# Network Anomalies Detection by Unsupervised Activity Deviations Extraction

Christophe Maudoux, Selma Boumerdassi

# Network Anomalies Detection by Unsupervised Activity Deviations Extraction

Christophe Maudoux ⓘ
*CNAM/Cedric*
Paris, France

Selma Boumerdassi ⓘ
*CNAM/Cedric*
Paris, France

*Abstract*—More and more organizations are under cyberattacks. To prevent this kind of threats, it is essential to detect them upstream by highlighting *abnormal activities* within networks. This paper presents our anomalies detection approach that consists of aggregating pre-processed network flows into sectors. Then for each sector, data are split into equal time periods. Finally an unsupervised clustering algorithm is employed to extract that we called *activity deviations*. If a specific sector network activity for one specific period differs from others, it means that a network anomaly has been detected. Our experiments are based on a *real dataset* provided by a French mobile operator. With our proposed method, we have been able to detect anomalies corresponding to real crowded events like fire, soccer match or concert.

*Index Terms*—Geohash, anomalies detection, activity deviation, unsupervised machine learning, networks, aggregation, security

## I. INTRODUCTION

Each day, some organizations are targetted by cyberattacks. Many types of threat exist like DDoS (Distributed Deny of Service) [1], security breach exploits that steal personal data, spam campaigns that try to reach a higher account level by deceiving users, stealing credentials to perform horizontal scanning and to spread malware for further gain. One common feature that characterizes all these cyberthreats is that specific data flows are exchanged over networks. To defeat or mitigate these kinds of attack, it is essential to deploy systems able to detect suspicious flows by extracting anomalies in networks.

*Network anomalies* can be defined as a 'behaviour that deviates from what is normal, standard, or expected'. So to detect network anomalies, it means that you might have a concept of expected or normal behaviour. Some researches have been done on this topic by monitoring network traffic and analysing message headers or payload content [2]. Countermeasures mainly consist of analysing traffic flows by deploying network probes like Intrusion or Anomaly Detection Systems [3].

Anomaly detection is useful in identifying the patterns that are unexpected called *outliers*. Moreover, Machine Learning (ML) techniques are increasingly used to detect them. Several works are available in the literature. For example, authors in [4] proposed a ML algorithm for classifying heterogeneous network traffic and anomaly detection. Also, a recognition algorithm was used in order to validate the model. On the other side, authors in [5] proposed a solution based on both deep residual Conventional Neural Network (CNN) structure for dataset modelling and transfer learning for training the model to detect anomalies in Industrial Control Systems (ICS).

ML techniques can also be used to detect outliers in sensor data. Authors in [6] described several techniques of ML to detect outliers in IoT by classifying the existing techniques into different categories.

For our use case, we aim to detect behaviour changes relating to mobile application activity using an unsupervised ML technique and based on the fields described in Section III. To achieve this, we extracted that we call *activity deviations* by analysing network activity behaviours at a specific moment in time and for a particular network location.

This article presents our network anomalies detection approach using machine learning to extract deviations and detect outliers. Our paper is organized as follows. In Section II, we explain our proposed method. Section III describes the dataset that we used for performing our research that is based on *real mobile network data* and the provided raw flows. Section IV details our method implementation and summarizes obtained results. Finally, we suggest possible improvements and define our planned further works.

## II. UNSUPERVISED ACTIVITY DEVIATIONS EXTRACTION

Our proposed approach is composed of three phases *(i)* data parsing and aggregations performed by some *Perl parsers* *(ii)* activity deviations extracted by the unsupervised *X-Means* Machine Learning Algorithm (MLA) *(iii)* outliers detection computed by the *Local Outlier Factors* (LOF) algorithm.

### A. Data Parsing & Aggregation Processes

Due to the high number of sites and the huge amount of available flows, data cannot be directly and globally analysed. A pre-processing phase is needed. As depicted by Fig. 1, we have had to *(i)* aggregate raw network flows by sites: *by-site aggregation (ii)* followed by parsing them into sectors to extract the activity deviations: *by-sector aggregation*.

