repetition and/or effect different subsets of classes at different points in time.

The goal of a classification model operating on such streams is therefore multifaceted. Not only is it necessary to suggest labels for multiple classes of data in the stream on a real-time/anytime basis, but it is also necessary for the model to identify *what data to learn from.*[1] The process of identifying what to learn from constitutes a 'label request' as a human expert is ultimately responsible for providing ground truth labels. Moreover, it is only feasible for the model to request labels for a small fraction of the data (the cost of acquiring labels is high). Such constraints potentially appear in several applications, e.g. constructing trading agents for financial services or labelling satellite data.

In this work we are motivated by the particular issue of identifying Botnet behaviours in network traffic data. Botnets represent a networked collection of devices whose security was at some point compromised (the bots), so allowing a bot herder/master to remotely control the bots. The owners of the compromised devices are unaware of the ability of the bot master to control their devices. The bot master is then free to use the bots to launch a wide range of malicious behaviours while hiding their own identity. 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 resulting in multiple simultaneous 'fingerprints' for the same application; and, 6) the ratio of data pertaining to malicious versus non-malicious behaviour is very low.

The Botnet detection scenario 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.

* Corresponding author.
E-mail address: mheywood@cs.dal.ca (M.I. Heywood).

[1] The non-stationary properties of the stream imply that training data has to be identified interactively during the course of deployment.

http://dx.doi.org/10.1016/j.swevo.2017.09.008
Received 16 November 2016; Received in revised form 16 August 2017; Accepted 9 September 2017

Normal network data is also non-stationary, implying that it is also not feasible to pre-train models off-line and then deploy (such models will always 'go stale' as both normal and malicious data change at unpredictable points in time). Human expert(s) are available for providing true labels, *d*, for a *small* subset of the stream data (i.e. label budget) on a continuous basis. This is necessary because attacks against the machine learning algorithm itself lead to an attacker 'reprogramming' the classification of attack behaviours as normal by manipulating stream data content [5,6]. In order to decouple human experts from the raw throughput of the network data, only the GP framework will identify data for labelling, not the human, i.e. this step cannot assume access to the true labels. We assume that the human experts are trustworthy (otherwise GP models could again be misled). 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.[2] In framing the task this way, the proposed system has the ability to operate under incoming and/or outgoing network traffic on a wide range of network devices including servers and client devices. Such a framework would be deployed to protect institutions/infrastructure such as medical, financial or other institutions with human security experts acting as the ultimate source of trusted label information. Other scenarios might include IT security companies who provide the anytime classifier to service subscribers and retain the other components of the architecture.

In the following, we develop the topic by reviewing previous works that address both the ability to operate under label budgets and address the issue of class imbalance under streaming data (Section 2). The framework we propose assumes a teaming formulation for genetic programming (GP), where team GP formulations provide an evolutionary approach for adapting an 'ensemble' of GP programs to data content. Section 3 establishes how GP teams are evolved from a fixed size Data Subset, as per active learning, thus the following two critical decisions are addressed: 1) how to sample records from the stream to appear within the Data Subset *without* requiring label information; and 2) how to identify records for replacement from the Data Subset when the subset is full. Section 4 develops the methodology adopted for streaming classification algorithms operating under label budgets with class imbalance, and introduces the real-world Botnet datasets employed for benchmarking.

The ensuing empirical study both quantifies the significance of the GP teaming approach and compares to recent work capable of operating under label budgets (Section 5). We make specific recommendations regarding sampling versus replacement policies for GP and quantify the impact of operating under low label budgets while addressing class imbalance. Indeed, for streaming data applications to be appropriate for real-world applications, it is necessary for them to operate under both of these constraints simultaneously. Section 6 concludes the paper and suggests future research.

2. Related work

Several recent survey articles have appeared that provide overviews of the scope of model building for streaming data classification under non-stationary streams [2–4]. In the following we will concentrate on highlighting issues specific to the problem setting of streaming classification under label budgets: Section 2.1 reviews developments regarding imbalanced data, change detection, and (online) active learning from the perspective of non-evolutionary methods; and Section 2.2 provides an equivalent survey from the perspective of explicitly evolutionary methods.

2.1. Non-evolutionary methods

Change detection is a mechanism used to initiate retraining of a model. Thus, only when sufficient change is detected, will a model be updated. This potentially means that model building is decoupled from the need to provide labels. For example, Lindstrom et al. describe a process by which a reference distribution is constructed and used to calibrate the model [7]. As the model passes over the stream a divergence measure (expressing model confidence independent from label information) is used to trigger model reconstruction. Any model reconstruction is only performed from the most recent window content. Such an approach only requests labels once a change is detected. However, it also assumes that variation is solely captured by the unconditional distribution of data $p(\vec{x})$. Any change to the posterior distributions of data $p(y|\vec{x})$ remains undetected [8].

Active Learning implies that labels are explicitly sought for some fraction of the data, and employ some form of change detection/uncertainty threshold to initiate label requests. Several authors have proposed bias/variance minimization schemes for this purpose [9,10,1,11,12]. That said, empirical benchmarking has demonstrated that just sampling with uniform probability (up to the label budget) is sufficient to build surprisingly effective models, but only when class instances are well mixed [9,8]. Z˘liobaitė et al. introduced an active learning algorithm that balances both stochastic sampling with model based uncertainty sampling in order to simultaneously address *both* changes to $p(\vec{x})$ and $p(y|\vec{x})$; moreover, this is achieved within fixed label budgets. Such an algorithm combines (model driven) uncertainty sampling with random sampling. Additionally, the active learning approach was sufficiently generic to be deployed with both the streaming formulations for Naive Bayes and Hoeffding decision tree models of classification.

Several recent works investigate the issue of class imbalance under streaming data contexts. One approach is to adopt a formulation of bagging with under or oversampling in order to construct a 'Data Subset' from which model building is performed [13,14]. Specifically, Ditzler et al. emphasizes operation under an incremental (i.e., batch) updating while also supporting anytime labelling, whereas work by Wang et al. emphasize operating under an online (i.e., record-wise) updating constraint. Also of note is that even though Ditzler et al. assumed the SMOTE algorithm developed for operating under stationary data with class imbalance [15], this was not the most effective method investigated for operation under non-stationary data [13]. A second general approach adopted for addressing class imbalance under streaming data is the use of dynamically reweighing class costs [16,17], where this has also been reported under an active learning context [18]. Most recently, attention has been paid to scenarios in which classes repeatedly drop in and out from the stream against a general backdrop of classes that appear on a continuous basis [19], albeit with true labels known for the entire stream content. In the application context associated with this work, the generic number of classes is known (e.g. attack or normal), but when they might appear is not.

Various semi-supervised frameworks have also been proposed for operation under streaming data contexts, and provide a natural framework for addressing labelled versus unlabeled data [20,21], i.e. after an initial period of training from labelled data, the classifiers go 'online' using unsupervised learning alone. Specific points of interest include operation under class imbalance and non-stationary data. However, from the perspective of this work, operation online using unsupervised learning would make such an approach particularly susceptible to adversarial attackers [5,6].

[2] Predictions from the anytime classifier might also be used to prioritize the records identified for labelling, i.e. a record predicted as an attack class would be prioritized over a record carrying a normal class prediction.