Dynamic Decision Boundary for One-class Classifiers applied to non-uniformly Sampled Data

Riccardo La Grassa, Ignazio Gallo, Nicola Landro
University of Insubria,
Department of Theoretical and Applied Sciences,
Varese, Italy
Email: rlagrassa, ignazio.gallo, nlandro}@uninsubria.it

Abstract—A typical issue in Pattern Recognition is the non-uniformly sampled data, which modifies the general performance and capability of machine learning algorithms to make accurate predictions. Generally, the data is considered non-uniformly sampled when in a specific area of data space, they are not enough, leading us to misclassification problems. This issue cut down the goal of the one-class classifiers decreasing their performance. In this paper, we propose a one-class classifier based on the minimum spanning tree with a dynamic decision boundary (OCDmst) to make good prediction also in the case we have non-uniformly sampled data. To prove the effectiveness and robustness of our approach we compare with the most recent one-class classifier reaching the state-of-the-art in most of them. The code and all results are available online https://gitlab.com/artelabsuper/ocdmst.

I. INTRODUCTION

In the deep learning era, the best models need to have many hundreds or thousands of instances to make good predictions and to reduce as little as possible misclassification. However, in the scenario where we have a few samples of a specific class without knowing anything about other classes, the neural network models can not be used. In this case, different approaches exist that uses neural networks for this task ([11]). They train the model using a dataset of a different context than the one in which the model will then be used. Although it offers good results in terms of classification, it does not be applied to a specific context, for instance, in case the source of the test set is very different from the data of the pre-trained model and consequently, the discrimination power model will be very low to generalize these test instances. In conjunction with few objects available and time required to train a model with a huge dataset, deep models might not be the right choice for this specific task. In [2], we used a neural network pre-trained on a large dataset as Imagenet to extract deep features and use them as input for a one-class classifier (OCC) based on MST (OCSmst). They introduce another level of boundary used to label all uncertain images and construct another one-class MST using all negative samples classified. The final step will classify all uncertain value starting from a couple of MST. Although the idea is interesting they do not handle dynamic boundaries but only different levels of static boundaries are treated. In literature, various approaches exist for one-class classification. For example, [3] and [4] use ensemble methods to find the best partitions or to handle under/over-sampling in an unbalanced scenario. A classical SVM revised algorithm (OCSVM) ([5]) offers good results with different kinds of data. Other interesting models are graph-based ([6], [7], [8], [9]) and occur on a different domain as computer vision ([10], [11]), bioinformatics ([6], [11]), clustering ([9]).

Furthermore, in the literature, other well-known models exist, like hybrid models. In this scenario, neural network models are used as deep features extractors jointly to classical machine learning classification algorithms like a one-class support vector machine. ([5], autoencoder+ocsvm [12], autoencoder plus knn [13], autoencoder plus svdd [14]). Intuitively, the main advantages of using hybrid models are the reduced number of instances and to exploit the power of neural network models to provide the best discriminative features. In [15] and [10], the authors propose an approach to combine Convolutional Neural Networks (CNN) with SVDD, while optimizing the class boundary in the output features space. Some disadvantages of this methodology are the computational time required for the training step and in the scenario which we have few data, this approach might not have good results.

In this paper, we propose an extension of our previous works [7] and [8], using a new approach based on the inverse sigmoid function to handle the decision boundary in a dynamic methodology. We conduct a wide variety of experiments on different datasets to demonstrate the rightness of our model and compare it with the most recent one-class classifiers, overcoming them in many tasks. Today, finding good models in particular situations such as in medicine is a necessity. We tested our model on Covid-19 disease using clinical images from [16] and [17], obtaining interesting results. In the following sections, we describe the main approach using a minimum spanning tree as class descriptor II, we introduce our approach IV and finally, we will discuss our results compared with other known models V.

II. ONE-CLASS CLASSIFIERS

One-class novelty detection refers to the recognition of abnormal patterns on data recognized as normal. Abnormal data, also know as outliers are patters who belong to the different classes than a normal class. The goal of the novelty detection field is to distinguish anomaly patterns that are different from normal and classify them. The capability of many machine learning technique, in the field of one-class
classification is to decide whether a new instance belongs to the same class of data or if it has different behaviour such as to be considered as an outlier. Designing an OCC classifier is not a simple task due to the nature of data available and from the quality of data sampled. In Tab. I we show the general capabilities of one-class classifiers.

