Using Artificial Anomalies to Detect Unknown, Known Network Intrusions

Ragini Pant[304573]

Department of computer science, Universitätsplatz 1, Universität Hildesheim, 31141 Hildesheim, Germany
pant@uni-hildesheim.de

Abstract. This paper reviews the work done in the direction of using artificial anomalies to detect unknown and known network intrusions using traditional inductive learning algorithms. Anomalies are events that deviate from the normal behaviour. Injecting artificially generated anomalies in the training data can facilitate to classify between the normal and anomalous data. Distribution-based artificial anomaly generation or DBA2 is the proposed algorithm for generating artificial anomalies. The distribution of a feature's values across the training data is used to selectively generate artificial anomalies.Sparse regions are infrequent values of individual features in data set.Emphasis is made, to amplify the anomalies around the sparse region. This paper focuses on the arena of network based intrusions that is experimented upon pure anomaly detection models and both combined and misuse detection models.

Keywords: Anomaly.· Anomaly Detection Model · Misuse Detection Model · Artificial Anomaly

1 Introduction

Anomaly Detection is the method of identification of the anomalies in the system. It deals with recognizing any pattern that does not adhere to the normal pattern. It is a problem in many areas, like intrusion detection , fraud detection , medical and public health sector. Classically, there are two methods of analysing the data for anomalies :

1. **Classification Systems -** This system uses supervised learning to learn the pattern of known classes then it matches and identifies known labels for unknown data sets. Usually, the training data contains the instances of the known classes, and our aim is to simply detect the instances of these known classes.
   In intrusion detection systems, classification of known attacks is known as misuse detection. These attacks on the system are encoded into patterns, which are then used to match evidence from run-time activities to identify intrusions.

2. **Anomaly Detection Systems-** Categorization is done by the behaviour of the user and the server activities of the system. If they appear normal, then it is placed into the normal profile. The normal profile category, is either a single class or limited number of known instances. This class uses standards from which we can further recognize the significant deviations, or probable intrusions.

The author states that anomaly detection systems are not well-utilized unlike classification systems. Hence the leading commercial intrusion detection systems, apply the misuse detection techniques, which is a part of classification system. But in the real world, the intruder won't necessarily use similar or known patterns. Rather the intruder will always try to devise new techniques to obtrude upon the network. This would require the misuse intrusion systems to be updated frequently across many platforms, which is very expensive .Thus, misuse detection techniques won't be useful when new attacks are launched on the network.

Reviewing the studies, reported in [20] we have learned that there are three phases of a typical attack session. First is the learning phase, where the intruder learns about the target system's limitations, features and vulnerabilities. Second, is a standard attack phase , in which he tries to use standard vulnerabilities and known techniques to intrude the target system. A misuse detection system is useful to detect these known intrusions to the system. The final phase is when the intruder would use an innovative attack technique that is unknown to the intrusion detection model. In this phase, anomaly detection is the only hope to fight against the innovative and stealthy attacks. Thus, it can be concluded that a combined misuse and anomaly detection model is a suitable way of detecting intrusions. The authors intend to explore the use of traditional inductive learning algorithms for anomaly detection and misuse detection, from data set level. This is expected to achieve , by injecting artificial anomalies based on known classes into the model so that the learner can learn the hypothesis to separate the known class from the unknown class. The paper discusses the generation of anomaly detection models from pure normal data, and the generation of combined anomaly and misuse detection models from data that contains known classes. In this paper, we will be detecting known and unknown anomalies for network intrusions, using artificial anomalies.

The paper was published on 19 April 2004 in Springer and has been cited 278 times[21]. Some information about the authors:

1. Wei Fan currently works in Tencent Medical AI Lab. He has a total of 19290 citations [21].
2. Salvatore Stolfo is a Professor of Computer Science in Columbia University. He has a total of 33447 citations[21].
3. Wenke Lee is a Professor of Computer Science at Georgia Tech. He has a total of 36934 citations [21]
4. Philip K. Chan is a graduate student of Florida Institute of Technology. He has 12729 citations [21].

2 Related Work

SRI's IDES [16] measures abnormality of current system activity from the probability distributions of past activities.The activities they monitored are host events (e.g., CPU utilization and file accesses). In the current paper, the authors monitor network events. There are statistical approaches for anomaly detection models . In SRI's IDES [17] and NIDES [18], a user's normal profile consists of a set of statistical measures. In NIDES [18], the authors use a rule-based expert system component for misuse detection. These systems encode known system vulnerabilities and attack scenarios, as well as intuitions about suspicious behavior, into rules. For example, one such rule is: more than three consecutive unsuccessful logins within five minutes is a penetration attempt. Audit data is matched against the rule conditions to determine whether the activities constitute intrusions.Ghosh and Schwartzbard [19] applied neural networks to both anomaly and misuse detection and compared their relative performance.

