Intrusion detection system using decision tree-based attribute weighted AODE

Vasudha K. Deshpande

Research Scholar, ME Student, CSE, Government College Of Engineering, Aurangabad, India

Abstract: The number and severity of the network attacks have increased in past few years. So for securing the network from different network attacks, intrusion detection system (IDS) plays a key role. Detection of the intrusive activities by using resource intensive intelligent algorithms has been possible because of advancements in computing performance in terms of processing power and storage. In this paper, an efficient data mining algorithm called Naïve Bayes for anomaly based network intrusion detection has been implemented. However, Naïve Bayes assumes attributes are independent of each other, the same may affect the accuracy of the system. To solve this attribute independence issue and to increase accuracy another data mining algorithm named Averaged One Dependence Estimator i. e. AODE is implemented. And to improve the performance of AODE further another algorithm has been proposed named Decision Tree-based Attribute Weighted AODE (DTWAODE). In DTWAODE Decision Tree is used & weight is assigned to each attribute. The weight is set according to its depth in the decision tree building on the training samples. The training sample used is NSL KDD-99 data set. The performance of Naïve Bayes, AODE & DTWAODE is studied and analyzed on the NSL KDD-99 intrusion benchmark data set and the accuracy is calculated.

Keywords: IDS, NSL, KDD, AODE, DTWAODE.

I. INTRODUCTION

Now a days, Organization are prone to cyber attacks such as network intrusion, one of the root cause for the same is wide spread of high speed internet access. An intrusion detection system (IDS) is a software application or device which keep an eye on system or network for malicious activities for policy violations and act as a reporting mechanism for management unit. Intrusion Detection System detects intrusion both from inside and outside of the network. IDS and firewall has a thin line of difference between them. A firewall prevent intrusions and protects the flow of data where as IDS detects if the network is being attacked or if the security enforced by the firewall has been violated. Firewall and IDS together improves the security of network.

There are two main techniques of intrusion detection: misuse & anomaly detection. Anomaly detection detects unusual activity patters in the observed data. It is based on subject's (e.g. a system or a user) normal behaviour. Any significant deviation from the usual activity is considered as intrusive.

Misuse detection technique recognizes known attack patterns. It is based on signature of known attack. Any action that matches with the pattern of a known attack is considered as intrusive. There are two possibilities:

1. False positive: Anomalous activities which are non intrusive but are marked as intrusive.
2. False Negative: Anomalous activities which are intrusive but are marked as non intrusive.

For developing an Intrusion detection system data mining can be used. In misuse detection, every instance in a data set is tagged as ‘intrusion’ or ‘normal’ and a learning algorithm is trained over the tagged data. These techniques are capable of automatically retraining intrusion detection models on different input data that include new types of attacks, provided they have been tagged correctly. In Anomaly detection normal behavior models are built and any deviation from it is tagged as intrusive.

As a part of this paper author has implemented Naïve Bayes & AODE based intrusion detection system for estimating probabilities of observed network traffic to be normal or anomalous. And proposed DTWAODE based intrusion detection system to classify the network traffic to be normal or anomalous. The simulations are performed on NSL KDD-99 data set. This paper can be organized as follows: Section II provides literature review of other available data mining techniques for intrusion detection, Section III covers problem formation, in Section IV proposed algorithm is discussed. In section V, author provides implementation results. The final Section VI, concludes the work & future scope.

