Ensemble Consensus: an unsupervised algorithm for anomaly detection in network security data

Vincenzo Dentamaro[1], Vito Nicola Convertini[1], Stefano Galantucci[1], Paolo Giglio[1], Tonino Palmisano[1] and Giuseppe Pirlo[1]

[1] University of Bari, Dep. Of computer Science, Via Orabona 4, Bari, Italy

Abstract
Unsupervised network traffic monitoring is of paramount importance in cyber security. It allows to detect suspicious events that are defined as non-normal and report or block them. In this work the Anomaly Consensus algorithm for unsupervised network analysis is presented. The algorithm aim is to fuse the three most important anomaly detection techniques for unsupervised detection of suspicious events. Tests are performed against the KDD Cup '99 dataset, one of the most famous supervised datasets for automatic intrusion detection created by DARPA. Accuracies reveal that Anomaly Consensus performs on-par with respect to state-of-the-art supervised learning techniques, ensuring high generalization power also in borderline tests when small amount of data (5%) is used for training and the rest is for validation and testing.

Keywords
Unsupervised anomaly detection, isolation forest, elliptic envelope, local outlier factor, anomaly detection logs

1. Introduction

Computer systems, especially the networks that interconnect them, must be carefully monitored to avoid computer attacks. The detection of an attack is generally done through two methods: knowing the attack, one tries to understand if it is present in the analyzed traffic, or one goes to analyze if there are variations between the analyzed traffic and the "normal" network traffic.

The first approach is not always applicable, as it requires a large amount of information, i.e., a mapping of all possible attacks (signature) and such

(ITASEC) Italian Conference on Cybersecurity, April 7-9, 2021, Italy
EMAIL: name.surname@uniba.it
ORCID: 0000-0003-1148-332X (A. 1); 0000-0002-2567-5819 (A. 2); 0000-0002-3955-0478 (A. 3); 0000-0002-0426-7349 (A. 4); 0000-0002-6311-246X (A. 5); 0000-0002-7305-2210 (A. 6);

information must be in a format that allows its application to the analyzed problem. Moreover, the first approach does not allow in any way the revelation of unknown attacks.

A greater protection is given by the second approach, i.e., through the use of machine learning [16-17], releasing, in fact, the concept of the attack from the detection system, treating the traffic as mere data. The task to be solved is therefore linked to anomaly detection. The use of classifiers [13],[15] presents a number of problems: firstly, the availability of an adequate source of data on traffic under attack is not always available (absent data or unbalanced classes); secondly, there are difficulties in differentiating normal traffic from anomalous traffic.

In cybersecurity contexts, a non-excessive presence of false positives is acceptable since it is still up to the operator to evaluate suspicious traffic.

In this paper an algorithm based on the union of three anomaly detection algorithms (Elliptic Envelope, Isolation Forest, Local Outlier Factor) is proposed.

In the next section the state of the art of anomaly detection systems for cybersecurity will be described, while in section 3 the functioning of the Ensemble Consensus algorithm will be explained in detail. In section 4 the data of the conducted experiment are presented, and the evaluation of the results, conclusions and future developments are deferred to the last section.

2. State of the art review

Anomaly detection systems use machine learning to store the normal state of a batch of data and identify which of these deviates heavily from normal. Such approaches assume a character of primary importance within Intrusion Prevention/Detection Systems. In the following state of the art, the most interesting approaches proposed for the latter will be analyzed, as contextualization to cybersecurity domains and aimed at network data is of fundamental interest in the article.

Authors in [1] proposes an anomaly detection module uses a Self-Organizing Map (SOM) structure to map and model normal behavior. Anything that differs from normal behavior is classified as an attack. SOM relies on unsupervised learning to map nonlinear statistical relationships between high-dimensional input data into the output space, a two-dimensional lattice or grid. SOMs place highly correlated patterns in contiguous locations in the resulting output space with good results and provide visualization and projection options for high-dimensional data. Finally, there is a module that allows to determine the type of attack, called abuse detection module, which makes use of a J.48 decision tree. The model is trained with only normal traffic data. The classification of a connection into the classes of normal/anomalous is done through the parameters of quantization error (q_e) and best matching unit (bmu). If the quantization error is greater than a fixed threshold, it is anomalous.