3 Summary

To begin the task of anomaly detection in networks, the authors started without any examples of the anomalies in the training data. A machine learning algorithm interprets a boundary that can classify different known classes in the training data. The boundary is not specified beyond necessary to achieve generalization and avoid over fitting the model. The learner can also default classify the unknown instances of training data .This default prediction can be modified as anomalous. After experimentation with these methods, the authors did not render any reasonable performance.

Hence, the authors proposed artificial anomaly generation for such a task. In this method, artificial anomalies are injected into the training data so that the machine learning algorithm can learn the hypothesis to differentiate the known class from the unknown class. This helps the learner to discover the boundary around the original data. Artificially generated anomalies are also labelled as anomalies, by the author. The paradigm of generation of artificial anomalies can be defined as 'near-misses'. These are the instances that are close to the training data but are not in the training data.

The authors emphasized that the artificial anomalies are independent of the learning algorithm, since these are just supplemental to the training data.

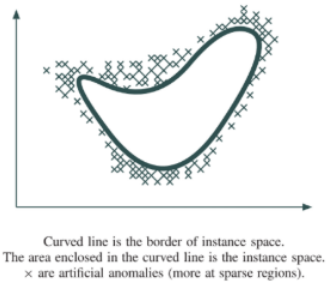


Fig. 1. Artificial Anomalies[5]

3. 1 Distribution based artificial anomaly

Artificial anomalies are generated through near-misses, a useful method to generate them is to randomly change one feature of an example observation, while leaving the other features unaltered.An important factor here is that the training data, used for the DBA2 Algorithm does not contain any anomalies. So, the artificial anomalies generated by changing one feature are always anomalous. To prevent the hypothesis from being overly general, the authors generate data that amplify the sparse regions. For amplifying the sparse regions, the authors proportionally generate more artificial anomalies around the sparse regions depending on their sparsity using the DBA2 Generation Algorithm, proposed by the authors.

Algorithm 1 Distribution Based Artificial Anomaly Generation Algorithm

Input: D , i.e the input data set
Output: D' , i.e the output data set
1: **for all** $v \in V(f)$ **do**
  1: **function** LOOP($i : countV < i \leq countV(max)$)
  1: To create d' replace $v(f)$with randomly chosen value v' s.t $v' \neq v \cap v' \neq v(f)$
  1: $D' \leftarrow D \cap d'$
  1: **return** D'

Working of the DBA2 Generation Algorithm :

1. Assume v to be the sparse values present in the data set.
2. Assume F to be the set of all the features of D.
3. Assume $V(f)$ to be the set of unique values of some features.
4. Assume $countV$ to be the number of occurrences of v in D and $countV(max)$ be the number of occurrences of the most frequently occurring value in $V(f)$.
5. d' is a randomly chosen data point of the data set D
6. Let the output D' be an empty set.

7. For a given feature, we calculate the difference between the number of occurrences of sparse values *(countV)* and the number of occurrences of the most frequently occurring values *countV(max)*.
8. Randomly sample *countV(max)* - *countV* data points from the training set.
9. To generate a near miss anomaly in *d'* ,for each data point d in this sample replace the value of the feature *v(f)* with any *v'*, such that *v' !=* $v \cap v'! = v(f)$
10. The learning algorithm used will then specifically cover all instances of the data with value v for feature f . This anomaly generation process is called distribution-based artificial anomaly generation or DBA2, as the distribution of a feature's values across the training data is used to selectively generate artificial anomalies.
11. The algorithm can be modified to take a factor n, and produce n * $|D|$ artificial anomalies.

3.2 Filtered Artificial Anomalies

An important assumption made by the authors is that the artificial anomalies do not intersect with the known anomalies. Since, checking for overlap between the known anomaly and the artificial anomaly is an expensive process. The authors suggest another approach that requires the learner to filter the artificial anomalies with the hypothesis learnt on the original data, interpreting the boundary between known classes and anomalies. Further, evaluating with the previously generated artificial anomalies and removing the anomalies classified as the known classes, this process is repeated until a stable size of artificial anomalies is achieved.

