Anomaly Detection in Computer Networks

The present invention relates to the detection of anomalies in communication on computer networks.

Increasing threats to network connected devices such as computer systems and networking appliances heighten a need to identify potentially threatening, malicious, erroneous or suspicious network traffic communicated via a computer network. Such traffic can be said to be anomalous where it is not consistent with traffic arising in the normal, accepted and/or expected operation of network connected devices.

Anomaly detection largely relies on feature extraction using techniques such as principal component analysis and mechanisms for learning normal network traffic characteristics, such as through the use of one-class support vector machines (SVMs). However, such models of normal traffic can suffer from overfitting relatively variable network communication (such as might be expected in the content of traffic communicated to a domain name server, for example) so reducing sensitivity to anomalies. Alternatively, such models can underfit network traffic by reflecting relatively consistent traffic such that sensitivity to anomalies is excessive and includes an excessive number of false-positive determinations of anomalous traffic.

Thus, there is a need to address these challenges while improving the identification of anomalies in network traffic.

The present invention accordingly provides, in a first aspect, a method of anomaly detection for network traffic communicated by devices via a computer network, the method comprising: clustering a set of time series, each time series including a plurality of time windows of data corresponding to network communication characteristics for a device; training an autoencoder for each cluster based on time series in the cluster; generating a set of reconstruction errors for each autoencoder based on testing the autoencoder with data from time windows of at least a subset of the time series; generating a probabilistic model of reconstruction errors for each autoencoder; and generating an aggregation of the probabilistic models for, in use, detecting reconstruction errors for a time series of data corresponding to network communication characteristics for a device as anomalous.

Preferably, the clusters are defined based on an autoencoder for converting each time series to a vector of features for the time series and a clustering algorithm clusters the vectors.

Preferably, the set of reconstruction errors for an autoencoder are generated based on the autoencoder processing each time series in a corresponding cluster of time series.

Preferably, the clustering are defined based on a random subdivision of the set of time series.

Preferably, the set of reconstruction errors for an autoencoder are generated based on the autoencoder processing each of the time series.

Preferably, each probabilistic model is a Gaussian model of reconstruction errors for an autoencoder.

Preferably, the aggregation of the probabilistic models is a Gaussian mixture model.

Preferably, the aggregation of the probabilistic models is a hidden Markov model.

The present invention accordingly provides, in a second aspect, a computer system including a processor and memory storing computer program code for performing the steps of the method set out above.

The present invention accordingly provides, in a third aspect, a computer program element comprising computer program code to, when loaded into a computer system and executed thereon, cause the computer to perform the steps of the method set out above.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram illustrating computer systems executing in virtualised computing environments under control of a botnet controller;

Figure 2 is a component diagram of an arrangement for detecting anomalies in network traffic according to embodiments of the present invention; and

Figure 3 is a flowchart of a method of anomaly detection for network traffic communicated by devices via a computer network according to embodiments of the present invention.

Figure 1 is a block diagram of a computer system suitable for the operation of embodiments of the present invention. A central processor unit (CPU) 102 is communicatively connected to a storage 104 and an input/output (I/O) interface 106 via a data bus 108. The storage 104 can be any read/write storage device such as a random access memory (RAM) or a non-volatile storage device. An example of a non-volatile storage device includes a disk or tape storage device. The I/O interface 106 is an interface to devices

for the input or output of data, or for both input and output of data. Examples of I/O devices connectable to I/O interface 106 include a keyboard, a mouse, a display (such as a monitor) and a network connection.

Figure 2 is a component diagram of an arrangement for detecting anomalies in network traffic according to embodiments of the present invention. A compute network 200, such as a wired, wireless, local, wide-area or any other suitable network, has communicatively connected devices 202a-202n such as, inter alia, computer systems, network appliances, pervasive devices, sensors, detectors, virtual computer systems etc. For example, devices 202a-202n can include one or more network appliances such as: a proxy; a firewall; a domain name server (DNS); a router; a gateway; a software appliance such as an intrusion detection and/or protection service; and other such appliances as will be familiar to those skilled in the art.