In Tab. I the data has multi-modality and exists more than one distribution in it [18]. The general approach of multi-density is not a simple task. An object might have many levels of significance in the proximity of decision boundary showing that a single density level in some cases can be reductive in terms of classification. The noise in the data can lower the general performance of classifiers lead it to misclassification. Arbitrarily shaped distribution of the target class is a common feature of the MST, SVDD or MoG models, that can cover the data according to their methodology.

In [19], the authors propose an interesting approach based on density modular ensemble classifiers for the one-class classification task. They applied a pruning method to split the original structure of an MST, built from the target class while creating different clusters. In this work, no cluster algorithms are applied. They used a Gaussian estimator algorithm to build a decision border for each cluster created in the previous step. Parameters considered are the level of the pruning step and all parameters behind the one-class chosen for each partition built. The authors describe some models related to OCSVM, where the authors mention that for each partition created in the previous step, a small partition of the target class as a class descriptor.

| Models                  | multi-modality | multi-density | noise | Shaped on target class |
|-------------------------|----------------|---------------|-------|------------------------|
| Gaussian estimator      | ✗              | ✗             | ✗     | ✗                      |
| MoG                     | ✗              | ✗             | ✗     | ✗                      |
| MST_CD                  | ✗              | ✗             | ✗     | ✗                      |
| OCmst                   | ✗              | ✗             | ✗     | ✗                      |
| OC-SVM (RBF)            | ✗              | ✗             | ✗     | ✗                      |
| SVDD (RBF)              | ✓              | ✓             | ✓     | ✓                      |
| OCmst (our)             | ✓              | ✓             | ✓     | ✓                      |

In the following section, we discuss the main approach using an MST as a class descriptor and then introduce our proposed model.

III. THE MAIN APPROACH USING A MINIMUM SPANNING TREE

Many task predictions have obtained successful results using specific structure in Graph Theory as spanning trees and have found many application contexts in the social network ([21]), biology ([22]), companies and political events ([23]). In this section, we focus on the minimum spanning tree used as a class descriptor and highlight the pros and cons of this model.

This brief overview is useful to understand our approach and a possible solution to increase the performance of this classifier. In general, MST_CD model uses the following two alternatives to measure the distance of a point \(x\) from the graph:

- projection \(p_{e_{i,j}}(x)\) of a point \(x\) on an edge \(e_{i,j} = (x_i, x_j)\), defined as follow
  \[
  p_{e_{i,j}}(x) = x_i + \frac{(x - x_i)^T(x - x_i)}{|x_i - x_j|^2}(x_j - x_i)
  \]
- \(\min(||x - x_j||, ||x - x_i||)\).

If \(p_{e_{i,j}}(x)\) lies on the edge \(e_{i,j} = (x_i, x_j)\), we compute \(p_{e_{i,j}}(x)\) and the Euclidean distance between \(x\) and \(p_{e_{i,j}}(x)\), more formally:

\[
0 < \frac{(x - x_i)^T(x - x_i)}{|x_j - x_i|^2} < 1
\]

then

\[
(x*e_{i,j}) = ||x - p_{e_{i,j}}(x)||
\]

Otherwise, we compute the Euclidean distance of \(x\) and the two points \(x_i\) and \(x_j\). More precisely:

\[
d(x|e_{i,j}) = \min(||x - x_j||, ||x - x_i||)
\]
TABLE II: Performance comparison of OCdmst and Resnet18 on Covid-19 dataset. We highlight the capability of OCdmst to make prediction even when the neural network does not recognize values from negative class. Target class used is Positive label.

| K-Fold | Positive class | Negative class |
|--------|----------------|----------------|
|        | OCdmst         | Resnet18       |
| 1      | 0.853          | 0.347          |
| 2      | 0.843          | 0.187          |

Then, a new instance $x$ is recognized from an MST if it lies within the boundary, otherwise, the object is considered an outlier. In [6], [8], [7] the decision whether an object is recognized by classifier or not is based on a threshold $\theta$ applied to the shape created in this phase. More formally:

$$d(x|X) \leq \theta$$

(5)

The threshold $\theta$ is a parameter used to assign the boundary dimension of a spanning tree. Then, given $\hat{e} = (|e_1|, |e_2|, ..., |e_n|)$ as an ordered set of edges weights, we define $\theta$ as

$$\theta = ||e_{[\alpha n]}||,$$ where $\alpha \in [0, 1]$ (6)

For instance, with $\alpha = 0.5$, we assign to $\theta$ the median value of all edges weights of the spanning tree. The curse of the number of instances is a possible weakness that is reflected in an increase in terms of computational time, it also does not handle non-uniformly sampled data. This lack results in a too-large decision border in the whole area with few data.