4 Experimental Setup

4.1 Data Set

For the experiments the authors used the data distributed by the 1998 DARPA Intrusion Detection Evaluation Program, which was conducted by MIT Lincoln Lab (available from the UCI KDD repository as the 1999 KDD Cup Dataset).The DARPA data was gathered from a simulated military network and includes a wide variety of intrusions injected into the network over a period of 7 weeks.The similar taxonomy for categorization of intrusions as was used by the DARPA evaluation, was used in this paper. This taxonomy places intrusions into one of four categories:

1. Denial of Service(DOS) , for example, ping-of-death, teardrop, smurf, syn flood, etc.,
2. Probing(PRB), for example, port-scan, ping-sweep, etc
3. Remotely gaining illegal remote access to a local account or service (R2L),for example, guessing password
4. Local user gaining illegal root access (U2R), for example, various of buffer overflow attacks

U2R	R2L	DOS	PRB
buffer_overflow	ftp_write	back	ipsweep
loadmodule	guess_passwd	land	nmap
multihop	imap	neptune	portsweep
perl	phf	pod	satan
rootkit	spy	smurf	
	warezclient	teardrop	
	warezmaster		

Fig. 2. Intrusions, Categories and Sampling[5]

4.2 Data Processing

The data was then processed into connection records which consists of a number of basic features of the data using MADAM ID[13]. This is a data mining framework used for constructing intrusion detection models. It's key idea is to apply data mining programs to audit data to compute misuse and anomaly detection models that accurately capture the patterns of intrusions and normal activities. Using MADAM ID, the inductively learned classification rules replace the manually encoded intrusion patterns. System features in the detection models and statistical measures in normal profiles are automatically constructed using the frequent patterns, i.e., association rules and frequent episodes, computed from the audit data. Meta-learning is used to learn the correlation of intrusion evidence from multiple detection models, and produce combined detection models [13].

For experimentation, a 10% sample was taken which maintained the same distribution of intrusions and normal connections as the original data (this sample is available as kdd cup data.10% from the UCI KDD repository). In this experiment, 80% of this sample is used as training data and left the remaining 20% unaltered to be used as test data for evaluation of learned models.

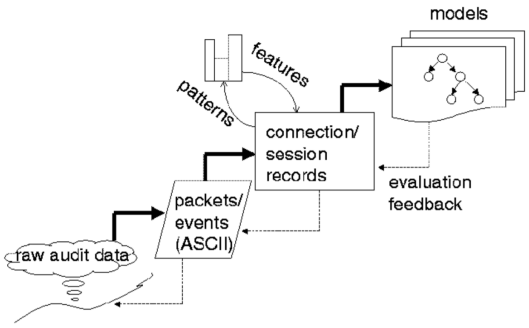


Fig. 3. The Data Mining Process of Building Intrusion Detection Models[13]

4.3 Learning Algorithm

Evaluation of the model is done using RIPPER Algorithm [14] , which is a rule learning algorithm. A RIPPER rule is learned in two phases, a growing phase and a pruning phase.

The RIPPER Algorithm, focuses on incremental reduced error pruning along with a separate and conquered rule learning algorithm. Here, the rules are formulated in a greedy manner, one rule at a time. Rule is grown from an empty conjunction, by repeatedly adding a condition that maximizes the FOIL Information Gain Criterion[15] until the rule covers no negative data items in the growing data set. After the growing phase, a rule is immediately pruned, i.e simplified by repeatedly deleting a condition (or a conjunction of conditions) that can lead to a more accurate rule. This stage improves both the generalization accuracy and the simplicity of the rule. To prune a rule, the implementation considers deleting any empty sequence of conditions from the rule and chooses the deletion that maximizes the function.

$$v(Rule, PrunePos, PruneNeg) = p + (N - n)/P + N \tag{1}$$

where P (respectively N) is the total number of examples in *PrunePos(PruneNeg)* and *p(n)* is the number of examples in *PrunePos (PruneNeg)* .This process is repeated until no deletion improves the value of v.

```
procedure IREP(Pos,Neg)
begin
    Ruleset := ∅
    while Pos≠ ∅ do
        /* grow and prune a new rule */
        split (Pos,Neg) into (GrowPos,GrowNeg)
          and (PrunePos,PruneNeg)
        Rule := GrowRule(GrowPos,GrowNeg)
        Rule := PruneRule(Rule,PrunePos,PruneNeg)
        if the error rate of Rule on
          (PrunePos,PruneNeg) exceeds 50% then
            return Ruleset
        else
            add Rule to Ruleset
            remove examples covered by Rule
              from (Pos,Neg)
        endif
    endwhile
    return Ruleset
end
```

Fig. 4. RIPPER Algorithm[14]

RIPPER rule	Meaning
smurf :- count $\geq$ 5, srv_count $\geq$ 5, service = ecr_i.	If the service is icmp echo request, and for the past 2 seconds, the number of connections that have the same destination host as the current one is at least 5, and the number of connections that have the same service as the current one is at least 5, then this is a smurf attack (a DOS attack).
satan :- rerror_% $\geq$ 83%, diff_srv_% $\geq$ 87%.	If for the connections in the past 2 seconds that have the same destination host as the current connection, the percentage of rejected connections is at least 83%, and the percentage of different services is at least 87%, then this is a satan attack (a PROBING attack).