In use, devices communicate via the network 200 using one or more protocols for network communication. A network analyser 204 is a hardware, software, firmware or combination component adapted to access and store information about network communication via the network 200. For example, NetFlow is a network facility developed by Cisco for collecting internet protocol (IP) traffic information and monitoring network traffic. Thus, the network analyser 204 can identify one or more of: ingress/egress interface information for a communication; source/destination network address information such as IP address information; protocol information; port information; network protocol service information; network communication metrics such as size, duration, numbers of packets, packet sizes etc.; and other information as will be apparent to those skilled in the art. Alternative network analysis methods can be employed including bespoke analysis. The network analyser 204 associates network data with a device 202a-202n connected with the network such that characteristics of the network data for the device can be determined. Example characteristics can include: a number of outgoing connections from the device; a number of distinct ports for outgoing connections; an average duration of a connection; an average number of bytes exchanged; and other characteristics as will be apparent to those skilled in the art. Such characteristics can be selected to reflect promising basis for identifying malicious, erroneous, or suspicious communications, for example.

A time series generator 206 is a hardware, software, firmware or combination component for generating a time series of network characteristics for each of a plurality of network connected devices 202. Each time series is defined by grouping network characteristics for each of a series of fixed length time windows, most preferably consecutive time windows, for each of which a set of network characteristics are identified based on the output of the

network analyser 204. Thus, a set of time series are generated, each for a different device 202, and each comprising characteristics over fixed length time windows.

A clustering process 208 is performed to cluster the set of time series into a plurality of clusters each constituting a subset of the set. In one embodiment, each cluster is defined based on a random division of the set of time series. In a preferred embodiment, each cluster is defined based on an autoencoder as input to a clustering algorithm such as k-means. For example, an autoencoder can be employed to convert a time series to a feature vector on which basis clustering is performed. Thus, for the set of time series each time series can be converted to a feature vector as input to a clustering algorithm such as k-means. In this way time series with common features determined by the autoencoder can be clustered together. In one embodiment, such clustering results in devices 202 having similar network communication characteristics being clustered together.

An autoencoder trainer 210 is a hardware, software, firmware or combination component for training an autoencoder 212 for each cluster defined by the clustering process 208 such that each cluster has a separately trained autoencoder 212. Thus, an autoencoder 212 is trained for a cluster based on each of the time series in the cluster on a time window by time window basis. The autoencoder trainer 210 operates on the basis of time series generated as training data, such as time series defined on the basis of network communication that is known to reflect normal, typical, non-suspicious and/or safe communication unencumbered by malicious, erroneous or suspicious network traffic or entities. Thus, such time series can be referred to as training time series.

Subsequently, an autoencoder tester 216, as a hardware, software, firmware or combination component, applies time series from the training time series to each autoencoder 212 to identify a set of reconstruction errors for each autoencoder 212. Thus, a time series from the training time series can be processed by an autoencoder to determine an accuracy of reconstruction (by backpropagation) of the autoencoder and a deviation from an accurate reconstruction constitutes a reconstruction error. A set of reconstruction errors occurring across all of the time series processed by an autoencoder 212 is subsequently used to define a statistical model such as a probabilistic model 218 of reconstruction errors for the autoencoder 212.

In one embodiment, the time series processed by an autoencoder 212 by the autoencoder tester 214 is only the time series included in a cluster corresponding to the autoencoder tester 214 (as indicated by the solid line between the clustering process 208 and the autoencoder tester 214). Such an approach is especially appropriate where each cluster is defined on the basis of feature vectors defined themselves by an autoencoder process. In an

alternative embodiment, all time series in the training time series can be processed by all autoencoders 212 (as indicated by the broken line between the clustering process 208 and the autoencoder tester 214). This thus provides cross-validation whereby metrics can be evaluated for each autoencoder such as a highest/lowest reconstruction error or an examination of the distribution of reconstruction errors as a basis for defining the probabilistic models 218 for each autoencoder 212.