Authors in [2] propose an Intrusion Detection System model that uses the Neighborhood Outlier Factor (NOF). The main idea of this approach is to assign to each data example a ranking of “outlier”, the NOF, and search for data that is very different from a heavy amount of data representing normality.

Important in the use of such a model is the amount of data: although the system recognizes almost all types of attacks, where the number of outliers exceeds the dataset that identifies normal behavior [14], the system behaves as an intrusion dataset. The great advantage of this approach is the execution time, which proves to be less than any other classifier machine learning, this potential makes it an interesting system for approaches to the problem of type online or highly responsive, which can be, in fact, the domain of cybersecurity. This advantage comes from the fact that less trained datasets, so calculating the distance between the training dataset and the testing dataset is particularly easy.

A hybrid approach to detect unknown attacks by clustering is proposed by [3]. The algorithm is called SSC-OCSVM, and is obtained by the union of Sub-Space Clustering (SSC) and One Class Support Vector Machine (OCSVM); the first is an evolution of the classical clustering mechanisms, the second is instead an evolution of the supervised Support Vector Machine algorithm particularly useful for use on unlabeled data, where the model of the support vectors is produced from single-class data, corresponding to the traffic normal. The algorithm, tested on the NSL-KDD dataset returns better results than the K-means and DBSCAN approaches, however it has a longer computation time, due to the sequential execution of each subspace, the authors however specify that each of these is independent and therefore can be addressed by parallel execution.

SwissLog [4] was created to act on the basis of two considerations: the first is that logs change frequently, in real application contexts, when dealing with software under development or otherwise with active maintenance; the second is the fact that performance problems in systems are indicative of partial failure problems. SwissLog merge time embedding and semantic embedding approaches to detect sequential log anomalies and performance issues with a unified deep learning model. The anomalies are identified on the basis of different types: the anomalous sequences with respect to the order and the change in the times (performance), understood as the decrease in the execution time of a single task. The operation takes place by means of processing on two phases: offline e online. SwissLog's log analysis methods extract multiple patterns through the stages of tokenization, dictionarization, and clustering of historical log data, which are tracked within the system for the construction of the single sessions, leading, finally, to the transformation into temporal information with semantic content.

An interesting method for anomaly detection is offered by [5], which applies it to logs expressed directly in natural language. The algorithm uses *template2vec* (a novel method) to extract semantic information from the logs. The concept of anomaly partially overlaps with the one intended by SwissLog, since the concept of anomaly in terms of sequence is maintained, i.e., when the sequence of logs differs from the normal order; the concept of change over time, therefore related to the performance of individual tasks, is dropped, and instead quantitative anomaly is introduced, which occurs when the temporal relationship breaks down on a group of logs. The main idea behind LogAnomaly is that log systems produce logs according to patterns, hence semi-structured data; understanding such methods of operation in NLP can aid anomaly detection. LogAnomaly therefore considers that, in a normal workflow, if there are no anomalies, given a certain log it is possible to predict its next one. In the offline training phase, an FT-Tree is used to generate the templates starting from the history and the consequent match of the data with the produced templates takes place; in the online testing phase it is verified if the log in object corresponds to one of the above-mentioned templates, otherwise an approximation of this is looked for to verify if there is a correspondence with the normal flow, if not it is defined anomalous.

3. Ensemble Consensus

There are three basic approaches to detect anomalies [6]. They are based on:

- Density
- Distance
- Isolation

Within the Ensemble Consensus algorithm, three anomaly detection techniques are fused together. Each technique belongs to one of the three fundamental approaches previously mentioned. The algorithm performs the weighted majority voting between the three different techniques on the same data making use of the weighted bagging approach.

At its core there are 3 different anomaly detection algorithms:

- Elliptic Envelope, which exploits the concept of Density. It creates an imaginary elliptical area around a given dataset. Values that fall within the area are considered normal data, and anything outside of that area (distribution density) is returned as an outlier [7].
- Isolation Forest. This algorithm 'isolates' observations by randomly selecting an element (feature) and then randomly selecting a division value between the maximum and minimum values of the selected element. It is an unsupervised algorithm and therefore does not need labels to identify the normal/anomaly. [8] The path distance is averaged and normalized to calculate the outlier score.
- Local Outlier Factor, which exploits the concept of "distance" in a manner similar to K nearest neighbor to identify anomalous tuples. [9]

Judgment of the outlier is made based on the previously calculated score for each technique. At the end, the tuple is classified as outlier only if at least 2 of the three algorithms deem it to be outlier for the "hard consensus" or if the average value of the normalized scores between -1 and 1 is over a soft threshold for the so-called "soft consensus".