IV. OUR APPROACH

Starting with our previous works [8] and [7] we use the same baseline called MST_CD gp. This last approach is different from [6] because it uses a couple of small MST built for each test instance to make a prediction and it handles the case in which both classifiers have right or wrong. To remain in the context of a one-class classifier, we use only a target class of size $\gamma$ to make predictions. In previous papers, we catch the structure of the target class using MST and built a decision boundary $\theta$ using Eq. 6. However, authors do not consider the problem of non-uniformly sampled data. In specifics regions of the euclidean space, it is possible to find the sparse zone and consequently to obtain a good boundary from the target class (see red line in Fig.4) can become a problem. To remedy this problem we need to build a general methodology to create a dynamic decision boundary and avoid the wrong threshold due to sparse data. For each test instance, we search the first nearest $\gamma$ nodes and use the Kruskal algorithm to build an MST_CD gp, then we simply apply a Breadth-first search starting to the nearest object to test instance. In this way we preserve the structure around a target class created by an MST of size $\gamma$ and takes $N$ nodes by BFS search, more formally, given $\hat{e} = (|e_1|, |e_2|, ..., |e_n|)$ be a set of weighted edges extracted by a BFS search algorithm applied on the MST_CD gp, we introduce a dynamics threshold $\theta$ as:

$$\hat{\theta} = ||BFS_{[\alpha n]}|| \ast \frac{1}{1 + e^{K(\hat{\sigma} - \hat{\sigma}_rg + \beta)}}$$

(7)

where $\alpha = 0.5$ in order to take the median of all the edges weights, $\hat{\sigma}$ is the normalized standard deviation of all nodes selected by BFS with variance computed with his data instances distribution (see blue line in Fig.4) and $K$ is the logistic growth rate or steepness of the curve (we used $K = 5$). $\hat{\sigma}_rg$ represents the median value of all standard deviation computed taking $N$ random group having the size returned by the BFS with depth $d$. This last is an important value that defines the inflection point of our inverse sigmoid. Furthermore, we use a variable $\beta$ useful to shift the inflection point and thus we slow the fall to 0 (see Fig. 2).

Note that our mean value of standard deviation will be the centroid computed considering all target class instances. From Popoviciu’s inequality ([24]), it is well-known that the upper bound of the variance of any random variable $X$ on a range $[x_{\min}, x_{\max}]$ is equal to:

$$V[X] \leq \frac{(x_{\max} - x_{\min})^2}{4}$$

(8)

Since we handle instances $x_i$ with dimension $d > 1$, the deviation standard is:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{d \ast n}}$$

(9)

Keep in mind the relationship in Eq. 8, the upper bound of $\sigma$
For all experiments we chose $\beta = 1.5$ to move the inflection point of the curve with respect to the median value of all the standard deviations starting from $N$ random samples in small MST built.

In most of the datasets used we obtain better results using BFS with different levels of depth on Covid-19 dataset. In most of the experiments, we reported total accuracy as performance metrics.

To prove the effectiveness and robustness of our OCdmst, we use the same experimental setting used in [25] and compute Matthews’s Coefficient Correlation (MCC) and in some experiments, we reported total accuracy as performance metrics. Matthews’s Coefficient Correlation (MCC) and in some experiments, we reported total accuracy as performance metrics.

V. EXPERIMENTS

To prove the effectiveness and robustness of our OCdmst, we use the same experimental setting used in [25] and compute Matthews’s Coefficient Correlation (MCC) and in some experiments, we reported total accuracy as performance metrics. In the binary classification problem, a model can obtain high accuracy on outliers class but very low performance to recognize target class or vice versa. In such a situation, using a global metric as the classical accuracy is not the right way to compute the classification score. To avoid this issue we use MCC as a metric that can be used even if we have classes with different sizes. MCC returns a value between -1 and +1 where a coefficient of +1 represents a perfect prediction, 0 random predictions and -1 indicates total
Fig. 4: Overview of OCdmst algorithm. This figure emphasizes the dynamic border relationship with the nearest instances.