The reconstruction error information generated by the autoencoder tester 214 is processed by a statistical model generator 216 to generate a probabilistic model 218 for each autoencoder. The statistical model generator 216 is a a hardware, software, firmware or combination component for generating a probabilistic model 218 for an autoencoder 212 based on a set of reconstruction errors 214 and/or some summary or characteristics of reconstruction errors provided by the autoencoder tester 214 and/or determined by the statistical model generator 216. For example, in one embodiment, a Gaussian (e.g. normal distribution) is defined each autoencoder 212 based on reconstruction error information for the autoencoder 212.

Subsequently, an aggregator 220 is a hardware, software, firmware or combination component for aggregating the probabilistic models 218 into an aggregate model 222. For example, the aggregate model 222 can be a Guassian mixture model as will be apparent to those skilled in the art. In an alternative embodiment, the statistical model generator 216 and the aggregator 220 are adapted to generate a Hidden Markov Model.

The aggregate model 222 thus statistically models reconstruction errors for all autoencoders 212 for the training time series and thus can be described as modelling “normal” communication of the devices 202 via the network 200, where “normal” communication is known non-suspicious, non-malicious and/or non-erroneous communication.

Subsequently, on the basis of the aggregate model 222, an anomaly detector 226 is configured to detect an anomaly in network traffic for a device 202 on the network 200. Network traffic (now in a production mode of operation, and thus not part of the traffic used for training) is accessed and/or received by the network analyser 204 and a time series (a “production” time series) is generated for it by the time series generator 206. The production time series for this network traffic is then received by the anomaly detector 224 which invokes one or more autoencoders 212 for each time window of the time series to determine reconstruction errors for the traffic. The autoencoder(s) 212 invoked can be either a specific autoencoder 212 identified based on an identification of an appropriate cluster for the production time series (e.g. by an application of the clustering process 208 to the time series

on the basis of the cluster definitions for the training time series) or can be all autoencoders 212. In particular, where the clustering process for training time series is based on a feature vector determined by an autoencoder, the production time series is preferably processed by an appropriate autoencoder determined based on the same clustering process (to determine an appropriate cluster for the production time series).

Thus, the anomaly detector 224 determines reconstruction error(s) for the production time series (for each time window) and compares these errors with the aggregate model 222 of reconstruction errors to determine if there is a distance exceeding a predetermined threshold. Where such distance between model and actual reconstruction errors exceeds the threshold then an anomaly is identified and reported. Such anomalies can trigger reactive action such as: an identification of one or more devices 202 involved in anomalous communication; preventing a device 202 from communication; disconnecting a device 202; tracking a device 202; increasing a degree of monitoring or scrutiny of a device 202; and other reactive actions as will be apparent to those skilled in the art.

Thus, network traffic, such as data generated by Netflow tools, can be used to generate time-series network characteristics. A multi-phase approach to anomaly detection is employed according to embodiments of the present invention. Traffic is constituted as a time-series on a per-device (e.g. host) basis for each of a series of time windows. An autoencoder can then be employed to inform a clustering algorithm (such as k-means) to separate traffic into clusters. In one embodiment, such clusters can constitute sets of like-devices (e.g. workstations, routers, DNS servers and the like) such that devices having common traffic features are clustered together. Time-series data for each cluster is subsequently used to train a cluster-specific autoencoder. The time-series data for a particular device in a particular time window is processed by a corresponding autoencoder to determine a reconstruction error of the autoencoder for the time-series data.

According to common understanding of those skilled in the art, a large reconstruction error could be considered an indicator of anomalous time-series data. However, this is not necessarily the case for time-series data that is unusual but normal, such as data arising from a DNS appliance. Accordingly, embodiments of the present inventio employ a statistical model of reconstruction errors generated by the autoencoders. For example, a Gaussian probability distribution of reconstruction errors can be applied such that multiple appliances in a cluster can generate a Gaussian, the combination of which for a plurality of clusters constitutes a Gaussian mixture model. Comparing a reconstruction error for a host in a time period with the Gaussian mixture model offers an opportunity to identify a disparity and an extent of that disparity between data over consecutive time periods for a host and known

normal data represented by the Gaussians in the Gaussian mixture model. Disparity exceeding a predetermined threshold can therefore be identified as an anomaly.