II. LITARATURE SURVEY

One of the commonly used neural network classification algorithm is Multilayer perceptron (MLP). During simulations with KDD dataset in MLP three layer feed forward neural network architecture is used. The three layer feed forward network consist of one input, one hidden & one output layer. For each neuron in both the hidden & output layer unipolar sigmoid transfer functions were used with slope value of 1.0. The learning algorithm used was stochastic gradient descent with mean squared error function. There were total of 41 neurons in the input layer (41-feature input pattern), and 5 neurons (one for each class) in the output layer.. MLP neural networks are trained by alteration of weights that are assigned to the interconnections between the neural network nodes. This can be achieved by using different functions during the training period for the algorithm, such as gradient-based optimization algorithm. When data is fed into the input layer, the output layer of the network shows the expected result provided the network should converges to the local minima of error.[1]

Gaussian Classifier (GAU) assumes inputs are uncorrelated and distributions for different classes differ only in mean values. Gaussian classifier is based on the Bayes decision theorem. Four distinct models were developed using the Gaussian classifier: quadratic classifier with diagonal covariance matrix, quadratic classifier with tilted covariance matrix, linear classifier with diagonal covariance matrix, and linear classifier with tilted covariance matrix. Linear discriminant classifier with full tilted matrix performed the best on the KDD testing dataset with cost per example value of 0.3622. [1]

Support vector machine(SVM) is a machine learning algorithm. It is used for regression as well as for classification. For classification of data some standard SVMs are used as powerful tool. In SVM data is classified in two-category points. In which data is assigned to one of two disjoint half spaces either in original input space or in higher dimensional feature space for nonlinear classifier.[2]

In this paper, we apply the efficient data mining algorithms called Naïve Bayes & AODE for anomaly based network intrusion detection. And another algorithm is proposed named Decision Tree based Attributed Weighted AODE (DTWAODE).

III. PROBLEM FORMATION

Naïve Bayes is the statistical inference learning algorithm which provides spam detection, document classification and intrusion detection. This technique performs better in terms of cost, computational time and false positive rate, when applied to NSL KDD'99 data sets as compared to a back propagation approach, a neural network based approach. Naïve Bayes algorithm considers all attributes are independent of each other. But there are certain attributes which are dependent on other. This attribute independent assumption may affect the accuracy of the system. This problem is resolved in AODE i.e. Averaged One Dependence Estimator algorithm. Thus results obtained from AODE are more accurate than Naïve Bayes. AODE is a technique capable of accurately estimating the probabilities of observed network traffic to belong to a given class i.e. either normal or anomalous, with a higher degree of accuracy. AODE also solves the problem of large computational overhead, which was evident in the performance of its predecessor classifiers. To improve the accuracy further another algorithm is proposed named Decision Tree-based Attribute Weighted AODE. The DTWAODE is an enhanced AODE algorithm by assigning each attribute different weight. The weight assignment is done according to the depth of the attribute in the tree. The NSL KDD-99 data set is used for simulation in these algorithms.

IV. PROPOSED IDS

In this section AODE & DTWAODE implementation on NSL KDD-99 data set has been discussed.

A. An Average One Dependence Estimators

The attribute independence issue associated with Naïve Bayes has been resolved through the development of the Average One Dependence Estimator (AODE) algorithm. Averaged one-dependence estimators is a semi Naïve Baysian Learning method. AODE performs classification of data by aggregating the predictions of multiple one-dependence classifiers. In multiple one dependence classifier all attributes depend on the same single parent attribute & the class. Instead of predicting single class, AODE predicts class probabilities. Missing data situations are handled by its probabilistic model.

Attribute independence issue associated with Naïve Bayes can be resolved by allowing an attribute to depend on other non class attribute. Efficiency can be maintained by utilizing one-dependence classifier, such as Tree Augmented Naïve Bayes (TAN). In TAN each attribute depends on the class & at most one other attribute. However, the approaches based on one-dependence classifier performs model selection, which is a process which usually gives substantial computational overheads & significantly increases variance related with Naïve Bayes.

B. Decision Tree-based Attribute Weighted AODE

An improvement to AODE is Decision Tree-based attribute weighted AODE. In DTWAODE different weights are assigned to each attribute as per the attribute depth in the tree. A special tree augmented naïve bayes is built for each attribute. The special tree consists of root attribute & other attributes. The root attribute is the parent of all other attribute. The average of aggregate of these tree augmented Naïve Bayes is used for prediction by AODE. In AODE each attribute is treated equally, but attribute may play different role in classification, in many real applications. Thus, DTWAODE is proposed, which assigns different weight to each attribute. With this new algorithm the classification accuracy of AODE is increased & retaining its low time complexity & simplicity.