Fig. 5: In (a) the BFS path results starting from the node closest to the $X$ test using $depth = 2$ and $\gamma = 9$. In (b), the structure created without the use of BFS. The difference is that in (a) we capture the neighbor’s relations in terms of connectivity instead in (b) we use the Euclidean distance.

disagreement between prediction and observation.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

(12)

All our experiments use five-fold cross-validation to split target class. To compose the complete test set we use the fold used as a test, together with all the outliers samples. Finally, we repeat the approach twenty times taking means and variance to compare our results with other models. In our experiments we use six numerical datasets from UCI repository (see, Table. V) and a novel dataset of Covid-19 that contains chest X-ray or CT images of positive/negative patients (see Fig. 1). In order to have the right comparison criterion in Covid-19 dataset, we extract confusion matrix as reported in Tab. IV and compute the precision metric described in Eqs. 13 and 14. These last measures are needed to compare the class accuracy of the proposed OCdmst and the Resnet18 used in our experiments.

$$\text{PPV} = \frac{TP}{TP + FP}$$

(13)

$$\text{NPV} = \frac{TN}{TN + FN}$$

(14)

Like for OCSVM and many other models, we need to do fine-tuning to find good parameters and obtain good results. Considering the lack of MST due to its high computational time, it is not easy to find optimal parameters. Therefore, we use BFS with depth $d \in \{1, ..., 7\}$ for all UCI dataset (see details in Tab. V), $d \in \{2, ..., 11\}$ only for Covid-19 dataset, a growth rate $k \in \{5, 6, 7, 8, 9, 10, 20\}$ and a variable $\beta \in \{1.05, 1.1, 1.5\}$ to shift the logistic function (see Fig. 2).

Furthermore, in fine-tuning parameters, we set the size of small MST from the $\frac{\text{length target class}}{2}$ to $\frac{\text{length target class}}{4}$.

A. First experiment

In this section, we compare our OCdmst with others one-class classifiers more recent in the literature showing the results in Tab. III. For clarity, in Liver dataset, we use the last field for classification. Also if it does not represent the presence/absence of a liver disorder, we have to use it to obtain the comparison with the other papers. In the Heart dataset (Cleveland) we distinguish values 1,2,3,4 as the presence of disease, 0 otherwise and removed instances with missing values according to the experiments in the literature. Results demonstrate the rightness of our model and as reported in Tab. III we outperform the state-of-the-art in terms of the global mean for many classifiers (except FS-EOCC) and for many datasets.

B. Second experiment

In the second experiment, we analyze our model with MCC accuracies in y-axis and the depth from 1 to 7 for each size of small MST (x-axis) (see Fig. 6, 3). Depth of BFS and size of MST are useful parameters to set up our model making as less as possible wrong classifications.

C. Third experiment

In the third experiment, we demonstrate by experimental results the effectiveness of our model with few data and compare it with a neural network (Resnet18). We simulate the scenario in which we have too few data to deny common neural networks to make right discrimination of different concepts (two classes in our case). Therefore we apply a 2-Fold cross-validation considering all data available from Covid-19 dataset and we extract the deep features for each model trained (Resnet18). All images have been re-scaled to 256x256 pixels and transformed to grey-scale. Learning rate and batch-size are set to 0.001 and 10 respectively. As evaluation metric, we do not consider the total accuracy but
TABLE III: Performance comparison of OCdmst and state-of-the-art OCC methodology on UCI datasets using MCC metric. In bold our average result.