Fig. 5. Example RIPPER Rules for DOS and PROBING attacks [13]

RIPPER rule	Meaning
guess :- failed_logins >= 5.	If number of failed logins is greater than 5, then this telnet connection is "guess", a guessing password attack.
overflow :- hot = 3, compromised = 2, root_shell = 1.	If the number of hot indicators is 3, the number of compromised conditions is 2, and a root shell is obtained, then this telnet connection is a buffer overflow attack.
...	...
normal :- true.	If none of the above, then this connection is "normal".

Fig. 6. Example RIPPER Rules for R2L and U2R Attacks [13]

4.4 Setup for Combined Anomaly and Misuse Detection Model
To evaluate this model, the authors group the intrusions into 13 small clusters. The data set is created by incrementally adding each cluster into the normal data set and re-generating artificial anomalies. This simulates the process of intervention of new intrusions and their incorporation into the training set. A key point to note is that each cluster contains intrusions that require similar features for active detection. A model that is trained to detect intrusions from one cluster may have difficulties detecting intrusions from another cluster. For clusters with intersecting feature sets, the authors desire that a model learned using training instances of intrusions from some cluster may be used to detect intrusions of other clusters as anomalies.

5 Experimental Evaluation

5.1 Pure Anomaly Detection Models
Figure 7 represents the cumulative anomaly detection rate over all intrusions and false alarm rate.

$\%a_{ttl}$	94.26
%far	2.02

Fig. 7. Cumulative Anomaly Detection Rate and False Detection Rate of Pure Anomaly Detection [9]

The model successfully detects 94.26% of the total anomalies in the test data, and has a false alarm rate of 2.02%.

$$\%far = (|A \cap W_{normal}/|A|) * 100\% \quad (2)$$

In specific intrusion classes, the anomaly detection model is capable of detecting most of the anomalies. For intrusions like *buffer overflow, guess password, back ,phf* the model has 100% accuracy rate. In 3 out of 4 categories, the model has more than 50% accuracy rate of the intrusion occurrences of that category. Also, it is noteworthy to know that U2R, R2L and DOS are potentially more harmful than PRB. Though the pure anomaly detection model has a high false alarm rate,the boundaries learned using artificial anomalies can be sharpened using real intrusions.

5.2 Combined Misuse and Anomaly Detection Models

1. **Evaluation of True Class Detection**: Ideally using artificial anomalies should allow the model to classify true anomalies, i.e any anomaly not in the training set , without degrading it's performance to detect known anomalies. From Figure 9, we notice that the true class detection rate,i.e percentage of

	%a		%a
buffer_overflow	100.00√	ftp_write	50√
loadmodule	66.67√	guess_passwd	100.00√
multihop	57.14√	imap	83.33√
perl	-	phf	100.00√
rootkit	10.00	spy	-
		warezclient	64.25√
		warezmaster	80.00√
U2R	47.06√	R2L	66.67√
back	100.00√	ipsweep	-
land	75.00√	nmap	-
neptune	80.52√	portsweep	4.81
pod	9.62	satan	0.32
smurf	99.94√		
teardrop	-		
DOS	94.31√	PRB	1.34

√: significant or %a $\geq$ 50%

Fig. 8. Anomaly Detection Rate and False Detection Rate of Pure Anomaly Detection [9]

