ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

The R Package anomalyDetection an environment for cyber analytics

B.Annapurna,
Head of the Department Computers Science,
annapurnagandrety@gmail.com.
Ch.S.D.St. Theresa's Autonomous College for Women
Eluru, Andhra Pradesh, IndiaEluru, Andhra Pradesh, India

Dr.K.S.V.K.S. Madhavi Rani
Department of Zoology,
madaviraniksvks@gmail.com.
Ch.S.D.St. Theresa's Autonomous College for Women

Mr.B.Srinivasa Rao
Department of Mathematics
Srinivasrao.banda@gmail.com
Sir C.R.R.Autonomous College,
Eluru, Andhra Pradesh, India

Abstract: Cyber-attacks are socially or politically motivated attacks carried out primarily through the Internet. Attacks target the general public or national and corporate organizations and are carried out through the spread of malicious programs (viruses), unauthorized web access, fake websites, and other means of stealing personal or institutional information from targets of attacks, causing far-reaching damage. A massive **ransomware** attack has hit businesses around the world, causing major companies to shut down their computer systems.We can use R programming to detect anomalies in a dataset. Anomaly detection can be used in a number of different areas, such as intrusion detection, fraud detection, system health, and so on. In R programming, these are called outliers.

Introduction:anomalyDetection is an open-source R package to detect anomalies which is robust, from a statistical standpoint, in the presence of seasonality and an underlying trend. The anomalyDetection package can be used in wide variety of contexts. Example, detecting anomalies in system metrics after a new software release, for problems in econometrics, financial engineering, political and social sciences.

Working: The underlying algorithm – referred to as Seasonal Hybrid ESD (S-H-ESD) builds upon the Generalized ESD test for detecting anomalies. S-H-ESD can be used to detect both global as well as local anomalies. This is achieved by employing time series decomposition and using robust statistical metrics, viz., median together with ESD. In addition, for long time series, the algorithm employs piecewise approximation - this is rooted to the fact that trend extraction in the presence of anomalies in non-trivial - for anomaly detection.

Besides time series, the package can also be used to detect anomalies in a vector of numerical values. We have found this very useful as many times the corresponding timestamps are not available. The package provides rich visualization support. The user can specify the direction of anomalies, the window of interest (such as last day, last hour), enable/disable piecewise approximation; additionally, the x- and y-axis are annotated in a way to assist visual data analysis.

Anomaly Detection Techniques

Anomaly detection can be approached in many ways depending on the nature of data and circumstances. Following is a classification of some of those techniques.

Static Rules Approach: Most simple, and may be the best approach to start with, is using static rules. The Idea is to identify a list of known anomalies and then write rules to detect those anomalies. Rules identification is done by a domain expert, by using pattern mining techniques, or a by combination of both.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

Static rules are used with the hypothesis that anomalies follow the 80/20 rule where most anomalous occurrences belong to few anomaly types. If the hypothesis is true, then we can detect most anomalies by finding few rules that describe those anomalies.Implementing those rules can be done using one of three following methods.If they are simple and no inference is needed, you can code them using your favourite programming language.

If decisions need inference, then you can use a rule-based or expert system (e.g. Drools) If decisions have temporal conditions, you can use a Complex Event Processing System (e.g. WSO2 CEP, Esper). Although simple, static rules based systems tend to be brittle and complex. Furthermore, identifying those rules is often a complex and subjective task. Therefore, statistical or machine learning based approach, which automatically learn the general rules, are preferred to static rules. [4] discussed about a method, End-to-end inference to diagnose and repair the data-forwarding failures, our optimization goal to minimize the faults at minimum expected cost of correcting all faulty nodes that cannot properly deliver data. First checking the nodes that has the least checking cost does not minimize the expected costin fault localization. We construct a potential function for identifying the candidate nodes, one of which should be first checked by an optimal strategy. We proposes efficient inferring approach to the node to be checked in large-scale networks.

When we have Training Data :Anomalies are rare under most conditions. Hence, even when training data is available, often there will be few dozen anomalies exists among millions of regular data points. The standard classification methods such as SVM or Random Forest will classify almost all data as normal because doing that will provide a very high accuracy score (e.g. accuracy is 99.9 if anomalies are one in thousand).

