Information Sciences 294 (2015) 95–108

Contents lists available at ScienceDirect

Information Sciences

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

A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks

Kelton A.P. Costa [a], Luis A.M. Pereira [b], Rodrigo Y.M. Nakamura [a], Clayton R. Pereira [c], João P. Papa [a,*], Alexandre Xavier Falcão [b]

[a] UNESP – Univ Estadual Paulista, Department of Computing, Bauru, Brazil
[b] UNICAMP – University of Campinas, Institute of Computing, Campinas, Brazil
[c] UFSCar – Univ Federal of São Carlos, Department of Computing, São Carlos, Brazil

ARTICLE INFO

Article history:
Received 1 July 2013
Received in revised form 18 August 2014
Accepted 7 September 2014
Available online 5 October 2014

Keywords:
Intrusion detection
Optimum-path forest
Meta-heuristic
Clustering

ABSTRACT

We propose a nature-inspired approach to estimate the probability density function (pdf) used for data clustering based on the optimum-path forest algorithm (OPFC). OPFC interprets a dataset as a graph, whose nodes are the samples and each sample is connected to its k-nearest neighbors in a given feature space (a k-nn graph). The nodes of the graph are weighted by their pdf values and the pdf is computed based on the distances between the samples and their k-nearest neighbors. Once the k-nn graph is defined, OPFC finds one sample (root) at each maximum of the pdf and propagates one optimum-path tree (cluster) from each root to the remaining samples of its dome. Clustering effectiveness will depend on the pdf estimation, and the proposed approach efficiently computes the best value of k for a given application. We validate our approach in the context of intrusion detection in computer networks. First, we compare OPFC with data clustering based on k-means, and self-organization maps. Second, we evaluate several metaheuristic techniques to find the best value of k.

1. Introduction

Intrusion constitutes a serious problem in computer networks, and may challenge system administrators to prevent unauthorized access to confidential and privileged information. Consequently, intrusion detection systems (IDS) have been developed to scan the network activity, and also to detect intrusion attacks.

Different sorts of monitoring approaches can be highlighted in the context of security management in computer networks. The reader can be referred to *anomaly* and *misuse* detection techniques, in which the former approaches are trained with information from normal access, and when a new sample (internet package, for instance) comes to be analyzed, the system tries to fit it in the normal access model: if it does not fit, the sample is then classified as an attack. In the opposite side, misuse-based techniques are trained with intrusion (attack) samples, and any sample that does not fit in this model is then classified as a normal access [23].

[*] Corresponding author.

E-mail addresses: kelton@fc.unesp.br (K.A.P. Costa), luis.pereira@ic.unicamp.br (L.A.M. Pereira), rodrigo.mizobe@fc.unesp.br (R.Y.M. Nakamura), clayton.pereira@dc.ufscar.br (C.R. Pereira), papa@fc.unesp.br (J.P. Papa), afalcao@ic.unicamp.br (A. Xavier Falcão).

http://dx.doi.org/10.1016/j.ins.2014.09.025
0020-0255/© 2014 Elsevier Inc. All rights reserved.

Other monitoring networks are referred to network intrusion detection systems, which usually rely on machine learning algorithms that are able to recognize attacks and suspicious activities, being such approaches usually trained with samples from normal access and attacks. Supervised pattern recognition techniques [18] have been extensively evaluated in this context. Examples are artificial neural networks with multi-layer perceptrons in [14,3,5], self-organization maps in [53,21,24], and support vector machines in [7,25,16]. Moreover, Wang and Wu [45] proposed the combination of entropy analysis and Holt-Winters estimation to detect intrusions, and Jucá et al. [19] proposed an approach inspired on the human immunological system to predict network intrusion. Some recent works have emphasized the importance of intrusion detection in cloud environments [28,32], as well as anomaly detection [1,39] and adversarial attacks [9]. An interesting review about intrusion detection systems is reported by Liao et al. [26].

Although such methods can achieve good results, labeled data are not easily available due to the high cost of manual annotation, highlighting the need for unsupervised pattern recognition techniques. Zhong et al. [54], for instance, compared k-means, mixture-of-spherical Gaussians, self-organizing maps, and neural-gas in network intrusion detection. Portnoy et al. [34] employed some traditional clustering techniques for anomaly detection, and Ye and Li [52] proposed an incremental clustering algorithm for the same task.