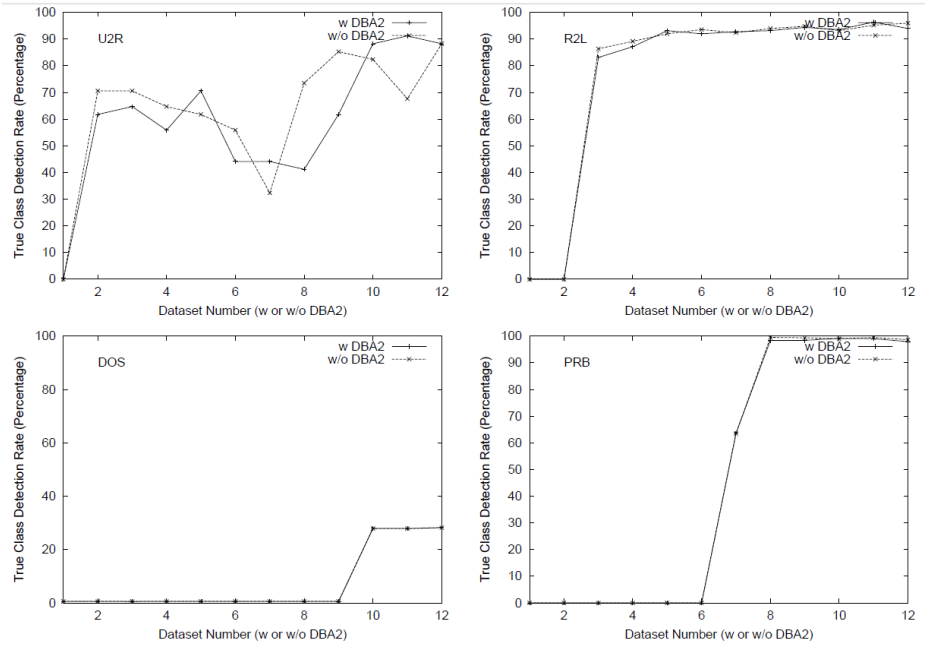


Fig. 9. Comparison of True Class Detection Rate ($\%tc$)of Data sets with and without DBA2[5]

detection of classes (normal or intrusions), with DBA2 and without DBA2 is almost similar for R2L, DOS and PRB intrusions. And for U2R, there is a reasonably small difference.

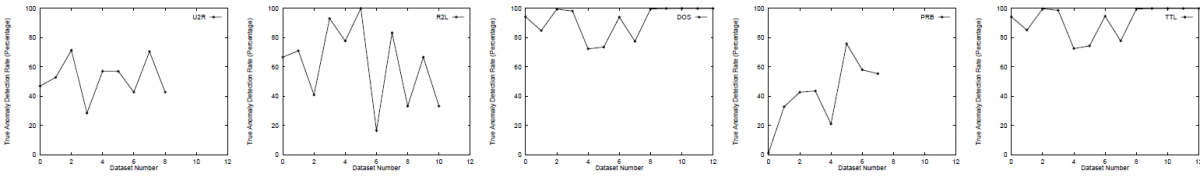
2. **Evaluation of True Anomaly Detection** : Anomaly Detection Model is considered to be stable when the anomaly detection rate is more than 50%. According to Figure 10, for each experiment setting the percentage of significant cases is more than approximately 60%. From Figure 11, we can see

Dataset	0	1	2	3	4	5	6
Anomaly Types	22	21	17	14	13	12	11
Significant	13	15	14	10	8	9	7
%	59	71	82	71	62	75	64

Dataset	7	8	9	10	11	12
Anomaly Types	10	7	6	5	4	2
Significant	8	6	5	5	3	2
%	80	86	83	100	75	100

Fig. 10. Percentage of Significant True Anomaly Detections [5]

that the true anomaly detection rate for all true anomalies remains relatively constant as we inject more clusters of intrusions.The curves for U2R, R2L and PRB categories are more bumpy than DOS because the anomaly detection model catches more in one category and fewer in the others.



Dataset	0	1	2	3	4	5	6	7	8	9	10	11	12
U2R	47.06	52.94	71.43	28.57	57.14	57.14	42.86	70.59	42.86	na	na	na	na
R2L	66.67	71.08	40.96	93.10	77.78	100.00	16.67	83.33	33.33	66.67	33.33	0.00	0.00
DOS	94.31	84.82	99.53	98.22	72.37	73.61	94.08	77.52	99.65	99.96	99.98	99.95	100.00
PRB	1.34	32.89	42.79	43.64	21.15	75.92	58.07	55.44	na	na	na	na	na
TTL	94.26	84.78	99.46	98.18	72.33	73.57	94.02	77.46	99.65	99.96	99.97	99.95	100.00

Fig. 11. Percentage of True Anomalies Detected as Anomalies (%a) [5]

3. **Evaluation of known intrusions detected as anomalies** :
Here,the authors wanted to find if the proposed approach can detect the unclassified known intrusions as anomalies. In the paper, the authors assumed that an anomaly detection model will significantly compensate for misuse detection if either the anomaly detection increases the total rate of detection to nearly 100% or the anomaly detection rate is less than 0.25 of the true class detection rate, when the percentage of true class detection is very low. From Figure 12, we see the total detection rate of anomalies. There is a general trend of increase for the total detection rate of the anomalies. Comparing with Figure 9, there is a significantly higher total detection rate than true class detection rate