| Dataset | FS-EOCC (Gauss) | FS-EOCC (MST_CD) | TOCC | OCClustE (Gauss) | OCClustE (MST_CD) | Gauss | MST_CD | OC-SVM | OCdmst |
|---------|----------------|-----------------|------|-----------------|-----------------|-------|--------|--------|--------|
| Breast (Benign) | 0.765 (0.006) | 0.864 (0.003) | 0.847 (0.004) | 0.899 (0.001) | 0.905 (0.003) | 0.895 (0.006) | 0.894 (0.002) | 0.473 (0.002) | 0.774 |
| Breast (Malignant) | 0.471 (0.022) | 0.218 (0.009) | 0.215 (0.012) | 0.208 (0.027) | 0.313 (0.009) | 0.19 (0.007) | -0.015 (0.011) | 0.234 (0.005) | 0.204 |
| Diabetes (Absent) | 0.096 (0.008) | 0.005 (0.002) | 0.05 (0.003) | -0.01 (0.005) | 0.051 (0.004) | 0.009 (0.005) | -0.015 (0.004) | 0.010 (0.007) | 0.066 |
| Diabetes (Present) | 0.186 (0.004) | 0.233 (0.008) | 0.209 (0.003) | 0.228 (0.004) | 0.21 (0.012) | 0.161 (0.016) | 0.206 (0.003) | 0.203 (0.016) | 0.178 |
| Glass (Float) | 0.505 (0.017) | 0.567 (0.016) | 0.471 (0.012) | 0.452 (0.006) | 0.507 (0.005) | 0.45 (0.011) | 0.422 (0.005) | 0.389 (0.005) | 0.535 |
| Glass (NoFloat) | 0.375 (0.022) | 0.32 (0.003) | 0.314 (0.015) | 0.329 (0.004) | 0.336 (0.002) | 0.25 (0.015) | 0.267 (0.017) | -0.149 (0.035) | 0.238 |
| Heart (Present) | 0.082 (0.029) | 0.095 (0.026) | 0.004 (0.017) | 0.001 (0.010) | 0.03 (0.011) | 0.01 (0.021) | 0.071 (0.005) | 0.069 (0.005) | 0.037 |
| Heart (Absent) | 0.082 (0.037) | 0.231 (0.006) | 0.115 (0.011) | 0.095 (0.024) | 0.249 (0.004) | 0.328 (0.008) | 0.161 (0.025) | -0.005 (0.020) | 0.117 |
| Liver (Disorder) | 0.151 (0.001) | 0.016 (0.03) | 0.001 (0.006) | -0.012 (0.01) | 0.124 (0.006) | -0.084 (0.009) | -0.06 (0.022) | 0.056 (0.005) | 0.099 |
| Liver (Healthy) | 0.94 (0.006) | 0.033 (0.022) | 0.053 (0.004) | 0.07 (0.014) | 0.112 (0.01) | 0.062 (0.018) | 0.047 (0.01) | 0.065 (0.002) | 0.073 |
| Sonar (Mines) | 0.42 (0.11) | 0.563 (0.006) | 0.511 (0.009) | 0.396 (0.039) | 0.351 (0.008) | 0.356 (0.031) | 0.255 (0.008) | 0.133 (0.019) | 0.672 |
| Sonar (Rocks) | 0.215 (0.005) | 0.147 (0.02) | 0.045 (0.15) | 0.11 (0.009) | 0.108 (0.014) | 0.151 (0.027) | 0.166 (0.032) | 0.054 (0.233) | 0.356 |
| Average | 0.285 | 0.274 | 0.237 | 0.233 | 0.275 | 0.23 | 0.2 | 0.128 | 0.277 |

TABLE IV: Confusion matrix template. In x-axis predicted values, y-axis true labels value.

| True labels | Predicted Positive | TP | FP | Positive Negative | NF | TN |
|-------------|-------------------|----|----|-------------------|----|----|
| Positive    |                   |    |    |                    |    |    |
| Negative    |                   |    |    |                    |    |    |

TABLE V: Number of features, instances for all the datasets used in our experiments.

| Datasets     | Features | Instances |
|--------------|----------|-----------|
| Sonar        | 60       | 208       |
| Liver        | 5        | 345       |
| Breast Cancer (Wisconsin) | 9 | 699 |
| Diabetes     | 8        | 768       |
| Glass        | 9        | 214       |
| Heart        | 13       | 297       |

we evaluate the accuracy for each class (Precision). Then, we use the deep features as input to our OCdmst using the dataset split in the previous step. The results reported in Tab. II show the flexibility of our model to operate also with Resnet18 learned features using cross-entropy loss. As showed in Tab. II, Resnet18 does not offer the right prediction to recognize instances from negative class. This result is reasonable considering we have too few data with negative labels. Even when we used features extracted by Resnet18 to our model, but OCdmst can recognize value from both classes and overcome the right prediction also on positive class. The confusion matrix showed in Fig. 7 and accuracies for each class demonstrate the capability of OCdmst to operate also in a scenario with few data, as well as overcome deep neural networks. This last result is plausible because it represents one of the weaknesses of deep models.

VI. CONCLUSION

In machine learning, non-uniformly sampled data is a problem that normally leads the predictive models to make misclassifications. To contrast this issue, in this paper, we propose a dynamic boundary approach for one-class classification able to handle their boundary also in a non-uniformly sampled data. The effectiveness and solidity of our approach have been compared with many one-class classifiers and for many of them, we have passed the state of the art. We believe that our model can be improved in terms of computational time, to be applied also in computer vision.