*1) By-site aggregation:* this process is used for transposing and aggregating raw network flows from the same *Base Transmitting Station* (BTS). Indeed, a BTS or site can serve several antennas where each antenna is defined by a unique identifier or location.

*2) By-sector aggregation:* this step aims to define some specific areas or *sectors* for further detailed analysis. The employed method consists of gathering data from neighbouring sites by using the *Geohash* concept [7].
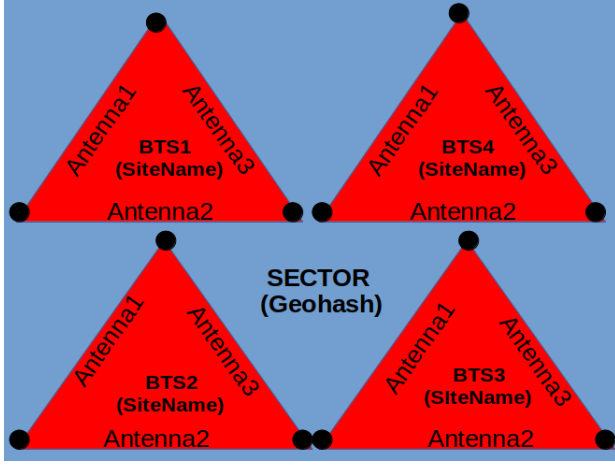
Fig. 1. Aggregations overview



(a) Precision 5 (u09wm)  (b) Precision 6 (u09wmd)  (c) Precision 7 (u09wmdw)

Fig. 2. Sector size depending on Geohash precision value

This is a public domain geocode system invented in 2008 by *Gustavo Niemeyer* which uses GPS coordinates to encode a geographic location into a short string of letters and digits [8]. It is a hierarchical spatial data structure which subdivides space into buckets of grid shape.

The sector area depends on the 'Geohash length' also known as *Precision*. The higher the Geohash precision value is, the smaller its corresponding sector area. Table I lists some cells dimensions depending on Geohash precision. To determine the most accurate precision value, different Geohash lengths from 5 to 7 characters were tested. A 6-character long Geohash has been selected because its precision fits the best understandable well-known places to study like *Notre-Dame de Paris* (N-d-P) or *Stade de France* (S-d-F) as depicted by Figs. 2a to 2c. A 6-character Geohash defines a cell with a height of 1.2km and a width equal to approximatively 0.6km.

### B. Activity Deviations Extraction

This phase consists, for each sector, in joining data points together to define clusters for some given time periods or slices based on the attributes computed during the pre-processing phase. If a period-activity differs from others, it is deemed suspicious and highlights an abnormal behaviour. This step

TABLE I
PRECISIONS & CORRESPONDING SECTOR DIMENSIONS

| Precision | Height | Width |
|---|---|---|
| 1 | 5,000 | 5,000 |
| 2 | 1,250 | 625 |
| 3 | 156 | 156 |
| 4 | 39 | 19.5 |
| 5 | 4.9 | 4.9 |
| 6 | 1.2 | 0.6 |
| 7 | 0.153 | 0.153 |
| 8 | 0.038 | 0.019 |

Distances in km

could be performed with the unsupervised *K-Means* clustering MLA. But this algorithm imposes a provision of 'expected number of clusters' (known as '*K*'). Problem is that we do not know in advance how the clusters will be constituted. Therefore, we selected the *X-Means* algorithm, an improved and better unsupervised MLA.

*X-Means* is a variation of *K-Means* clustering which further refines cluster assignments by repeatedly attempting subdivision, and keeping the best resulting splits [9]. The algorithm begins by randomly positioning centre points named *starter means*. It then associates with the same cluster observations closest to those *means*, calculates the average observations from each cluster and moves corresponding *means* to a computed position. It continues to reassign the observations to the nearest means and so on, to achieve consensus. To ensure the stability of located groups, the *starter means* selection is repeated several times to overcome some initial draws that may give a different configuration from the majority of cases [10].