The “hard consensus” is just a majority voting anomaly detection. The “soft consensus” technique requires further deepening.

For the so-called “soft consensus”, a normalized [-1,1] outlier (or anomaly) score is built with respect to each technique. The average of the normalized scores is taken. If this averaged score of a sample is lower than a soft threshold, the sample is reported as anomalous, normal otherwise:

$$\mu_i = \frac{1}{3}\sum_{k=1}^{3} \overline{S}_i(T_k) \qquad (1)$$

$$\overline{S} = \frac{S_i - min(S)}{max(S) - min(S)} \qquad (2)$$

In equation (1) μ_i is the average score for the i-th sample. T_k is the technique (Isolation Forest, Local Outlier Factor or Elliptic Envelope). $\overline{S}_i$ from equation (2) is the normalized score of the i-th sample with respect to the T_k technique.

$$A_i = \begin{cases} normal\ if\ \ \mu_i >= \theta \\ anomaly\ otherwise \end{cases} \qquad (3)$$

θ is the soft threshold. This threshold is found via grid search technique and depends on the dataset used.

The scores depend on the technique used. For Elliptic Envelope, the score is the negative Mahalanobis distance among samples as shown in equation (4).

$$D = \sqrt{(S - \widehat{S})^T C^{-1} (S - \widehat{S})} \qquad (4)$$

Where S is the vector of observations (the samples of the dataset), $\widehat{S}$ is the mean of independent variables of the dataset (the mean taken column wise) and C^{-1} is the inverse of the covariance matrix of the independent variables.
Thus

$$S_i = -1 * D_i \qquad (5)$$

For Isolation Forest, the score is defined in equation (6):

$$S_i = 2^{-\frac{E(h(x))}{c(n)}} - (-0.5) \qquad (6)$$

Where $h(x)$ is the length of the path for the sample x, $c(n)$ is the mean of the length of the unsuccessful dichotomic binary tree search where n is the number of external nodes and $E(h(x))$ is the average of $h(x)$ from a list of isolation trees [10]. The offset value -0.5 is the same used in work [10].

For Local Outlier Factor in equation (7) it is important first to define the local reachability density in equation (8)

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \qquad (7)$$

$$lrd_{MinPts}(p) = 1 / \left(\frac{\sum_{o \in N_{MinPts}(p)} reach\text{-}dist_{MinPts}(p, o)}{|N_{MinPts}(p)|} \right) \qquad (8)$$

lrd is the inverse of the mean reachability distance of a point p from its neighbors. In theory the density is inversely proportional to “how far” the neighbors are from the point with respect to a predefined distance metric (i.e., Euclidean). For the used Local Outlier Factor in [9], the offset is set to -1.5 as shown in equation (9).

$$S_i = LOF(x) - (-1.5) \qquad (9)$$

4. Experiment

4.1. Dataset

In this work it has been used the KDD'99 [11] dataset. It is one of the highly used datasets in network security. The dataset was built on the networking information captured in DARPA'98 intrusion detection system program and it is composed by over 4 gigabytes binary tcpdump data over 7 weeks of network traffic, which make up over 5 million records (rows). Each row contains 41 features ending with a label which identifies if the row is normal or an attack and the exact type of attack. The simulated attacks are grouped in four big families namely:

1. Denial of Service (DoS)
2. User to Root (U2R)
3. Remote to Local (R2L)
4. Probing

In this work, this dataset will be used completely in a non-supervised setting showing effectiveness of this solution, even when labels are not present, as normally happens in network traffic analysis.
In this work, all 4 simulated attacks are grouped into one single non-normal class, thus the experiment is performed in a binary classification fashion.

4.2. Experiment

The KDD dataset was parsed, and all non-numerical (string) variables were encoded to categorical variables. The dataset was z-scored and split in 60% for training, 20% validation and 20% test.