Figure 3 is a flowchart of a method of anomaly detection for network traffic communicated by devices via a computer network according to embodiments of the present invention. Initially, at step 302, a set of training time series is clustered by a clustering process 208. At step 304 an autoencoder 212 is trained for each cluster based on each of a plurality of time windows of each training time series in the cluster. At step 306 reconstruction errors for each autoencoder are generated based on the training time series. At step 308 a probabilistic model is generated for each autoencoder. At step 310 an aggregation of the probabilistic models is generated such that, in use for production time series, reconstruction errors for the production time series can be detected as anomalous based on the aggregation of probabilistic models.

Insofar as embodiments of the invention described are implementable, at least in part, using a software-controlled programmable processing device, such as a microprocessor, digital signal processor or other processing device, data processing apparatus or system, it will be appreciated that a computer program for configuring a programmable device, apparatus or system to implement the foregoing described methods is envisaged as an aspect of the present invention. The computer program may be embodied as source code or undergo compilation for implementation on a processing device, apparatus or system or may be embodied as object code, for example.

Suitably, the computer program is stored on a carrier medium in machine or device readable form, for example in solid-state memory, magnetic memory such as disk or tape, optically or magneto-optically readable memory such as compact disk or digital versatile disk etc., and the processing device utilises the program or a part thereof to configure it for operation. The computer program may be supplied from a remote source embodied in a communications medium such as an electronic signal, radio frequency carrier wave or optical carrier wave. Such carrier media are also envisaged as aspects of the present invention.

It will be understood by those skilled in the art that, although the present invention has been described in relation to the above described example embodiments, the invention is not limited thereto and that there are many possible variations and modifications which fall within the scope of the invention.

The scope of the present invention includes any novel features or combination of features disclosed herein. The applicant hereby gives notice that new claims may be formulated to such features or combination of features during prosecution of this application or of any such

further applications derived therefrom. In particular, with reference to the appended claims, features from dependent claims may be combined with those of the independent claims and features from respective independent claims may be combined in any appropriate manner and not merely in the specific combinations enumerated in the claims.

CLAIMS

1. A method of anomaly detection for network traffic communicated by devices via a computer network, the method comprising:

clustering a set of time series, each time series including a plurality of time windows of data corresponding to network communication characteristics for a device;

training an autoencoder for each cluster based on time series in the cluster;

generating a set of reconstruction errors for each autoencoder based on testing the autoencoder with data from time windows of at least a subset of the time series;

generating a probabilistic model of reconstruction errors for each autoencoder; and

generating an aggregation of the probabilistic models for, in use, detecting reconstruction errors for a time series of data corresponding to network communication characteristics for a device as anomalous.

2. The method of claim 1 wherein the clusters are defined based on an autoencoder for converting each time series to a vector of features for the time series and a clustering algorithm clusters the vectors.

3. The method of any preceding claim wherein the set of reconstruction errors for an autoencoder are generated based on the autoencoder processing each time series in a corresponding cluster of time series.

4. The method of claim 1 wherein the clustering are defined based on a random subdivision of the set of time series.

5. The method of claim 4 wherein the set of reconstruction errors for an autoencoder are generated based on the autoencoder processing each of the time series.

6. The method of any preceding claim wherein each probabilistic model is a Gaussian model of reconstruction errors for an autoencoder.

7. The method of claim 6 wherein the aggregation of the probabilistic models is a Gaussian mixture model.

8. The method of any of claims 1 to 5 wherein the aggregation of the probabilistic models is a hidden Markov model.

9. A computer system including a processor and memory storing computer program code for performing the steps of any preceding claim.

10. A computer program element comprising computer program code to, when loaded into a computer system and executed thereon, cause the computer to perform the steps of a method as claimed in any of claims 1 to 8.