Generally, the class imbalance is solved using an ensemble built by resampling data many times. The idea is to first create new datasets by taking all anomalous data points and adding a subset of normal data points (e.g. as 4 times as anomalous data points). Then a classifier is built for each data set using SVM or Random Forest, and those classifiers are combined using ensemble learning. This approach has worked well and produced very good results.

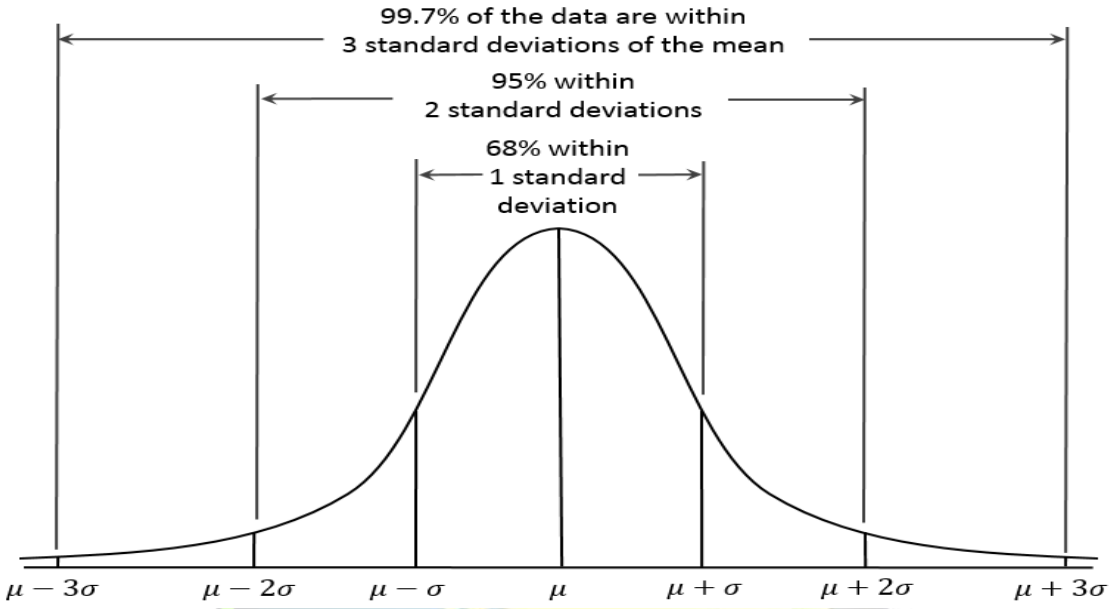
If the data points are auto correlated with each other, then simple classifiers would not work well. We handle those use cases using time series classification techniques or Recurrent Neural networks.

When there is no Training Data: If you do not have training data, still it is possible to do anomaly detection using unsupervised learning and semi-supervised learning. However, after building the model, you will have no idea how well it is doing as you have nothing to test it against. Hence, the results of those methods need to be tested in the field before placing them in the critical path.

No Training Data: Point Anomalies

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017



Point anomalies will only have a one field in the data set. We use percentiles to detect point anomalies with numeric data and histograms to detect Detecting point anomalies in categorical data. Either case, we find rare data ranges or field values from the data and predict those as anomalies if it happens again. For example, if 99.9 percentile of my transaction value is 800$, one can guess any transaction greater than that value as the potential anomaly. When building models, often we use moving averages instead of point values when possible as they are much more stable to noise.

No Training Data: Univariate Collective Outliers

Time series data are the best examples of collective outliers in a univariate dataset. In this case, anomalies happen because values occur in unexpected order. For example.the third heart beat might be anomalous not because values are out of range, but they happen in a wrong order.

There are three several approaches to handle these use cases.