The label column was removed in order to transform this dataset in unsupervised learning. A copy of the entire dataset maintains the label column for accuracies comparisons.

The exhausting search called grid search was used to determine the best hyperparameters combination that deliver the highest accuracy. This exhaustive search was performed on the contamination ratio of each internal algorithm and on the soft threshold described previously.

It was found that with a contamination ratio of 0.3 for all the internal algorithms used within the Anomaly Consensus and a soft threshold of -0.2 the accuracy was the highest as reported in Table 1.

For simplicity, results presented in Table 1 were performed by fixing the soft threshold to -0.2.

It was, in addition performed a borderline test using only the 5% of the dataset for training, 30% for validation and the remaining 65% of the entire dataset for testing. Accuracies are reported in Table 1.

With F1 score it is meant the F1 weighted score, which is used to have more realistic accuracies accounting for class imbalance.

Test type	Contamination	Precision	Recall	F1 Score
Normal	0.1	0.91	0.90	0.86
Borderline	0.1	0.72	0.69	0.57

Table 1: Results obtained with normal test and borderline test

Table 2 shows the comparison of the unsupervised Anomaly Consensus algorithm with respect to state of art supervised learning algorithms such as Naive Bayes classifier, Support Vector Machines (with linear kernel), Random Forest classifier, Neural Network and Decision Tree as used by other authors in [12].

Algorithm	Precision	Recall	F1 Score
Naïve Bayes Classifier	0.99	0.85	0.91
SVM	0.98	0.93	0.95
Random Forest	0.99	0.90	0.95
Neural Network	0.99	0.91	0.95
Anomaly Consensus	**0.91**	**0.90**	**0.86**

Table 2: Comparison of the unsupervised Anomaly Consensus algorithm (in bold) with supervised learning algorithms

4.3. Discussion

Experiment results and comparisons in Table 2 suggest the effectiveness of the Anomaly Consensus algorithm comparing its accuracies with state-of-the-art supervised learning algorithms. It is important to state that in some cases, such as Naïve Bayes Classifier, Random forest and neural network, the recall is better or almost on par with others. From Table 1 it is possible to observe that the quality of the result strongly depends on the amount of data used to train the algorithm. This is somehow intuitively explained by the intrinsic properties of each algorithm composing the Anomaly Consensus, who requires a statistically representative number of instances in order to infer a specific pattern and thus recognize anomalies (outliers) effectively. Using only 1000 instances for training and knowing that non-normal (anomaly) data is only a fraction (about 0.001%) of the whole data, achieving a precision of 0.72 and a recall of 0.69 with a final weighted F1-score of 0.57 is encouraging. This suggest that this algorithm can be used effectively also when low quantity of data is present and can leverage the presence of new fresh data in an online learning fashion.

5. Conclusions and future work

The paper proposes an ensemble approach for the anomaly detection task, through an application to the network security domain, having used the KDD99 dataset.

Ensemble Consensus offers the benefit of not being a supervised algorithm, and with that, the fact that it can be used where we have no particular data about attacks on the application network. Therefore, the problem of class imbalance does not arise. The system is immediately applicable, being able to recognize attacks without having information about them.

As widely discussed in the previous sections, the number of training examples is an extremely heavy parameter for generating an accurate classifier. However, the proposed approach demonstrates that the system is applicable even in data-poor contexts, as it possesses not inconsiderable precision and recall parameters with respect to the modest amount of data provided.

The method is particularly suitable for network contexts that have a high temporal variability, as it allows an online application, using as comparison data (training) the data of a certain time slot, considered representative of the normality of traffic. Since it is an unsupervised machine learning algorithm, it can be directly applied without requiring data labeling by an operator, system administrator, or other means.

An effective application of the Ensemble Consensus algorithm also depends on the soft threshold parameter θ, used to determine the threshold below which an example is considered anomalous. In the experiment, the parameter is derived through a grid search process aimed at maximizing performance. In a real-world context, the value can be calibrated to the needs of the system, though it depends in particular on the submitted data stream. A higher value of θ will therefore tend to produce more values deemed anomalous and can be chosen when a system is desired to produce more alerts, so when more control is desired. A lower value of θ instead will produce fewer alerts: on the one hand the advantage of detecting with more certainty anomalies (as they are more evident, regarding the parameter), on the other hand the risk of not recognizing some anomalous data because they are above the threshold.