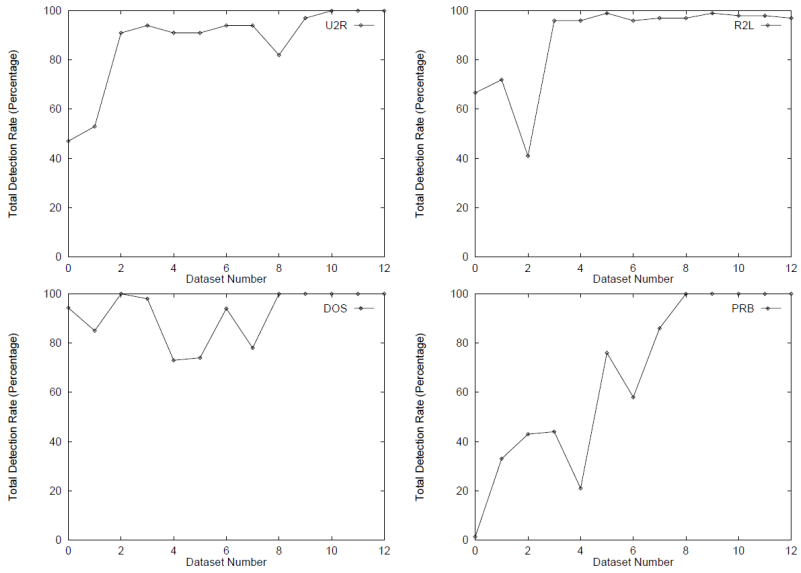


Fig. 12. Total Detection Rate ($\%ttl$) [5]

4. **Overall Performance**
From Figure 13, we can see that the anomaly detection rate decreases as the data is augmented with more clusters of intrusions. This is because the learner has learnt the misuse rules for more intrusions leaving less room for the intrusions to be detected as anomalies. This can be established by the inversely related anomaly detection curve to their true class detection curves as depicted in Figure 9. Thus the overall performance of the model decreases as the intrusions imcrease.

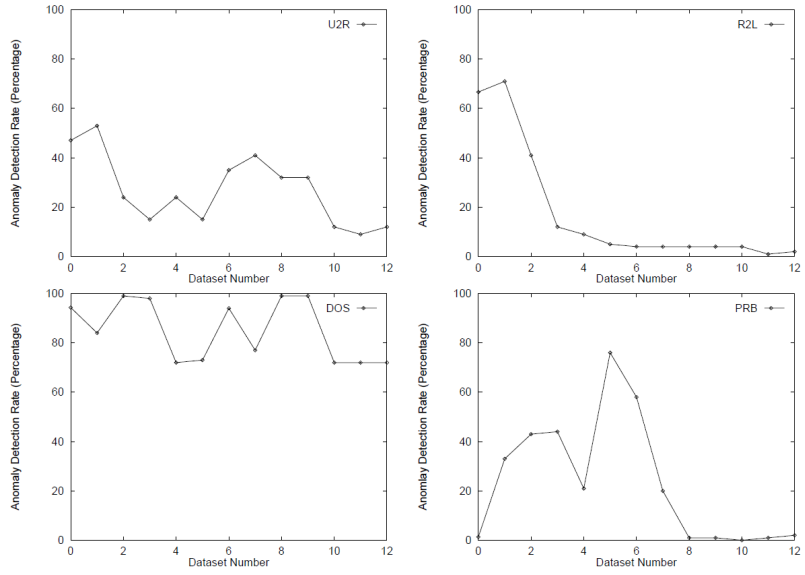


Fig. 13. Percentage of Known Intrusions and True Anomalies Detected as Anomalies ($\%a$) [5]

Effects of Cluster Ordering

The authors also revealed that the results are not influenced by the order in which the clusters are added into the training set. This was done by reversing the cluster ordering, and also by randomly ordering the cluster order.

6 Additional Issues

Experimenting with different amounts of injected anomalies , the authors discovered that when the injection amount is increased the true class detection rate of normal data decreases slightly and the false alarm rate slightly increases. This is based on the ground that, when the percentage of artificial anomalies are more, the learner tends to generate more anomaly rules.

Also, experimenting with other forms of RIPPER that employs ordered rule sets does not have a distinct difference, in contrast to the un-ordered rule sets. Thus, the priority of the rule in an ordered rule set has no effect on the model performance which does not seem to be appropriate.