REFERENCES

[1] P. Perera and V. M. Patel, “Learning deep features for one-class classification,” IEEE Transactions on Image Processing, vol. 28, no. 11, pp. 5450–5463, 2019.
[2] R. La Grassa, I. Gallo, and N. Landro, “Ocstm: One-class novelty detection using convolutional neural network and minimum spanning trees,” 2020.
[3] D. M. Tax and R. P. Duin, “Using two-class classifiers for multiclass classification,” vol. 2, pp. 124–127, 2002.
[4] B. Krawczyk, M. Galar, M. Woźniak, H. Businicle, and F. Herrera, “Dynamic ensemble selection for multi-class classification with one-class classifiers,” Pattern Recognition, vol. 83, pp. 34–51, 2018.
[5] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” Neural computation, vol. 13, no. 7, pp. 1443–1471, 2001.
[6] P. Juszczak, D. M. Tax, E. Pe, R. P. Duin et al., “Minimum spanning tree based one-class classifier,” Neurocomputing, vol. 72, no. 7-9, pp. 1859–1869, 2009.
[7] R. La Grassa, I. Gallo, A. Calefati, and D. Ognibene, “A classification methodology based on subspace graphs learning,” pp. 1–8, 2019.
[8] R. La grassa, I. Gallo, A. Calefati, and D. Ognibene, “Binary classification using pairs of minimum spanning trees or n-ary trees,” 2020.
[9] J. Liu, Q. Miao, Y. Sun, J. Song, and Y. Quan, “Fast structural ensemble detection using convolutional neural network and minimum spanning trees,” 2019.
Fig. 6: Analysis using different parameters on UCI datasets. In x-axis the size of small MST and y-axis the depth used in BFS search. Red line and blue line represent the related max and standard deviation of MCC accuracies.

[10] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, “Deep one-class classification,” pp. 4393–4402, 2018.
[11] Y. Zhang, B. Zhang, F. Coenen, J. Xiao, and W. Lu, “One-class kernel subspace ensemble for medical image classification,” EURASIP Journal on Advances in Signal Processing, vol. 2014, no. 1, p. 17, 2014.
[12] J. T. Andrews, E. J. Morton, and L. D. Griffin, “Detecting anomalous data using auto-encoders,” International Journal of Machine Learning and Computing, vol. 6, no. 1, p. 21, 2016.
[13] H. Song, Z. Jiang, A. Men, and B. Yang, “A hybrid semi-supervised anomaly detection model for high-dimensional data,” Computational intelligence and neuroscience, vol. 2017, 2017.
[14] S. Kim, Y. Choi, and M. Lee, “Deep learning with support vector data description,” Neurocomputing, vol. 165, pp. 111–117, 2015.
[15] R. Chalapathy, A. K. Menon, and S. Chowla, “Anomaly detection using one-class neural networks,” arXiv preprint arXiv:1802.06360, 2018.
[16] radiopaedia, https://radiopaedia.org/.
[17] sirm, https://www.sirm.org/.
[18] M. Hovelynck and B. Chidlovskii, “Multi-modality in one-class classification,” pp. 441–450, 2010.
[19] J. Liu, Q. Miao, Y. Sun, J. Song, and Y. Quan, “Modular ensembles for one-class classification based on density analysis,” Neurocomputing, vol. 171, pp. 262–276, 2016.
[20] D. M. Tax and R. P. Duin, “Support vector data description,” Machine learning, vol. 54, no. 1, pp. 45–66, 2004.
[21] Z. Wang, M. Hou, G. Yuan, J. He, J. Cui, and M. Zhu, “Hierarchical community detection in social networks based on micro-community and minimum spanning tree,” IEICE TRANSACTIONS on Information and Systems, vol. 102, no. 9, pp. 1773–1783, 2019.
[22] C. Isaza, J. F. Rosas, E. Lorenzo, A. Marrero, C. Ortiz, M. R. Ortiz, L. Perez, and M. Cabrera-Rios, “Biological signaling pathways and potential mathematical network representations: biological discovery through optimization,” Cancer medicine, vol. 7, no. 5, pp. 1875–1895, 2018.
[23] B. A. Memon, H. Yao, and R. Tahir, “General election effect on the network topology of pakistan’s stock market: network-based study of a political event,” Financial Innovation, vol. 6, no. 1, p. 2, 2020.
[24] T. Popoviciu, “Sur les équations algébriques ayant toutes leurs racines réelles,” Mathematica, vol. 9, pp. 129–145, 1935.
[25] J. Wang, W. Liu, K. Qui, H. Xiong, and L. Zhao, “Dynamic hypersphere svdd without describing boundary for one-class classification,” Neural Computing and Applications, vol. 31, no. 8, pp. 3295–3305, 2019.