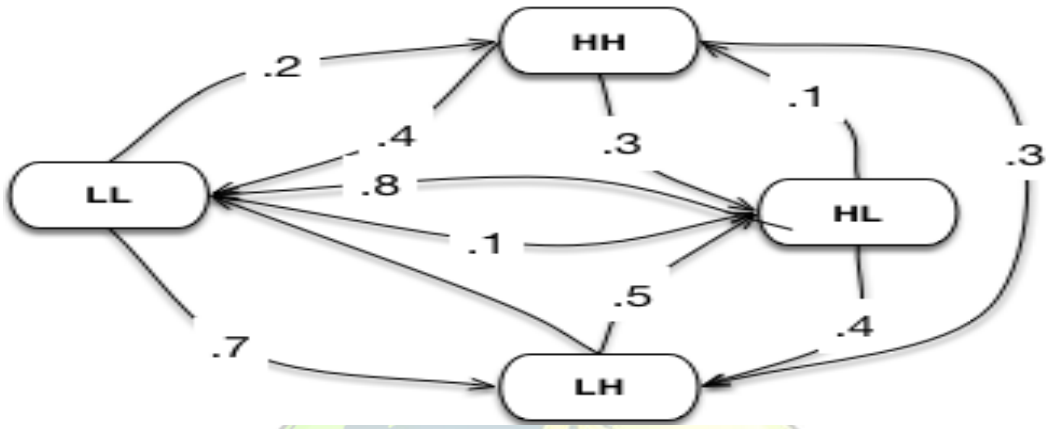
Solution 1: build a predictor and look for outliers using residues: This is based on the heuristic that the values not explained by the model are anomalies. Hence we can build a model to predict the next value, and then apply percentiles on the error (predicted value – actual value) as described before. The model can be built using regression, time series models, or Recurrent Neural Networks.

Solution 2: Markov chains and Hidden Markov chains can measure the probability of a sequence of events happening. This approach builds a Markov chain for the underline process, and when a sequence of events has

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

happened, we can use the Markov Chain to measure the probability of that sequence occurring, and use that to detect any rare sequences.



FraudFor example, let's consider credit card transactions. To model the transactions using Markov chains, let's represent each transaction using two values: transaction value (L, H) and time since the last transaction (L, H). Since Markov chain's states have to be finite, we will choose two values Low (L), High (H) to represent variable values. Then Markov chains would represent by states LL, LH, HL, HH and each transaction would be a transition from one state to another state. We can build the Markov chain using historical data and use the chain to calculate sequence probabilities. Then, we can find the probability of any new sequence happening and then mark rare sequences as anomalies. The blog post "Real Time Fraud Detection with Sequence Mining" describes this approach in detail.

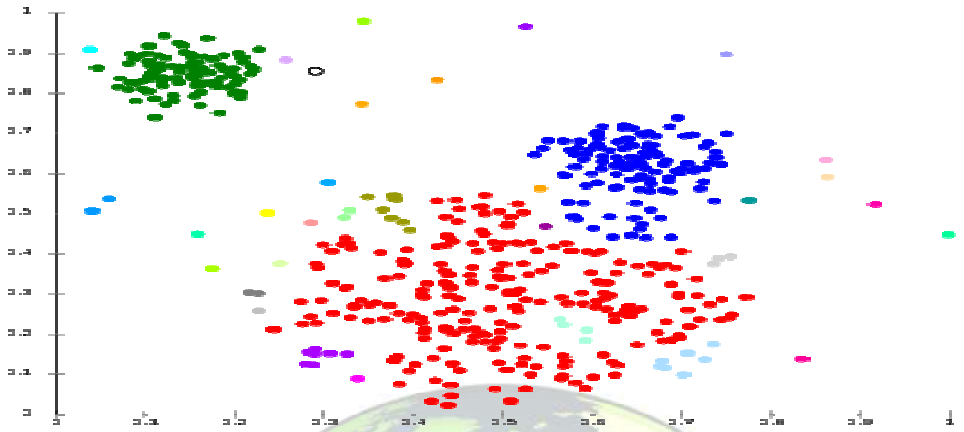
No Training Data: Multivariate Collective Outliers (Unordered)

Here data have multiple reading but does not have an order. For example, vitals collected from many people are such a multi-variate but not ordered dataset. For example, higher temperatures and slow heartbeats might be an anomaly even though both temperature and heartbeats by itself are in a normal range.