Using the filtering method for DBA2 as proposed in Section 3.2 did not yield any significant improvement in the performance. Another method that required filtering anomalies using the manual approach was used. But, no artificial anomalies were removed. This concludes that most artificial anomalies are truly anomalous and do not collide with the training data

7 Discussion

It is known that MADAM ID produces classification rule, i.e. RIPPER rules, for the intrusion detection model. This model is helpful since these learned rules have the standard if-else format, with minimum processing and they can be used in several rule-based intrusion detection systems. Another advantage to note, is that RIPPER rules are concise and intuitive, and can be inspected and edited by security experts when needed. There will always be new attacks on the system, thus the intrusion detection systems require simple models for the sake of efficiency. RIPPER Algorithm helps to achieve good generalization accuracy and produces results based on concise and simplified conditions.

An important discussion that was missing from the paper was, how does the learner learn the pattern of Normal and Abnormal Sequences. From my intellect, first artificial anomalies are generated using labeled 'normal' training data. These generated artificial anomalies are labeled as 'abnormal' data. Then, to learn the patterns of the 'normal' and 'abnormal' sequences of data, pre-labeled 'normal' and 'abnormal' sequences are supplied as the training data to the learner.

Reviewing [13] ,we know that a sliding window is used to scan the normal traces. A list of normal traces is called a normal list. Next, scanning is done for the intrusion traces. For each sequence of system calls, it is looked up in the normal list. When it is not found there, the RIPPER is used to predict if the sequence is normal or abnormal. Though it is not necessary that the trace being analyzed by the RIPPER is an intrusion. This depends on the accuracy of the rules, when classifying a sequence as abnormal. Unless the accuracy is close to 100%, it is not likely that a predicted abnormal sequence is a part of intrusion trace, since it can just be an error of RIPPER rules.

A few other approaches for anomaly detection, that were discussed in the seminar were: [1] mines outliers from large datasets based on the distance of a point from it's kth nearest neighbour. This is optimized by using a partition based algorithm that prunes out points whose distances from their nearest neighbours are very small, reducing the computation cost. [2] focuses on detecting intrusions in unlabeled data, using unsupervised anomaly detection techniques where data points are mapped to feature space to better capture the intrusions as outliers. In [3] datasets are partitioned into clusters , and the sparsely populated clusters are compared with the dense clusters using similarity measure to classify the outlier. [4] aims at transforming outlier detection to classification, then by applying classification to a labeled dataset containing artificially generated examples that act as potential outliers. Then a selective sampling mechanism based on active learning is invoked. In [6], anomalies are detected by isolating instances Binary tree structure, called Isolation Trees are constructed to isolate instances where anomalies are generally isolated closer to the root of the tree. This enables a profile of the instance space to be constructed using path length. [7] is another Unsupervised Outlier Removal problem, which utilizes the reconstruction error of autoencoders. Based on this, discriminative information is injected in the learning process of autoencoders to make the inliers and the outliers more separable.In [8] evaluates the quality of Unsupervised Anomaly Detection Model , on

unlabeled data. The evaluation criteria is based on existing Excess - Mass and Mass - Volume curves. In [9] uses unsupervised learning to identify anomalies in large dataset, using AnoGAN which is a deep convolutional generative adversarial network . [10] applies deep learning techniques for anomaly detection in large image dataset, using generative adversarial networks. It applies to searching for good representation of the sample in the latent space.In [11] focuses on the arena of anomaly detection in images. Though, in the proposed approach the model is trained using deep neural networks on out-of-distribution images. This helps in generating features that can identify anomalous images based on soft-max activation statistics, when applied on transformed images.In [12] a semi-supervised technique for anomaly detection is being used which adopts deep-learning techniques based on the idea that the entropy of the latent distribution of normal data should be lower than the entropy of anomalous data.

Comparing with these, the paper presented here injects artificial anomalies to detect known and unknown intrusions. [4] also focuses on a similar method, where artificial outliers are injected into the data so they can act as potential outliers. The performance of outlier detection inherently depends on the exact choice of the sampling distribution of the artificial examples. Active learning effectively alters the sampling distribution to focus on the decision boundary between the normal examples and outliers,and weakens the dependence on this distribution using an ensemble based minimum margin approach. The benefits of this approach are two-fold:

1. Selective sampling based on active learning is able to provide improved accuracy for outlier detection.
2. The use of selective sampling provides the data scalability that we need for typical applications of outlier detection.This helps to handle high dimensional data, with less computational power and memory.