### C. Outliers Detection

The third phase performs the *Outliers detection*. It aims to detect network anomalies by pointing out the *outliers*. It is performed by the *Local Outlier Factors* algorithm [11]. LOF is based on the concept of local density, where locality is given by $k$ nearest neighbours, and whose distance like *Euclidean*, *Mahalanobis* or *Chebyshev* is used for estimating the density. By comparing the local density of an object to the local densities of its neighbours, it is possible to identify regions of similar density, and points that have a substantially lower density than their neighbours. These are considered as *outliers* and so highlight anomalies.

## III. EMPLOYED DATASET

For this study, we have employed a dataset that has been generated to serve several research objectives [?]: *(i)* collecting new measurements that describe mobile network data traffic for different mobile services separately. Data are gathered in an operational nationwide network by combining the output of multiple monitoring technologies *(ii)* evaluating existing analytics for classification, prediction and anomaly detection within real-world high-detail per-service mobile network data that represents several terabytes of traffic per hour *(iii)* demonstrating the integration of data analytics within next-generation network architectures in practical case studies.

This dataset consists of data collected from a major French mobile operator at large scale. It is made of novel *anonymized high-detail flows* of contextualized user mobility and mobile service usage traffic data records (xDR). Network flows have been obtained by leveraging passive probes deployed in the French operator mobile network. These data sources have different levels of spatio-temporal accuracy.

It characterizes the mobile traffic and application usage over a 3-month period for the entire national territory of France. Data has been erased according to GDPR rules and the national telecom code of conduct but anonymized aggregates are available for analyses. This paper focuses on the Ile-de-France area to detect anomalies in a mobile network based on recurring behaviour analysis at an application level. The dataset employed in this study is composed of 26 '.csv' files describing each radio access site sorted in alphabetical order. The total file size represents *184Go* of raw data. Provided network flows provided in each file are composed of 11 fields:

**PortApp** Mobile application type, **LocInfo** Antenna's identifier, **Coord_X/Y** Site coordinates, **SiteName** Site's name, **TimeSlot** Date and time in 30mn slots, **Duration** TCP session duration, **Users** Number of distinct users, **nPktUp & nPktDn** Total number of packets, **Flows** Number of traffic flows exchanged during TCP session.

## IV. EXPERIMENTS & RESULTS

For this study, we chose to work on a 3-month period of data extracted from 16/03/2019 to 14/06/2019 divided into 30-minute time slots. A snippet describing few lines of raw data is presented by Fig. 3. Some values have been obfuscated to avoid privacy issues.

```
PortApp,LocInfo,COORD_X,COORD_Y,SiteName,TimeSlot,nPktUp,nPktDn,Duration,Users,flows
65677,XXXXX01,AAAA75,BBBB83,U_ARENA,2019-05-13 14:00:00,AAA,XXX,YYYY.ZZZZ,1,1
65648,XXXXX02,AAAA75,BBBB83,U_ARENA,2019-05-13 10:00:00,AAA,XXX,YYYY.ZZZZ,2,9
65807,XXXXX04,AAAA10,BBBB40,UNIV_PARIS,2019-03-18 19:30:00,AAA,XXX,YYY.ZZZZZ,3,4
```

Fig. 3. Raw data snippet

### A. Data Aggregations

*By-site* and *by-sector* aggregation processes aim to reduce the amount of raw data and to extract the underlying structure for a better understanding and further analyses. Employed `Perl` parsers and tools can be freely downloaded from our repository [12].

*1) By-site aggregation:* during this process, raw traffic flows are aggregated by BTS to split data into sites. As described in Section III, 11 fields are available. To perform this initial aggregation step, we employed a *3-tuple aggregation key* built as follows 'TimeSlot; SiteName; AppGroup'. 'PortApp' defines the application types and different 'PortApp' can correspond to the same 'AppGroup'. So, we have had to transpose 'PortApp' into 'AppGroup' in first place. For this purpose, the dictionary detailed below has been generated from the correspondence table provided by the mobile operator and used for merging flows related to the same 'AppGroup':
1) Unknown 2) Web 3) P2P 4) Download 5) CloudStorage 6) Mail 7) DB 8) Others 9) Control 10) Games 11) Streaming

12) Chat 13) VoIP 14) MailOperator 15) VPN 16) VVM 17) MMS 18) StreamAVSP 19) Portal.