Approach 1: Clustering – the underline assumption in the first approach is that if we cluster the data, normal data will belong to clusters while anomalies will not belong to any clusters or belong to small clusters.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017



Then to detect anomalies we will cluster the data, and calculate the centroids and density of each cluster found. When we receive a new data point, we calculate the distance from the new data point to known large clusters and if it is too far, then decide it as an anomaly.

Furthermore, we can improve upon the above approach by first manually inspecting ranges of each cluster and labelling each cluster as anomalous or normal and use that while doing anomaly check for a data point.

Approach 2: Nearest neighbour techniques – the underline assumption is new anomalies are closer to known anomalies. This can be implemented by using distance to k-anomalies or using the relative density of other anomalies near the new data point. While calculating the above, with numerical data, we will break the space into hypercubes, and with categorical data, we will break the space into bins using histograms.Both these approaches are described in ACM Computing Survey paper "Anomaly Detection: A Survey" in detail.

No Training Data: Multivariate Collective Outliers (Ordered)

This class is most general and consider ordering as well as value combinations. For example, consider a series of vital readings taken from the same patient. Some reading may be normal in combination but anomalous as combinations happen in wrong order. For example, given a reading that has the blood pressure, temperature, and heart beat frequency, each reading by itself may be normal, but not normal if it oscillates too fast in a short period of time.

Combine Markov Chains and Clustering – This method combines clustering and Markov Chains by first clustering the data, and then using clusters as the states in a Markov Chain and building a Markov Chain. Clustering will capture common value combinations and Markov chains will capturing their order.

Installation

We can install anomalyDetection two ways.

Using the latest released version from CRAN:

install.packages("anomalyDetection")

Using the latest development version from GitHub:

if (packageVersion("devtools") < 1.6) {

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

install.packages("devtools")

}

devtools::install_github("AFIT-R/anomalyDetection", build_vignettes = TRUE)

To get started with anomalyDetection, read the intro vignette: vignette("Introduction", package = "anomalyDetection"). This will provide a thorough introduction to the functions provided in the package.

Functions in anomalyDetection

Name	**Description**
bd_row	Breakdown for Mahalanobis Distance
factor_analysis	Factor Analysis with Varimax Rotation
mc_adjust	Multi-Collinearity Adjustment
%>%	Pipe functions
principal_components	Principal Component Analysis
principal_components_result	Easy Access to Principal Component Analysis Results
horns_curve	Horn's Parallel Analysis
inspect_block	Block Inspection
factor_analysis_results	Easy Access to Factor Analysis Results
get_all_factors	Find All Factors
security_logs	Security Log Data
tabulate_state_vector	Tabulate State Vector
kaisers_index	Kaiser's Index of Factorial Simplicity
mahalanobis_distance	Mahalanobis Distance

Data processing scenario:

In this example, data comes from the well-knownwikipedia, which offers an API to download from R the daily page views given any {term + language}.

In this case, we've got page views from term “icc”, language en, from 2014-05-20 up to today.

Detecting Anamolies:

install.packages("devtools")

devtools::install_github("petermeissner/wikipediatrend")