Comparing the proposed approach presented in this paper, to [4] it can be argued that the approach proposed in this paper is not the best method since it will be computationally expensive to convert high dimensional data into connection records that produces classification rules, using MADAM ID. For RIPPER Algorithm to generate rules to classify 'abnormal data', the training data only requires a small percentage of abnormal data as compared to the normal data ,which is not always possible.There might be instances where the data set has no abnormal data, hence the RIPPER Algorithm won't be effective. Also, there is no way of measuring the accuracy of the RIPPER Rule set, so if the rule is incorrect , it is not necessary that it was correctly classified.

An important detail to note, is that direct application of the approach of injecting artificial anomalies can fail to work, since the real world examples may adhere to some hidden constraints that the artificial examples violate, and hence it may be trivial to classify the 'normal' and 'abnormal' groups of examples apart.

Also, there are alternative machine learning algorithms that can be used as learning techniques with the proposed approach of injecting artificial anomalies to detect known and unknown network intrusions, example Hidden Markov Model (HMM's), neural networks to construct intrusion detection model.

8 Conclusion

This paper has suggested to inject artificial anomalies in the training data. It has also emphasized on the requirement of producing artificial anomalies that intensify the sparse data. The flexibility of the approach with other machine learning models is very pivotal, since it can use different learning techniques.

Usage of the RIPPER Algorithm, an inductive rule based learning algorithm also has a significant role in the findings. It provides simple, efficient and intuitive rules.

The findings discovered, after experimentation, correspond with the objective of the paper. The proposed paper points towards many significant avenues that can be pursued for the optimization and improved intrusion detection models.

Evident from the results for the DARPA dataset, the approach presented in this paper produces significantly robust models that are capable of detecting the adverse anomalies.

Bibliography

[1] Ramaswamy, Sridhar, Rajeev Rastogi, and Kyuseok Shim. "Efficient algorithms for mining outliers from large data sets." Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000.

[2] Eskin, Eleazar, et al. "A geometric framework for unsupervised anomaly detection." Applications of data mining in computer security. Springer, Boston, MA, 2002. 77-101."

[3] He, Zengyou, Xiaofei Xu, and Shengchun Deng. "Discovering cluster-based local outliers." Pattern Recognition Letters 24.9-10 (2003): 1641-1650.

[4] Abe, Naoki, Bianca Zadrozny, and John Langford. "Outlier detection by active learning." Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006.

[5] Fan, Wei, et al. "Using artificial anomalies to detect unknown and known network intrusions." Knowledge and Information Systems 6.5 (2004): 507-527.

[6] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation-based anomaly detection." ACM Transactions on Knowledge Discovery from Data (TKDD) 6.1 (2012): 1-39.

[7] Xia, Yan, et al. "Learning discriminative reconstructions for unsupervised outlier removal." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[8] Goix, Nicolas. "How to evaluate the quality of unsupervised anomaly detection algorithms?." arXiv preprint arXiv:1607.01152 (2016).

[9] Schlegl, Thomas, et al. "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery." International conference on information processing in medical imaging. Springer, Cham, 2017..

[10] Deecke, Lucas, et al. "Image anomaly detection with generative adversarial networks." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, 2018.

[11] Golan, Izhak, and Ran El-Yaniv. "Deep anomaly detection using geometric transformations." Advances in Neural Information Processing Systems. 2018.

[12] Ruff, Lukas, et al. "Deep semi-supervised anomaly detection." arXiv preprint arXiv:1906.02694 (2019).

[13] Lee, Wenke. A date mining framework for constructing features and models for intrusion detection systems. Diss. Columbia University, 1999.

[14] Cohen, William W. "Fast effective rule induction." Machine learning proceedings 1995. Morgan Kaufmann, 1995. 115-123.

[15] Quinlan, J. Ross. "Learning logical definitions from relations." Machine learning 5.3 (1990): 239-266.

[16] Javitz, Harold S., and Alfonso Valdes. "The SRI IDES Statistical Anomaly Detector." IEEE Symposium on Security and Privacy. 1991.

[17] Lunt, Teresa F., Ann Tamaru, and F. Gillham. A real-time intrusion-detection expert system (IDES). SRI International. Computer Science Laboratory, 1992.

[18] Jagannathan, Raj, et al. System design document: Next-generation intrusion detection expert system (NIDES). Vol. 7. Technical Report, 1993.

[19] Ghosh, Anup K., and Aaron Schwartzbard. "A Study in Using Neural Networks for Anomaly and Misuse Detection." USENIX security symposium. Vol. 99. 1999.

[20] Jonsson, Erland, and Tomas Olovsson. "A quantitative model of the security intrusion process based on attacker behavior." IEEE Transactions on Software Engineering 23.4 (1997): 235-245.

[21] Citation counts taken from https://scholar.google.com/. Accessed on 24-March-2020.