ABSTRACT

Anomaly Detection in Computer Networks

A method of anomaly detection for network traffic communicated by devices via a computer network, the method comprising: clustering a set of time series, each time series including a plurality of time windows of data corresponding to network communication characteristics for a device; training an autoencoder for each cluster based on time series in the cluster; generating a set of reconstruction errors for each autoencoder based on testing the autoencoder with data from time windows of at least a subset of the time series; generating a probabilistic model of reconstruction errors for each autoencoder; and generating an aggregation of the probabilistic models for, in use, detecting reconstruction errors for a time series of data corresponding to network communication characteristics for a device as anomalous.

(Figure 2)

FIGURE 1

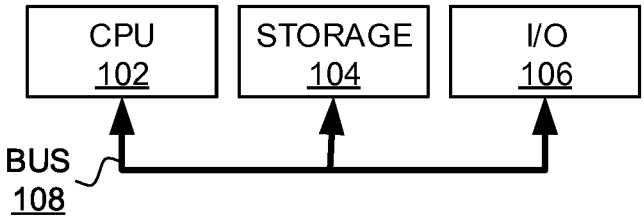


FIGURE 2

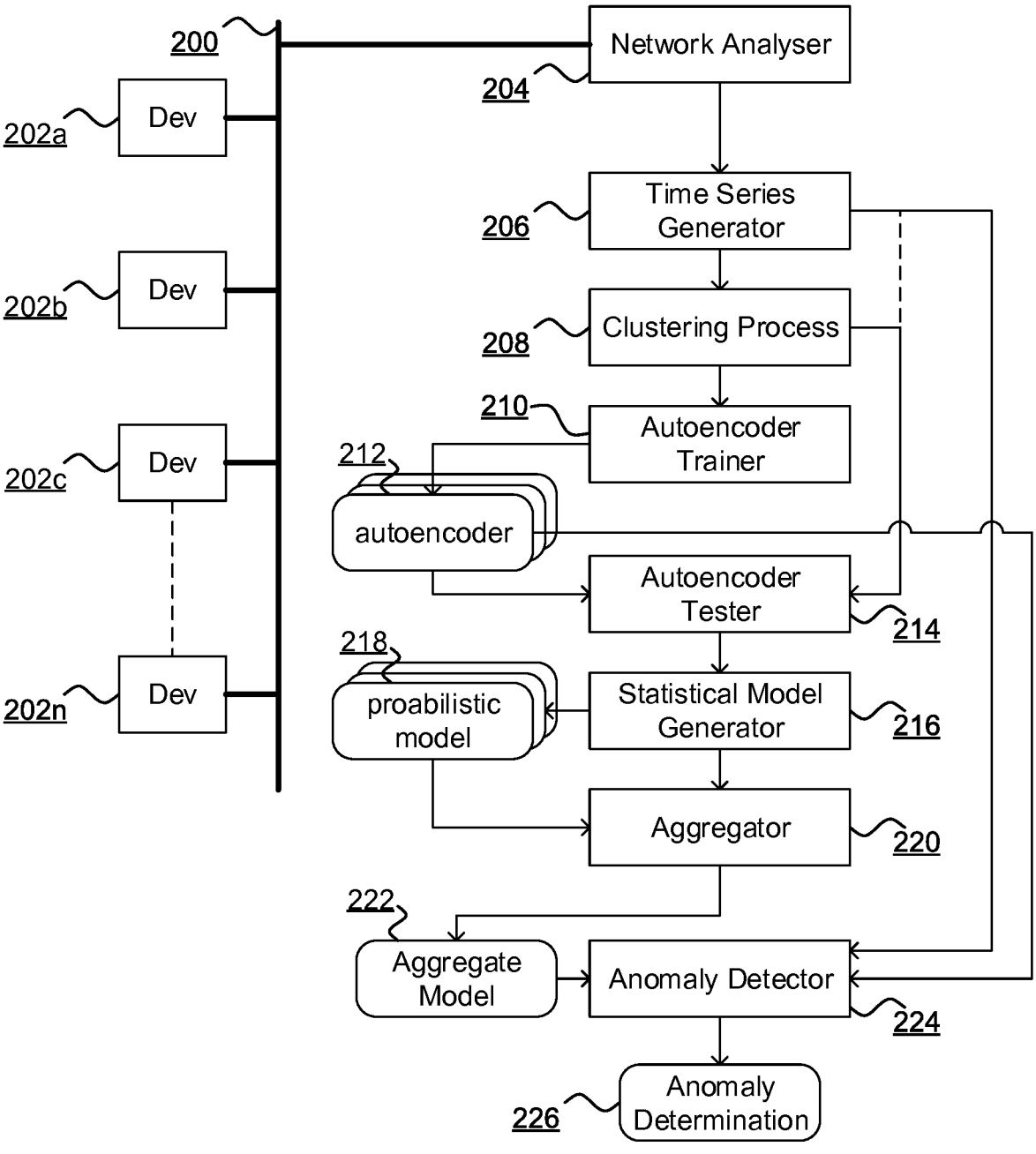
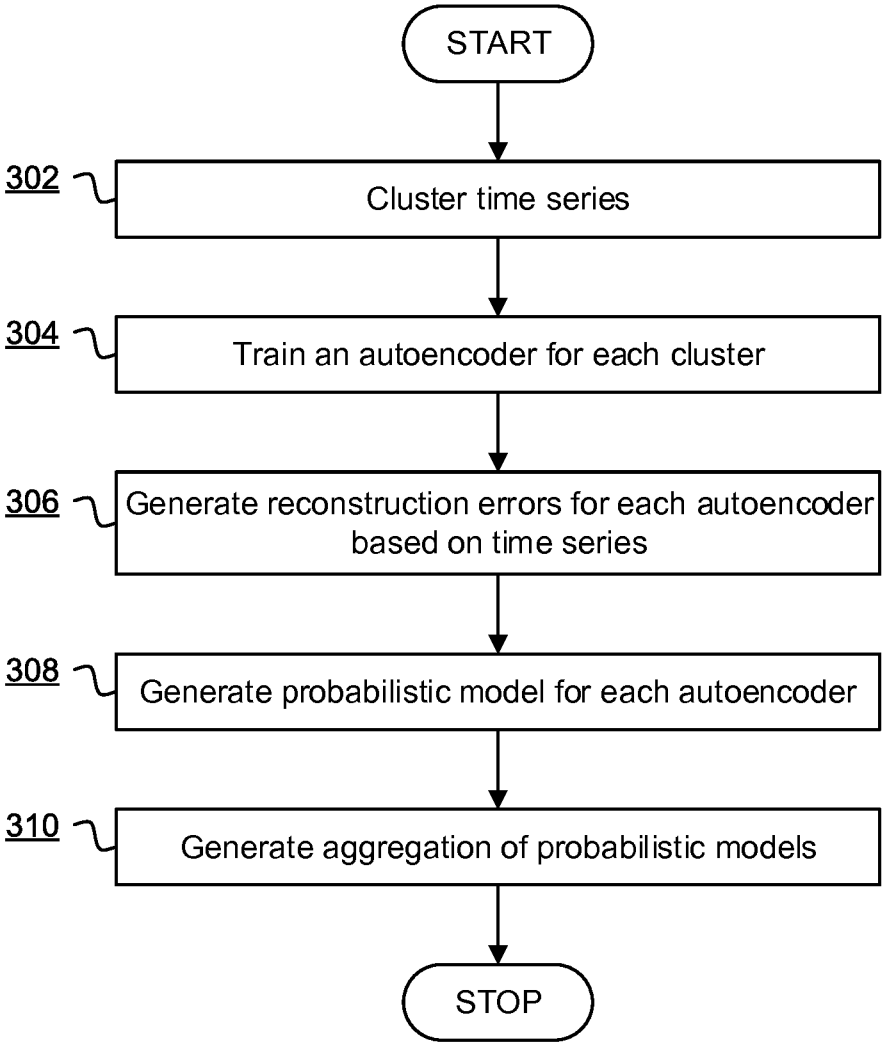


FIGURE 3


START
302
Cluster time series
304
Train an autoencoder for each cluster
306
Generate reconstruction errors for each autoencoder based on time series
308
Generate probabilistic model for each autoencoder
310
Generate aggregation of probabilistic models
STOP