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

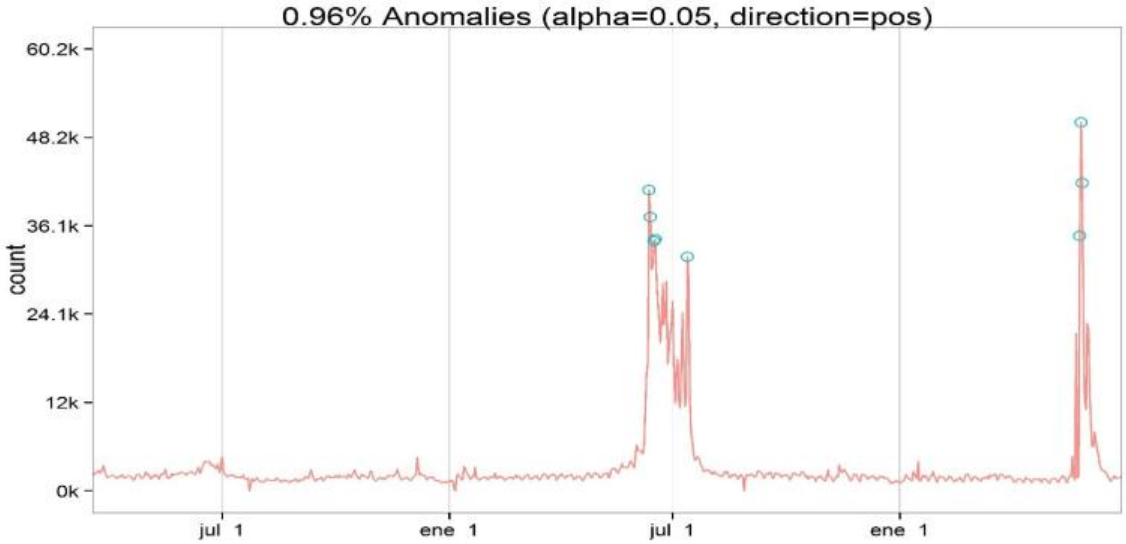
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

```
devtools::install_github("twitter/AnomalyDetection")

install.packages("Rcpp")


library(wikipediatrend) ## Library containing API wikipedia access

library(AnomalyDetection)

library(ggplot2)


## Download wiki webpage "icc"

icc_data = wp_trend("icc", from="2014-05-20", lang = "en")


## Plotting data

ggplot(icc_data, aes(x=date, y=count, color=count)) + geom_line()


## Convert date variable

icc_data$date = as.POSIXct(icc_data$date)


## Keep only desiered variables (date & page views)

icc_data=icc_data[,c(1,2)]

## Apply anomaly detection

data_anomaly = AnomalyDetectionTs(icc_data, max_anoms=0.01, direction="pos", plot=TRUE, e_value = T)

jpeg("03_icc_wikipedia_term_page_views_anomaly_detection.jpg", width= 8.25, height= 5.25, units="in",
res=500, pointsize = 4)

## Plot original data + anomalies points

data_anomaly$plot

dev.off()


## Calculate deviation percentage from the expected value

data_anomaly$anoms$perc_diff=round(100*(data_anomaly$anoms$expected_value-
data_anomaly$anoms$anoms)/data_anomaly$anoms$expected_value)


## Plot anomalies table

anomaly_table=data_anomaly$anoms
```

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

After applying the algorithm, we can plot the original time series plus the abnormal points in which the page views were over the expected value.



Algorithm

Parameters in algorithm are max_anoms=0.01 (to have a maximum of 0.01% outliers points in final result), and direction="pos" to detect anomalies over (not below) the expected value.As a result, 8 anomalies dates were detected. Additionally, the algorithm returns what it would have been the expected value, and an extra calculation is performed to get this value in terms of percentage perc_diff.

Anomalies Detection:

Last plot shows a line indicating linear trend over a specific period -clearly decreasing-, and two black circles. It's interesting to note that these black points were not detected by the algorithm because they are part of a decreasing tendency.A really nice shot by this algorithm since the focus on detections are on the changes of general patterns. Just take a look at the last detected point in that period, it was a peak that didn't follow the decreasing pattern.

References:

[1] https://github.com/pablo14/anomaly_detection_post
[2] https://iwringer.wordpress.com/2015/11/17/anomaly-detection-concepts-and-techniques/
[3] J. Breier and J. Branišová. Anomaly detection from log files using data mining techniques. In InformationScience and Applications, pages 449–457. Springer, 2015
[4] Christo Ananth, Mary Varsha Peter, Priya.M., Rajalakshmi.R., Muthu Bharathi.R., Pramila.E., "Network Fault Correction in Overlay Network through Optimality", International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Volume 2, Issue 8, August 2015, pp: 19-22
[5] https://doi.org/10.1007/978-3-662-46578-3_53.
[6] Boehmke and R. Gutierrez.anomalyDetection: Implementation of Augmented Network Log AnomalyDetection Procedures, 2017
[7] https://CRAN.R-project.org/package=anomalyDetection.