Obtained data after this aggregation phase are composed of 9 fields: 'TimeSlot, SiteName, Coord_X, Coord_Y, AppGroup, Packets, Duration, Users, Flows'. This first step allowed us to reduce the total amount of raw data from 184 GB to 12 GB that represents a reduction factor near 16 as detailed in Table II with a resulting number of extracted BTS equal to 2,736. You can notice that no BTS with a name starting with the 'X' letter exists.

*2) By-sector aggregation:* this second aggregation consists in grouping BTS into sectors. For each flow, the corresponding Geohash is computed by using the 'Coord_X/Y' values and the `Perl` Geohash package [13] with a precision value equal to 6. Then, the merging process is based on the *3-tuple primary key* 'TimeSlot; Geohash; AppGroup'. As depicted by Fig. 4, structure of the obtained aggregated data corresponds to this scheme: 'TimeSlot, Geohash, AppGroup, Packets, Duration, Users, Flows'. The last 4 fields are employed to characterize network activity for each sector and time slice. Table III lists the number of obtained sectors depending on the Geohash precision value.

### B. Activity Deviations Extraction

To test and validate our proposed anomalies detection method, we focused on April from Monday 1st to Sunday 28th and some days in May, 2019. Parsed data relative to these dates have been split by weeks, days or time slices using the bash 'grep' command to filter desired timestamps.

Activity deviations extraction is performed with the *X-Means* algorithm. This unsupervised MLA is used for clustering parsed data for each sector for specific time slices based on the 4 fields describing the network activity previously computed.

TABLE II
RAW VS AGGREGATED DATA

| File | Raw | Aggregated | File | Raw | Aggregated |
|------|-----|-----------|------|-----|-----------|
| df_A | 9.7G | 645M | df_M | 13G | 806M |
| df_B | 14G | 811M | df_N | 6.6G | 377M |
| df_C | 23G | 1.4G | df_O | 5.9G | 572M |
| df_D | 3.2G | 205M | df_P | 19G | 1.2G |
| df_E | 3.6G | 217M | df_Q | 1.9G | 87M |
| df_F | 5.3G | 295M | df_R | 13G | 739M |
| df_G | 6.7G | 431M | df_S | 18G | 1.1G |
| df_H | 4.0G | 220M | df_T | 3.8G | 346M |
| df_I | 3.0G | 154M | df_U | 681M | 38M |
| df_J | 1.5G | 88M | df_V | 13G | 814M |
| df_K | 881M | 53M | df_W | 211M | 8.7M |
| df_L | 18G | 1.1G | df_Y/Z | 70M/316K | 5.4M/128K |

By-site aggregation details – Ratio ≈ 16

```
TimeSlot,Geohash,AppGroup,Packets,Duration,Users,Flows
"2019-03-01 17:30:00",u09ty5,2,72511,1972528.375,1,1
"2019-03-02 19:00:00",u09tvm,7,3157,1182862.625,1,1
"2019-03-05 08:30:00",u09wjd,1,16582,1297180.375,1,1
```

Fig. 4. By-sector aggregated data snippet

TABLE III
GEOHASH PRECISION VS NUMBER OF SECTORS

| Geohash precision | 5 | 6 | 7 | 8 |
|---|---|---|---|---|
| Sectors | 109 | 1,233 | 2,545 | 2,707 |
| File size (MB) | 16 | 139 | 246 | 258 |

Number of sectors and file size increase with precision value



(a) Gare Lyon – 2nd week

(b) Gare Lyon – 3rd week

(c) N-d-P – 1st week

(d) N-d-P – 2nd week

(e) N-d-P – 3rd week

(f) N-d-P – 4th week

Fig. 5. Activity deviations by week



(a) Stade de France – 04/20
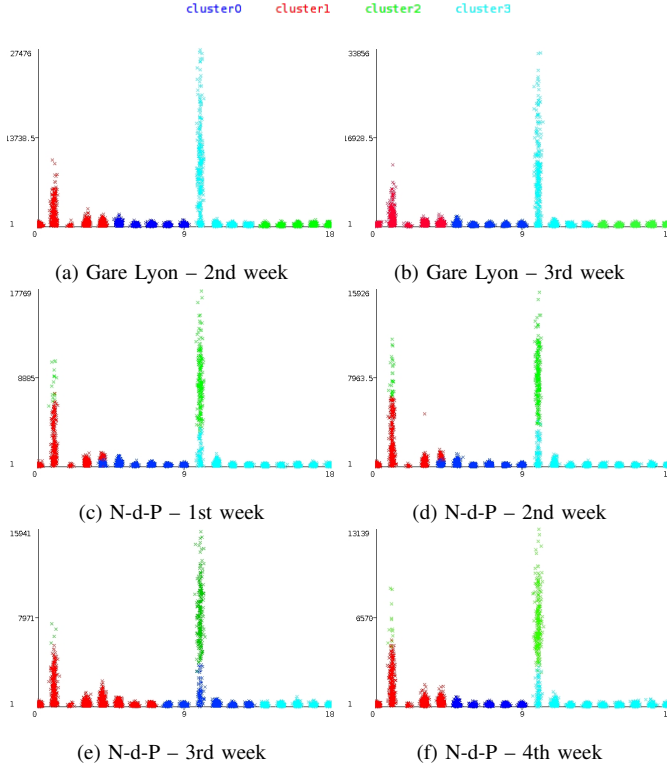
(b) Stade de France – 04/27

(c) Stade de France – 05/04

Fig. 6. Activity deviations by day

It has been implemented using Weka [14]. We employed the *Waikato Environment for Knowledge Analysis Toolbox* [15] to conduct our experiments. It is a freely available open source software developed at the University of Waikato in New Zealand providing a set of visualization tools, different machine learning algorithms for data analysis and extensions, together with graphical user interfaces for ease of access to these functions. Overall, data must be transformed into an ASCII text file that describes a list of instances sharing a set of attributes to be analysed with Weka. This step is done by the parsers during the aggregation processes.

An activity analysis is performed for each time period and for each sector [*sector's name (Geohash)*]:

*1) By-week time periods:* Figure 5 is a representation *by week* of cluster assignments with number of users based on the application groups for *Notre-Dame de Paris (u09tvm)* and *Gare de Lyon (u09ty5)*.

From Figs. 5a and 5b, we can deduce that activity of *Gare de Lyon* is similar each week. This is also the case for the first and fourth week of April, 2019. We can conclude that no anomaly exists and therefore no particular event occurred.
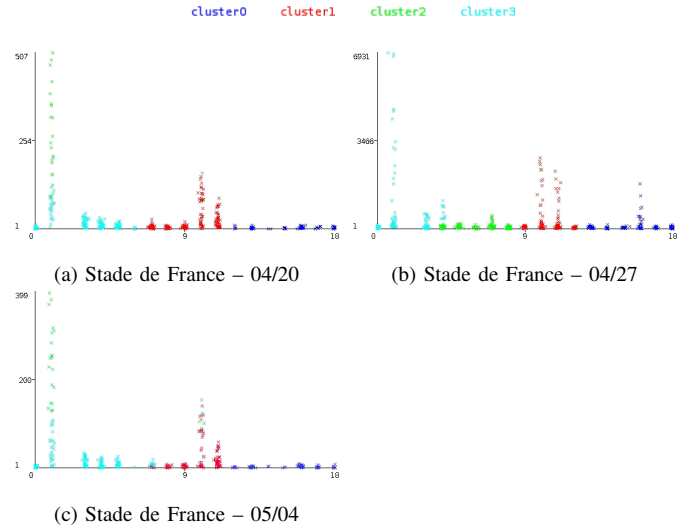
Figures 5c to 5f represent *Notre-Dame de Paris*, we can observe that activity of weeks 1, 2 and 4 is uniform. An exception can be found in Fig. 5f exposing fourth week clusters where an insignificant evolution appears regarding the 'AppGroup' 4 clusters distribution.

However, by focusing on Fig. 5e, it is obvious that the network activity is very different from the other ones. The 'AppGroups' 5 (Mail), 6 (DB), 7 (Others), 10 (Streaming), 11 (Chat), 12 (VoIP) and 13 (MailOperator) are not assigned to the same clusters.

*2) By-day time periods:* Figure 6 is a representation *by day* of cluster assignments with number of users based on the application groups for *Stade de France (u09wmd)*. We selected the 20th, the 27th of April and the 4th of May, 2019 which are 3 Saturdays.

By focusing on Fig. 6b, it is obvious that clusters distribution is completely different from the others. We can deduce that network activity has deviated compared to the other days. So, the network activity has deviated.

*3) By-hours time periods:* Figure 7 is a representation *by hours* of cluster assignments with number of users based on the application groups for *Stade de France (u09wmd)*. We selected the 5th, the 12th and the 19th of May, 2019 which are 3 Sundays and then we extracted a 4-hour time slice from 8pm to 11pm for each Sunday.

Like by-day analysis, if we focus on Fig. 7b, we can conclude that network activity is very different from the other time slices. So, this particular day, the network activity has changed for this specific time period.

This activity deviations extraction highlights some network behaviour changes that required further analysis. It is performed during the next step by means of an unsupervised classifier.
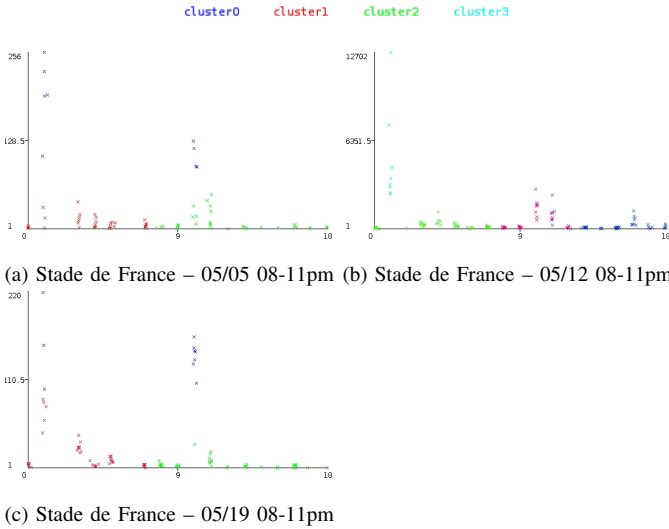
(a) Stade de France – 05/05 08-11pm (b) Stade de France – 05/12 08-11pm



(c) Stade de France – 05/19 08-11pm

Fig. 7. Activity deviations by hours

## C. Outliers Detection

The LOF algorithm allowed us to detect 15 outliers in 5,075 instances (0.3%) from sector *Notre-Dame de Paris (u09tvm)* for the third week of April. About *Stade de France (u09wmd)* sector, 18 outliers in 580 instances (3.1%) for April 27, 2019 and 1 outlier in 132 instances (0.8%) for May 12, 2019 from 8pm to 11pm have been extracted.

From Table IV, where some of the detected outliers are summarized, we can deduce that an anomaly concerning sector *Notre-Dame de Paris* occurred from April 15th, 2019 at 6pm to April 17th, 2019 at 10am. This event can be correlated with actualities and is relative to the *Notre-Dame de Paris* fire. The other anomalies correspond to the *French soccer cup final match* on 27th of April at 8pm or the *Metallica concert* on 12th of May, 2019 at 9pm that were held in *Stade de France*.

## V. CONCLUSION & FURTHER WORKS

From this study, we can conclude that our proposed process based on an *Unsupervised Activity Deviations Extraction* is able to detect network anomalies. In this use case, anomalies

are defined by a behaviour change in network activity. We selected this kind of behaviour because application activity was a feature available in our dataset but it can be transposed over other networks with different parameters.

Our approach consists in the first place to *aggregate data into sectors* that represent specific areas. Then, an unsupervised MLA is employed to extract *network behaviours* from these sectors. Based on those features, we characterize some *activity deviations* for each sector and for particular time periods. After that, we extracted outliers to detect any network anomalies. This last step helped us to highlight particular events.

Furthermore, we can classify sectors depending on the type of activity by selecting more sectors to compare. We plan to apply our concept on smaller sectors defined by 7-character Geohash values, to detect less significant events or a lack of particular network traffic where a usual high activity level is the norm or required. We also want to detect network security anomalies by using an hybrid approach [16].

## TABLE IV
OUTLIERS EXTRACTED BY THE LOF ALGORITHM

| Instance | TimeSlot | AppGroup | Packets | Geohash | Flows |
|---|---|---|---|---|---|
| 528 | 2019-04-15 18:00:00 | 12 | 1745 | u09tvm | 9 |
| 714 | 2019-04-15 23:30:00 | 12 | 360 | u09tvm | 2 |
| 808 | 2019-04-16 02:30:00 | 4 | 8347 | u09tvm | 68 |
| 926 | 2019-04-16 07:00:00 | 4 | 9425 | u09tvm | 175 |
| 1012 | 2019-04-16 10:00:00 | 1 | 29726 | u09tvm | 235 |
| 1355 | 2019-04-16 20:00:00 | 10 | 731148 | u09tvm | 9194 |
| 1767 | 2019-04-17 10:00:00 | 0 | 4372 | u09tvm | 15 |
| 379 | 2019-04-27 18:00:00 | 15 | 2354 | u09wmd | 6 |
| 443 | 2019-04-27 20:00:00 | 10 | 8256608 | u09wmd | 8550 |
| 559 | 2019-04-27 23:00:00 | 5 | 43482 | u09wmd | 135 |
| 74 | 2019-05-12 22:00:00 | 16 | 167652 | u09wmd | 605 |

Events detection

## REFERENCES

[1] J. Russell, "The world's largest DDoS attack took GitHub offline for fewer than 10 minutes," Mar. 2018. [Online]. Available: https://social.techcrunch.com/2018/03/02/the-worlds-largest-ddos-attack-took-github-offline-for-less-than-tens-mn

[2] E. S. C. Vilaça, T. P. B. Vieira, R. T. de Sousa, and J. P. C. L. da Costa, "Botnet traffic detection using RPCA and Mahalanobis Distance," in *WCNPS*. Brasilia, Brazil: IEEE, Oct. 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8896228/

[3] S. B. Wankhede, "Anomaly Detection using Machine Learning Techniques," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, Mar. 2019, pp. 1–3.

[4] A. Guezzaz, Y. Asimi, M. Azrour, and A. Asimi, "Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 18–24, Mar. 2021.

[5] W. Wang, Z. Wang, Z. Zhuo, H. Deng, W. Zhao, and C. Wang, "Anomaly Detection of Industrial Control Systems Based on Transfer Learning," *Tsinghua Science and Technology*, vol. 26, no. 6, Oct. 2021.

[6] N. Ghosh, K. Maity, R. Paul, and S. Maity, "Outlier Detection in Sensor Data Using Machine Learning Techniques for IoT Framework and Wireless Sensor Networks," in *2019 International Conference on Applied Machine Learning (ICAML)*, May 2019, pp. 187–190.

[7] "Geohash Intro — Big Fast Blog," Jan. 2012. [Online]. Available: https://web.archive.org/web/20120112004608

[8] W. Hill, "Geohashing," Apr. 2017. [Online]. Available: https://medium.com/@bkawk/geohashing-20b282fc9655

[9] D. Pelleg and A. Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," *Machine Learning*, Jan. 2002.

[10] S. Ghosh and S. Kumar, "Comparative Analysis of K-Means and Fuzzy C-Means Algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 4, May 2013.

[11] "Detect Outlier (LOF) - RapidMiner Documentation." [Online]. Available: https://docs.rapidminer.com/latest/studio/operators/cleansing/outliers/detect_outlier_lof.html

[12] "Cmaudoux / digital-signatures-extraction." [Online]. Available: https://bitbucket.org/cmaudoux/digital-signatures-extraction/src/master/

[13] "Geohash - Great all in one Geohash library - metacpan.org." [Online]. Available: https://metacpan.org/pod/Geohash

[14] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools*, ser. MK Series in Data Management Systems. San Diego, CA, USA: Morgan Kaufmann, Oct. 2016.

[15] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/

[16] C. Maudoux, S. Boumerdassi, A. Barcello, and E. Renault, "Combined Forest: A New Supervised Approach for a Machine-Learning-based Botnets Detection," in *2021 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2021, pp. 01–06.