When it is necessary to work with more complete data, instead, Ensemble Consensus is able to stand comparison with supervised algorithms, providing, against a small decrease of the effectiveness scores, the advantage of working without labelled data. Important result is the recall value, which, not undergoing particular variations with respect to the supervised algorithms, allows the CERT operator an adequate data flow to control.

The main system improvement task for the future is to identify ways to increase recall without penalizing overall precision, as for a critical infrastructure the presence of false negatives could be dangerous. As such, the possibilities of varying the soft threshold θ to achieve higher or lower recall will also be investigated. The difference, in absolute value, between θ and the result obtained from the single example could also represent an estimate of the criticality of the anomaly; this analysis will be deferred to later work.

6. Funding

Research activity co-financed by "Fondo Europeo di Sviluppo Regionale Puglia POR Puglia 2014 - 2020 - Axis I - Specific Objective 1a - Action 1.1 (R&D) - Project Title: Suite products CyberSecurity and SOC" and by BV TECH S.p.A.

7. References

[1] Depren, Ozgur, Topallar, Murat, Anarim, Emin, Ciliz and M. Kemal, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert systems with Applications,* pp. 713-722, 2005.

[2] Jabez, Ja and M. B, "Intrusion detection system (IDS): anomaly detection using outlier detection approach," *Procedia Computer Science,* pp. 338-346, 2015.

[3] Pu, Guo, Wang, Lijuan, Shen, Jun, Dong and Fang, "A hybrid unsupervised clustering-based anomaly detection method," *Tsinghua Science and Technology,* pp. 146-153, 2020.

[4] "SwissLog: Robust and Unified Deep Learning Based Log Anomaly Detection for Diverse Faults," in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020.

[5] Meng, Weibin, Liu, Ying, Zhu, Yichen, Zhang, Shenglin, Pei, Dan, Liu, Yuqing, Chen, Yihao, Zhang, Ruizhi, Tao, Shimin, Sun, Pei and others, "LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs.," in *IJCAI*, 2019.

[6] X. Yu, L. A. Tang and J. Han, "Filtering and Refinement: A Two-Stage Approach for Efficient and Effective Anomaly Detection," in *2009 Ninth IEEE International Conference on Data Mining*, Miami, 2009.

[7] Rousseeuw, J. Peter, Driessen and V. Katrien, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics,* pp. 212-223, 1999.

[8] Liu, F. Tony, Ting, K. Ming, Zhou and Zhi-Hua, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD),* pp. 1-39, 2012.

[9] Breunig, M. M, Kriegel, Hans-Peter, Ng, R. T and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000.

[10] Liu, F. Tony, Ting, K. Ming, Zhou and Zhi-Hua, "Isolation forest," in *2008 eighth ieee international conference on data mining*, 2008.

[11] KDD Cup '99. (n.d.). Retrieved Febraury 2021, from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[12] Sapre, Suchet, Ahmadi, Pouyan, Islam and Khondkar, "A robust comparison of the KDDCup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms," *arXiv preprint arXiv:1912.13204,* 2019.

[13] Impedovo, Donato, and Giuseppe Pirlo. "Automatic signature verification in

the mobile cloud scenario: survey and way ahead." *IEEE Transactions on Emerging Topics in Computing,* 2018.

[14] Pirlo, G., and D. Impedovo. "A new class of monotone functions of the residue number system." *Int. J. Math. Models Methods Appl. Sci 7.9,* 2013: 803-809.

[15] Pirlo, Giuseppe, and Donato Impedovo. "Cosine similarity for analysis and verification of static signatures." *IET biometrics 2.4*, 2013

[16] Impedovo, Donato, and Giuseppe Pirlo. "Updating knowledge in feedback-based multi-classifier systems." 2011 *International Conference on Document Analysis and Recognition*. *IEEE*, 2011.

[17] Pirlo, Giuseppe, Claudia Adamita Trullo, and Donato Impedovo. "A feedback-based multi-classifier system." *10th International Conference on Document Analysis and Recognition. IEEE*, 2009.