Guan and Ghorbani [15] presented a clustering technique to handle intrusion detection, called Y-means, and Eskin [10] addressed anomaly detection by learning the probability distribution of the samples. Chaki and Chaki [6] addressed intrusion detection in mobile networks, and Sen [40] presented a distributed intrusion detection architecture for wireless-based ad hoc networks. Recently, Wu and Banzhaf [46] presented an interesting review about supervised and unsupervised methods applied to intrusion detection in computer networks.

Rocha et al. [37] proposed a data clustering algorithm that interprets the dataset as a graph, whose nodes are the samples and each sample is connected to its k-nearest neighbors in a given feature space (a k-nn graph). The nodes of the graph are weighted by their pdf values and the pdf is computed based on the distances between the samples and their k-nearest neighbors. Once the k-nn graph is defined, the algorithm finds one sample (root) at each maximum of the pdf and propagates one optimum-path tree (cluster) from each root to the remaining samples of its dome. Since the result is a collection of optimum-path trees, this approach is known as Optimum-Path Forest Clustering (OPFC). However, clustering effectiveness in OPFC depends on the estimation of the pdf, which relies on the parameter k. The original version employs an exhaustive search for the best value of k within a given interval $[k_{\min}, k_{\max}]$. Exhaustive search is prohibitive for datasets with millions of samples, which is the case of intrusion detection in computer networks. Therefore, we address this optimization problem by proposing a nature-inspired approach.

Nature-inspired optimization techniques can be easily implemented and have been successfully applied to a wide range of applications. In order to find the best value of k for pdf estimation, we evaluate Bat Algorithm (BA) [49–51], Firefly Algorithm (FFA) [48,12], Gravitational Search Algorithm (GSA) [36], Harmony Search (HS) [13], and Particle Swarm Optimization (PSO) [47]. As main contributions, this paper: (i) addresses intrusion detection in computer networks using OPFC, and (ii) proposes a considerable speed up for this algorithm by estimating the best value of k through evolutionary-based optimization techniques.

Although the supervised version of the OPF classifier [30,29] has been already employed to intrusion detection in computer networks [33], the unsupervised OPF has been poorly evaluated in this context so far. Therefore, this work can contribute with this lack of research, as well as to provide to the reader more insights about the OPF-based classification process. The remainder of the paper is organized as follows. In Sections 2 and 3 we review the optimum-path forest clustering and the metaheuristic algorithms used in this paper. Section 4 presents a detailed definition of the proposed approach. Section 5 describes the experimental settings used to validate our approach, while Section 6 discusses the results. Finally, we conclude the paper in Section 7 by analyzing the implications of this study.

2. Optimum-path forest clustering

The design of pattern classifiers based on *Optimum-Path Forest* (OPF) has been proposed as a graph-based methodology to exploit connectivity relations between data samples in a given feature space. The methodology interprets a training set as a graph, whose nodes are the samples and the arcs connect pairs of samples that satisfy a given *adjacency relation*. For a suitable *path-value (connectivity) function*, the optimum-path forest algorithm [11] partitions the graph into optimum-path trees rooted at some key samples, named *prototypes*. The prototypes compete among themselves for the most closely connected samples in the training set, such that each sample is assigned to the tree whose prototype offers to it an optimum path. Classification of a new sample is done by finding its most closely connected root in an incremental way through the evaluation of the optimum-path values of the training samples.

The OPF methodology has been exploited for supervised [30] and unsupervised [37] learning and successfully applied to several applications [29,4,38]. In this context, the choice of prototypes, adjacency relation, and connectivity function constitutes different classifiers. We improve the unsupervised version [37] in this work, as described next.

Let $\mathcal{Z}$ be a dataset such that for every sample $s \in \mathcal{Z}$ there exists a feature vector $\vec{v}(s)$. Let $d(s,t)$ be the distance between s and t in the feature space. For instance, $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$ — the Euclidean distance between $\vec{v}(t)$ and $\vec{v}(s)$. A graph $(\mathcal{Z}, \mathcal{A}_k)$ can be defined such that the arcs $(s,t) \in \mathcal{A}$ connect k-nearest neighbors in the feature space. The arcs are weighted by $d(s,t)$ and the nodes $s \in \mathcal{Z}$ are weighted by a probability density value $\rho(s)$: