Proceedings of Machine Learning Research 128:1–13, 2020 Conformal and Probabilistic Prediction and Applications

A Conformalized Density-based Clustering Analysis of Malicious Traffic for Botnet Detection

Bahareh Mohammadi Kiani BAHAREH.MOHAMMADI@SHOMAL.AC.IR
Department of Information Technology
Shomal University
Amol, IRAN

Abstract

In this work, we present a clustering technique within the conformal prediction framework and describe its application to bot-generated network traffic in order to build botnet behavioral models, with a view to improving the detection of compromised hosts. The technique has a natural connection to density-based clustering. Once a required significance level has been set, this technique can discover the clusters and the noise in the data. To obtain a clustering of the underlying distribution, we use conformal prediction in combination with a density estimator which is used for point prediction, to identify a few so-called focal points that are indeed the centers of possibly overlapping spheres or ellipsoids, that represent the clusters. There are several advantages to the developed technique: the number of clusters is determined automatically. Furthermore, the technique is able to find non-linearly separable clusters. Moreover, a new conformity measure related to BotFinder, an algorithm for finding bots in network traffic, is developed that can be used as a method for point prediction. We performed an experimental evaluation of the proposed approach in terms of efficiency and accuracy. The results suggest that the approach obtains relatively high accuracies and is more effective when compared with previous conformal clustering techniques.

Keywords: Information security, Botnet, Conformal prediction, Clustering, Density

1. Introduction

A botnet is a malicious network of bots coordinated by a single entity called the botmaster. A bot is a malware performing illegal activities based on commands received from botmaster over a command and control channel. Botnets cause a wide range of threats on the Internet of which the most typical ones are distributed denial of service attack, click fraud, data theft, illegal resource consumption, and spamming. Thereby, rapid detection of botnets is essential to confront security problems rooted in these networks (Antonakakis et al., 2017).

There are many different methods proposed for detecting bots, which are divided into two main categories. The first category is host-based methods used to spot bot-infected hosts by investigating changes occurred in an individual host such as creating new files, opening ports, or changing computer registries. The second category is network-based methods considered as a complementary solution to the host-based detection approaches. In this category, bot-infected hosts are detected by similarity analysis of network traffic by comparing traffic against previously-extracted behavioral models from malicious con-

nections. Moreover, network-based methods include two main methods, namely domain name system (DNS) traffic analysis and the main traffic analysis including non-DNS traffic. These methods are all based on the following observations which define the core network behavior of bots: (a) Bots of a certain family follow a similar network behavior pattern in their communication with command and control server. (b) The relationship between the botmaster and bots has a one-to-many pattern (Khattak et al., 2013; Amini et al., 2015).

Machine learning, as the main component of modern artificial intelligence (AI), has shown great potential in analyzing security data, quickly identifying attack mechanisms, and classifying threats. In this context, various methods have been developed for botnet analysis and detection, which rely mostly on clustering analysis of main network traffic to identify bots' group behavior recurring methodically over their lifetime. Some of the approaches in this direction are (Tegeler et al., 2012; Dietrich et al., 2013; Cherubin et al., 2015) in which by capturing hosts network-level activities, extracting related statistical characteristics, and using a clustering technique, a model of the network communication is developed in the normal or attack conditions and used in intrusion detection systems for identifying botnets. The work in (Cherubin et al., 2015) applied conformal prediction (Vovk et al., 2005) for the first time in clustering for botnet detection to produce clusters as the models of core network-level behavior of some certain bot families. It developed a conformalized clustering technique, has a natural connection to hierarchical clustering where the percentage of objects that are left outside any clusters is regulated by setting up a required level of confidence. It extended the work in (Smith et al., 2014) which was focused on a binary unsupervised problem in the anomaly detection context to a multi-class unsupervised learning task of clustering. The work in (Cherubin et al., 2015) has several drawbacks including inefficiency of the proposed algorithm and producing many small clusters leading to poor clustering quality. Besides, it used *purity* as their main evaluation criterion which is found to be inefficient in evaluating the clustering quality, as this criterion does not take into account the number of created clusters and high purity is easy to obtain when the number of produced clusters is large - in particular, maximum purity is achieved if each object gets its own cluster (Manning et al., 2008).

The aim of this paper is to improve the conformal clustering and anomaly detection technique developed in (Cherubin et al., 2015) in terms of efficiency and accuracy. This paper makes the following contributions:

- Creating a density-based clustering framework where conformal prediction is used in combination with a nonparametric density estimator, to identify a few so-called focal points that are indeed the centers of possibly overlapping balls or ellipsoids, that represent the clusters.

- Utilizing an entropy-based conformity measure, adopting the cluster quality rating function in the BotFinder (Tegeler et al., 2012) in conformal prediction.

The rest of this paper is organized as follows. Section 2 deals with the basic concepts, introduces the conformal prediction, and describes the drawbacks of the previous works. In Section 3, we present our clustering approach. The evaluation results using the pre-processed bot-generated traffic dataset is presented and discussed in Section 4. Finally, section 5 provides our conclusions and describes some future works.

2. Conformal Clustering

Conformal clustering is based on the conformal prediction technique. In this section, we first introduce the conformal prediction, and subsequently, we describe the previously proposed conformal clustering approach and discuss its drawbacks in some detail.

2.1. Conformal Prediction

Let $Z_1, \ldots, Z_n$ be a random sample from distribution P and let $\mathrm{Z} = (Z_1, \ldots, Z_n)$. For any given $z \in R^d$, we provisionally set $Z_{n+1} = z$. Let $a_i = A(\{Z_1, \ldots, Z_{n+1}\}, Z_i)$ be a conformity measure that represents numerically how much Z_i conforms to $\{Z_1, \ldots, Z_{n+1}\}$. The only requirement is that Z be exchangeable. Conformal prediction is an approach where a conformity measure is used to predict, with a level of confidence $(1-\alpha)$, whether a new object Z_{n+1} conforms to the underlying distribution of a previously observed random sequence Z. To test the conformity of the new object Z_{n+1}, the hypothesis $H_0 : Z_{n+1} = z$ is tested by computing the p-value

$$\pi(z) = \frac{1}{n+1} \sum_{i=1}^{n+1} 1[a_i \leq a_{n+1}].$$

As the data is exchangeable, under H_0, $\pi(z)$ is uniformly distributed among $\{1/(n+1), ..., 1\}$ and we have $\pi(z) \geq \widetilde{\alpha}$ where $\widetilde{\alpha} = \lfloor (n+1)\alpha \rfloor / (n+1) \approx \alpha$, Z_{n+1} takes the value z and a prediction set is constructed: $\hat{C}^{(\alpha)}(Z_1, \ldots, Z_n) = \{z : \pi(z) \geq \alpha\}$, such that $P^{n+1}(Z_{n+1} \in \hat{C}^{(\alpha)}) \geq 1 - \alpha$ for any distribution P (Lei et al., 2013).
The algorithm of the conformal prediction can be shown as follows:

Algorithm 1: Conformal Prediction using a new object

Data: Bag of objects $Z = \{Z_1, \ldots, Z_n\}$, conformity measure A, significance level α, a new object z
Result: *True*, if z is conform to Z
Set provisionally $Z_{n+1} = z$ and $Z = \{Z_1, \ldots, Z_{n+1}\}$;
for $i \leftarrow 1$ **to** $n+1$ **do**
  $a_i \leftarrow A(Z, Z_i)$;
end
$\tau = U(0, 1)$;
$\pi(z) = \frac{\tau}{n+1} \sum_{i=1}^{n+1} 1[a_i \leq a_{n+1}]$;
if $\pi(z) \geq \alpha$ **then**
  Output True;
else
  Output False;
end

In this algorithm, the parameter τ is sampled uniformly at random between zero and one by which a smooth conformal predictor is obtained, as suggested by (Gammerman and Vovk, 2007).

2.2. Conformal Clustering

Conformal prediction is usually applied to classification, more specifically to the problem of anomaly detection. Recently, there have been efforts to apply conformal prediction to the unsupervised learning task of clustering (Nouretdinov et al., 2019; Shin et al., 2019; Cherubin et al., 2015). The main idea in the work (Shin et al., 2019) is to combine conformal prediction with some traditional clustering approach such as k-means or density-based clustering. They applied this technique to solve several problems in clustering such as how to tune model parameters, how to merge clusters, and how to accommodate clusters of more general shapes and sizes. However, the work in (Cherubin et al., 2015) proposed a clustering technique solely based on conformal prediction, comparable to hierarchical clustering, which allows regulating the number of instances left outside of any clusters by setting up a required confidence level. They applied the proposed clustering technique to a botnet detection problem. This work used conformal prediction to construct a sequential prediction set of possible objects on a grid which conform to the dataset and thus are not anomalies, by using a chosen conformity measure. Then, the resulting prediction set was interpreted as a hierarchy of clusters where a clustering was given by regulating the depth of the hierarchy with a suitably chosen significance level. The conformal clustering approach described in (Cherubin et al., 2015) has several drawbacks. The first issue is, in fact, not considering clusters as high-density regions of the underlying distribution while developing a multi-class unsupervised learning task of clustering. Considering the true number of easily separable classes in the dataset, the second main issue has to do with managing an acceptable number of clusters to be produced. Due to architectural issues in the algorithm, the number of clusters is overestimated and roughly, each object gets its own cluster indicating a poor clustering quality. Recently, the work in (Nouretdinov et al., 2019) extended the conformal clustering technique developed in (Cherubin et al., 2015) to run at a range of significance levels to allow for fine regulating of the depth of the clusters' hierarchy. However, this paper did not address the main objectives of the original paper.

3. The Proposed Method

Our approach includes following main stages:

1. Dimensionality reduction using the t-SNE algorithm
2. Splitting data into train and detection set
3. Applying density-based conformal clustering on training data
4. Assigning detection set objects to the created clusters

Accuracy of clustering plays an important role in lowering the rate of false positive and false negative when applied to botnet detection. The numerical data presented in the next section compares the quality of detection in the proposed scheme to the previous schemes.

3.1. Dimensionality Reduction

We use t-SNE (Maaten and Hinton, 2008) to embed data in two-dimensional space. In general, the data in the feature space is multi-dimensional. This makes clustering difficult

due to the possibility of putting each data in a separate cluster. The t-SNE algorithm is an effective technique that preserves the small pairwise distances when mapped to a two-dimensional space. This makes it more reliable compared to similar algorithms such as Principal Component Analysis (PCA) (Ng, 2017), which preserves large pairwise distances.

3.2. The Proposed Conformalized Density-based Clustering Technique

In our approach, we use the reliability and validity of conformal prediction in combination with the quality of the density-based clustering to arrive in a highly effective clustering scheme. In this paper, we focus on clusters that are defined as the connected components of high-density regions of the underlying distribution. Given a random sample from a distribution P and an appropriate density estimator $\hat{\mathrm{p}}$ which is used as a conformity measure, we apply conformal prediction to construct a prediction set consisting of a few so-called focal points that are the centers of possibly overlapping balls or ellipsoids. Let $L = \{z \in R^d : \hat{p}(z) > t\}$ be the density level set for a given density estimator $\hat{\mathrm{p}}$. We use this set for clustering. In fact, conformal prediction is used to turn this set into a union of balls or ellipsoids through a grid structure by which a clustering of the underlying distribution is given. It is worth noting that L is hard to compute and represent. Thereby, this set is approximated with a union of spheres.
Let the training data be a random sequence $\mathrm{Z} = (Z_1, \ldots, Z_n)$ sampled from a distribution P. In our implementation, we partition the training data space into a l^d grid structure, where l is a given number of points per side of the grid and d is the number of features. It is important to note that in partitioning, the number of cells (grid points) is considered much smaller than the number of data points. Each grid point z is then considered as a new object $Z_{n+1} = z$ centered at the smallest sphere that contains an effective number of neighbors of this point, where the number of neighbors is controlled by scaling the tuning parameter in the chosen density estimator. By using the density estimator as a conformity measure, a density from the augmented dataset $\{Z_1, \ldots, Z_{n+1}\}$, $A_i = \hat{p}(Z_i)$, is thus estimated. Then, by applying the conformal prediction, p-value is then computed

$$\pi(z) = \frac{1}{n+1} \sum_{i=1}^{n+1} 1[\hat{p}(Z_i) \leq \hat{p}(z)].$$

This process is repeated for each point on the grid and eventually, conform points to the underlying distribution are obtained with respect to a given significance level α and the prediction set $\hat{C}^\alpha$, union of balls with $(1-\alpha)$ coverage, is thus constructed

$$\hat{C}^\alpha = \{z : \pi(z) \geq \alpha\}.$$

In this context, α can be used to trim the outliers or isolated points that cannot form the cluster, while the remaining ones are possibly density-connected points that will be in the same cluster. The conform points are called focal points. Eventually, clusters are obtained by grouping these so-called focal points in such a way that two points which are neighbors on the grid are assigned to the same cluster. The proposed algorithm is described

in Algorithm 2:

Algorithm 2: The conformal clustering algorithm

1. Split the dataset randomly into training and detection set Z_{train}, $Z_{detection}$, where $Z_{train} = \{Z_1, ..., Z_n\}$
2. Partition the training data space into an adaptive grid structure, where # of cells $\ll$ # of data points, and let's consider it as a set of candidate focal points F.
3. Fix $Z_{n+1} = f$ where $f \in F$. Define the augmented dataset $D = \{Z_1, ..., Z_{n+1}\}$.
4. Estimate a density $\hat{p}$ from D.
5. Let
$$\pi(f) = \frac{1}{n+1} \sum_{i=1}^{n+1} 1[\hat{p}(Z_i) \leq \hat{p}(f)].$$
6. Repeat the above steps for each $f \in F$ and set
$$\hat{C}^{(\alpha)} = \{f : \pi(f) \geq \alpha \}.$$
7. Group the elements of $\hat{C}^{(\alpha)}$: union of balls with $(1-\alpha)$ coverage, such that two points f_k, f_j which are neighbors on the grid, are assigned to the same cluster.

There are a number of hyperparameters in our method that we have to find a good setting for them, including the number of points per grid side l, α, the hyperparameter for the kernel density estimator which is bandwidth h, the number of neighbors k, if using k-nearest neighbors density estimator, or σ in the entropy-based conformity measure. These hyperparameters are of importance because they can correctly identify the true underlying clustering structure. We can systematically organize these hyperparameters tuning process to converge on the best clustering result, depending on a measure of the strength of clusters such as index functions.

Once obtained the clusters, we assign detection set objects to them such that for each object Z_i in the detection set $Z_{detection}$, we compute its distance to the closest focal point $f_{j(i)}$, which is called the residual

$$R_i = \left\| Z_i - f_{j(i)} \right\|.$$

Let t_α be the $1-\alpha$ quantile of the residuals. We associate $Z_i \in Z_{detection}$ with the same cluster as its closest focal point $f_{j(i)}$, where $R_i = \left\| Z_i - f_{j(i)} \right\| \leq t_\alpha$.

3.3. Conformity Measures

We use three density-based conformity measures $A(\{Z_1, \ldots, Z_{n+1}\}, Z_i)$ including *k-Nearest Neighbors (k-NN) Density Estimator*, *Gaussian Kernel Density Estimator (KDE)*, and an *Entropy-based Density Estimator* which was adapted from the method in (Maaten and Hinton, 2008), respectively.

3.3.1. K-nearest Neighbors Density Estimator

Given $R_k(Z_i)$ to be the distance from Z_i to its k^{th} nearest-neighbor, A_i is:

$$\begin{aligned} A_i &= \frac{k}{n} \times \frac{1}{V_d \times R_k^d(Z_i)} \\ &= \frac{k}{n} \times \frac{1}{\text{Volume of a } d\text{-dimensional ball with radius being } R_k(Z_i)}, \end{aligned} \tag{1}$$

where $V_d = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}$ is the volume of a unit d-dimensional ball and $\Gamma(Z_i)$ is the Gamma function.

3.3.2. Gaussian Kernel Density Estimator

Given K to be the Gaussian kernel function, A_i is:

$$A_i = \frac{1}{(n+1)h^d} \sum_{j=1}^{n+1} K\big(\frac{Z_i - Z_j}{h}\big), \tag{2}$$

where d is the number of features and h is the kernel bandwidth.

3.3.3. Entropy-based Density Estimator

A_i, the entropy-based density estimator for object Z_i, is defined as:

$$\begin{aligned} A_i &= \frac{1}{2^{H(Z_i)}}, \\ H(Z_i) &= \sum_{j=1}^{n+1} p_{Z_i|Z_j} \log \frac{1}{p_{Z_i|Z_j}}, \\ p_{Z_i|Z_j} &= \frac{\exp(-||Z_i - Z_j||^2/2\sigma^2)}{\sum_{k \neq i} \exp(-||Z_i - Z_k||^2/2\sigma^2)}, \qquad if j \neq i \end{aligned} \tag{3}$$

where $H(Z_i)$ is the Shannon entropy of Z_i. $p_{Z_i|Z_j}$ measures the similarity between point Z_i and nearby points by centering a Gaussian kernel on Z_i and computing the density of all other points, in this case, Z_j under the Gaussian. σ is the variance of the Gaussian. If two points are close together then $p_{Z_i|Z_j}$ is a large value, and the resulting entropy becomes extremely small that is indicative of highly small pairwise distances. In our density-based clustering approach, this exponentially decreasing measure is of great importance because it simultaneously controls the number of neighbors (size) and small pairwise distances (compression). Moreover, the logic of entropy-based conformity measure is based on the cluster quality rating function in BotFinder. In BotFinder, each cluster quality is scored based on an exponentially decreasing function as follows:

$$q_{cluster} = \exp(-\beta \frac{sd}{c}) \tag{4}$$

with controlling factor β which is empirically set to 2.5. sd is the standard deviation of the cluster centered at c.

A cluster with larger size and highly similar values gains a higher score than the smaller and sparser cluster. Hence, in this respect, there is consistency between the entropy-based conformity measure and the cluster quality rating function in BotFinder.

4. Results

4.1. Data Overview

We use the same pre-processed dataset used in the original paper. To create the corresponding dataset, firstly, they captured traffic traces produced by 9 certain botnet families in a certain amount of time. Then, they extracted feature vector from every network trace. The created dataset is composed of 18 features or netflows characteristics including median and MAD (mean of absolute deviations) of netflows duration, communication frequency, median and MAD of exchanged bytes, percentage of using TCP and UDP protocols, percentage of using well-known, registered and dynamic ports, median and MAD of the number of received and sent bytes, median and MAD of the number of received and sent bytes by considering the bot as connection initiator. This dataset contains traffic from 4 classes of botnets, three of which represent three different botnet structures namely HTTP-based, P2P-based, and IRC-based, while the remaining fourth class represents Mebroot. Since the Mebroot traffic which is an HTTP-based bot was easily separable from the other classes, thus, it was considered as a separate class (Cherubin, 2014).

4.2. Evaluation Metrics

In the following, we evaluate created clusters as the detection models generated during our clustering analysis of botnet malicious connections, both in terms of homogeneity and detection capability. To this end, we use some evaluation measures including Normalized mutual information (NMI), Rand index, F-measure, Recall, and Precision (Manning et al., 2008). It is worth noting that we do not use the data label at the clustering stage, since clustering is an unsupervised task. However, we do use it for evaluation of the clustering.

NMI is a measure of cluster homogeneity which is defined as

$$\mathrm{NMI}(\Omega, \mathbb{C}) = \frac{I(\Omega; \mathbb{C})}{[H(\Omega) + H(\mathbb{C})]/2}, \tag{5}$$

where $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters, $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$ is the set of classes and

$$I(\Omega; \mathbb{C}) = \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log \frac{N|\omega_k \cap c_j|}{|\omega_k||c_j|},$$

$$H(\Omega) = -\sum_k \frac{|\omega_k|}{N} \log \frac{|\omega_k|}{N}.$$

$I(\Omega;\mathbb{C})$ measures the amount of information is given by clusters about what the classes might be, and H is entropy. It is worth noting that NMI provides a trade-off between the quality of the clustering and the number of clusters, and its value is always a number between 0 and 1. When the NMI is zero, a random clustering is reached with respect to class membership. If all data points are correctly assigned, then the NMI is one. Furthermore, to evaluate the detection capability of the corresponding models, we use the following measures:

$$\text{Rand index} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}, \tag{6}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8}$$

$$\text{F-measure} = 2\frac{Precision \times Recall}{Precision + Recall}, \tag{9}$$

where a true positive (TP) decision associates two similar data points to the same cluster. A true negative (TN) decision associates two dissimilar data points to different clusters. A false positive (FP) decision associates two dissimilar data points to the same cluster. A false negative (FN) decision associates two similar data points to different clusters.

4.3. Evaluating Multi-level Conformal Clustering and Anomaly Detection Technique

We first evaluate the proposed conformal clustering approach in the original paper using the aforementioned criteria and demonstrate our method improves upon the original conformal clustering approach. Similarly to the original paper, we reduce the data dimension to two using t-SNE algorithm. Then, we randomly choose 100 sample points out of 134 to build the training set, while the remaining 34 sample points constitute the detection set. To address a multi-level unsupervised learning task of clustering, naturally connected to hierarchical clustering, as previously mentioned, it used conformal prediction in clustering where the percentage of objects left outside of any clusters is controlled by setting up a required confidence level. To this end, it partitioned the data space into a large-sized grid structure (d-dimensional grid of l equally spaced points per side), where the training object falls into a position very close to its neighboring points on the grid. Operating on an object distance to its immediate neighbor, it then measured how much each grid point, as a new object, conforms to the underlying distribution. The two used conformity measures include *k-Nearest Neighbours* (k-NN) and *Kernel Density Estimation* (KDE). Subsequently, conform points to the dataset that are not anomalies, were obtained with respect to the chosen significance level. Conforming points were then clustered using a neighboring rule in such a way that two points which are neighbors on the grid are assigned to the same cluster. Eventually, the detection set data was associated with the resulting clusters where its distance from one of the cluster points is smaller or equal to the grid unit. Moreover, If an object was assigned to more than one cluster, those clusters were merged together.

In our experiments, this prescribed clustering technique is applied to the training set with the first-Nearest neighbor and Kernel density bandwidth 0.1 which is shown to achieve

the best results, in separate experiments. Eventually, for each experiment, the detection capability of the resulting clusters or network behavior models is assessed over all the remaining data. Table 1 provides the results obtained by the method with KDE with bandwidth 0.1. This table also provides the results obtained by the method with the first-Nearest neighbor. We set the significance level as 0.2 and the number of points per side of the grid l is set to 35 (Cherubin, 2014). The first column g gives the true numbers of clusters. The second column $\hat{g}$ gives the estimated numbers of clusters.

Conformity Measure	g	ĝ	NMI	Precision	Rand Index	Recall	F-Measure
k-NN	5	21	0.612959	1.000000	0.652406	0.071429	0.133333
KDE	5	21	0.612959	1.000000	0.652406	0.071429	0.133333

Table 1: Results obtained during conformal clustering proposed in the reference article with k-NN and KDE.

As shown in Table 1, in clustering, 34 detection set objects were assigned to 21 clusters ($\hat{g}$), while the true number of clusters (g) is 5. Therefore, many small clusters reduced the NMI and reasonably decreased detection rates. The reason is that *Purity* was used as the measure of clustering quality. However, Purity provides no trade-off between the clustering quality and the number of clusters. Hence, a higher Purity measure is achieved when the number of clusters is larger. While the true number of clusters is 5, as it is shown, 21 clusters are produced with this approach.

This method has low values on F-Measure and Recall. This could be due to the fact that many similar data points are assigned to different clusters for which the amount of false negative is increased. Precision reached its maximum because an increase in the number of clusters decreases the chance of putting two dissimilar data points in the same cluster, which leads to a lower false positive rate. On the other hand, an increase in the number of clusters increases the possibility that two dissimilar data points are put in different clusters, which affects the amount of true negative and thus, increases the Rand Index.

It is worth noting that, it used conformal prediction to construct a sequential prediction set on a large-sized grid. Thus, the training object falls into a position very close to its neighboring points on the grid. Thereby, constructing a prediction set with the first-nearest neighbor and Kernel density estimator with a too-small bandwidth that is shown to achieve the best results, without considering clusters as high-density regions of the underlying distribution, would make the grid point conformity more likely, even when that point falls into a position very close to an outlier. Besides, this approach led to producing portions of the feature space trimmed by the chosen significance level, which is an additional workload as well as computationally expensive.

4.4. Evaluating The Proposed Approach

Second, we apply our method to the corresponding training set. In general, prediction sets are constructed using the aforementioned three density-based conformity measures, each in a separate experiment. Moreover, in each experiment, there are hyperparameters that we

have to deal with, including the grid size l, α, and the corresponding hyperparameter in the used conformity measure. To tune these hyperparameters, we have to select a set of values, on an appropriate scale, to explore the hyperparameters. To this end, we randomly sample the points, accompanied by a coarse to fine sampling scheme, and try the corresponding hyperparameters on this randomly chosen set of points and then pick whichever hyperparameters lead to the best clustering result.

Conformity Measure	g	$\hat{g}$	NMI	Precision	Rand Index	Recall	F-Measure
k-NN density estimator	5	5	1.000000	1.000000	1.000000	1.000000	1.000000
KDE	5	7	0.891860	1.000000	0.910873	0.621212	0.766355

Table 2: Results obtained by the proposed conformal clustering approach with k-NN density estimator and KDE.

Table 2 gives the best result obtained by our approach with a good setting for l, h and α in KDE in which these tuning parameters are set to be 6, 0.5 and 0.1, respectively. This table also gives the best result obtained with a good setting for l , k, and α in k-NN density estimator in which these hyperparameters are set to be 6, 10, and 0.25, respectively. The first column gives the true numbers of clusters. The second column gives the estimated numbers of clusters. The results show that our technique improves upon the approach proposed in the original paper. The clustering approach with k-NN density correctly estimates the number of clusters and achieve high values on the six quality rating criteria. Although the true number of clusters is not accurately achieved by KDE, the estimated number of clusters is 7 which is very close to the truth. Our approach got a much larger value on the F-measure and Recall. The precision reached its maximum truly. Furthermore, the NMI value increased. Here, one true cluster is subdivided into smaller clusters that affected the values of NMI, Rand Index, and Recall.

Conformity Measure	g	$\hat{g}$	NMI	Precision	Rand Index	Recall	F-Measure
Entropy-based	5	4	0.948357	0.891892	0.971480	1.000000	0.942857

Table 3: Results obtained by the proposed conformal clustering approach with entropy-based conformity measure.

Table 3 gives the best result obtained by our approach with a good setting for l, σ and α in entropy-based conformity measure in which these hyperparameters are set to be and 6, 5 and 0.5, respectively. As shown in Table 3, our clustering approach with this conformity measure achieves high quality.

5. Conclusions and Future Works

In this paper, we described a conformalized density-based clustering technique, improving previous multi-level conformal clustering approach. We presented its application for cluster-

ing analysis of malicious traffic generated by bots. The objections of previous research were addressed by establishing a close connection between conformal clustering and density-based clustering. In this paper, we used some measures of clustering quality and detection capability including NMI, Recall, Rand Index, and F-measure. The results show that our approach using k-NN density estimator gives the desired number of clusters and also achieves high values on the corresponding quality rating criteria. Furthermore, it improves the previous research with KDE by increasing the NMI and Rand Index up to 30 percent. Moreover, the values of Recall and F-measure are increased up to 70 percent, and maximum precision is truly reached. Besides, we introduced a new entropy-based conformity measure based on the cluster quality rating function in BotFinder by which our clustering approach achieves high quality.

As previously mentioned, the proposed conformal prediction-based clustering approach is closely related to the density-based clustering. Our future work includes some research to investigate the efficiency of using conformal prediction in combination with the density-based clustering methods such as DBSCAN. Furthermore, we would like to propose an unsupervised machine learning approach for botnet detection that is closely connected to the underlying algorithm of BotFinder by using the aforementioned entropy-based measure.

References

Pedram Amini, Muhammad Amin Araghizadeh, and Reza Azmi. A survey on botnet: classification, detection and defense. In *2015 International Electronics Symposium (IES)*, pages 233–238. IEEE, 2015.

Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the mirai botnet. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1093–1110, 2017.

Giovanni Cherubin. *Bots detection by Conformal Clustering*. PhD thesis, Royal Holloway, URL https://giocher.com/files/docs/bdcc-msc-thesis.pdf, 2014.

Giovanni Cherubin, Ilia Nouretdinov, Alexander Gammerman, Roberto Jordaney, Zhi Wang, Davide Papini, and Lorenzo Cavallaro. Conformal clustering and its application to botnet traffic. In *International Symposium on Statistical Learning and Data Sciences*, pages 313–322. Springer, 2015.

Christian J Dietrich, Christian Rossow, and Norbert Pohlmann. Cocospot: Clustering and recognizing botnet command and control channels using traffic analysis. *Computer Networks*, 57(2):475–486, 2013.

Alexander Gammerman and Vladimir Vovk. Hedging predictions in machine learning. *The Computer Journal*, 50(2):151–163, 2007.

Sheharbano Khattak, Naurin Rasheed Ramay, Kamran Riaz Khan, Affan A Syed, and Syed Ali Khayam. A taxonomy of botnet behavior, detection, and defense. *IEEE communications surveys & tutorials*, 16(2):898–924, 2013.

Jing Lei, James Robins, and Larry Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

SC Ng. Principal component analysis to reduce dimension on digital image. *Procedia computer science*, 111:113–119, 2017.

Ilia Nouretdinov, James Gammerman, Matteo Fontana, and Daljit Rehal. Multi-level conformal clustering: A distribution-free technique for clustering and anomaly detection. *Neurocomputing*, 2019.

Jaehyeok Shin, Alessandro Rinaldo, and Larry Wasserman. Predictive clustering. *arXiv preprint arXiv:1903.08125*, 2019.

James Smith, Ilia Nouretdinov, Rachel Craddock, Charles Offer, and Alexander Gammerman. Anomaly detection of trajectories with kernel density estimation by conformal prediction. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 271–280. Springer, 2014.

Florian Tegeler, Xiaoming Fu, Giovanni Vigna, and Christopher Kruegel. Botfinder: Finding bots in network traffic without deep packet inspection. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 349–360. ACM, 2012.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Conformal prediction*. Springer, 2005.