C. Implementation

The flowchart for the intrusion detection system can be implemented as follows:

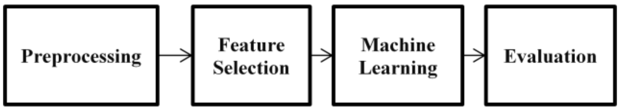

Figure 1. IDS implementation flowchart

1. Pre processing

Conversion of all the features to format that is intelligible to the machine learning algorithm is essential. In NSL KDD dataset protocol type, service & flag takes nominal values whereas all other features take numeric value. In preprocessing step all the features are converted into nominal format so that naïve Bayes, AODE & DTWAODE can operate on these values unaffected.

2. Feature Selection

Feature selection is the process which is commonly used in machine learning. In feature selection process subset of features available from the data is selected for the learning algorithm. Since computationally it is not feasible to use all available features also there are problems of estimation for the limited data samples (but a large number of features), feature selection is required. The features of the data set are identified as either being significant to the intrusion detection process or redundant. The redundant features are closely related with one or more other features. So omitting these redundant features from the data set will increase the accuracy of the system. The feature selection is used in naïve bayes & AODE. For DTWAODE, feature selection is not necessary. As in DTWAODE decision tree is used for classification.

As a part of feature selection for Naïve Bayes & AODE following three techniques are used:

2.1 Group Method for Data Handlings

Following steps are used to rank the features:

- Select three inputs at a time by changing the setting of model synthesis. This method is used to restrict the number of selected features to three which effectively helps in ranking the features in groups of maximum size three.
- The model can select features from the remaining less predictive features by removing selected features.
- Repeat the process until all features are selected or no features can be selected any more.

After ranking the features, a subset is selected by using the top ranked features one at a time, for as long as the accuracy of the selected model keeps increasing. The model is considered to be over fitted when the accuracy of the model starts to decrease. At this point the feature selection process is stopped.

2.2. Information Gain

In the information gain most prominent feature subset is created by filtering the features before the start of the learning process. Ranking of the attribute can be done individually on the basis of separation of classes of the training data elements that is individual rows of the data set. The attribute ranks with respect to the class are calculated using the following formula:

Information gain = (*dx*) - (*d¡x*)

Where dx is the information which includes attribute x, and d¡x is information which excludes attribute x.
The value of d¡x is calculated as the average of each value that this particular attribute can take.

2.3. Gain Ratio

Gain ratio is modification of information gain. It solves the issue of bias towards features with a larger set of values. Because of this bias the accuracy of learning algorithm degrades. When the attribute values are evenly spread gain ratio is high whereas when a attribute has single value gain ratio is low.

3. Machine Learning

3.1 Averaged one dependence Estimator (AODE)

Subsequent to filtering and selection of the highest ranked features for the intrusion detection process, the reduced data set is used for training and evaluating the machine learning scheme. Training is performed on a subset of the data. The performance of the resulting model is verified on the remaining parts of the data set. Naïve Bayes is a supervised learning algorithm that relies on probabilistic models and apriori knowledge for classifying data. It uses statistical inferences, with an assumption of attribute independence, where an attribute of a given data sample is a property of the sample with a given value.

The AODE algorithm may give more accurate classification than Naïve Bayes on data sets with non-independent attributes. The training phase of the AODE algorithm operates by iterating through a given data set with k features, and generating a set of frequency vectors as follows:

Let's consider:
1.data1[z] - number of data elements belonging to a given class z
2.data2[n] - number of times a given data element is found to possess a value, iterated over all n features
3.data3[wi] - number of times the value wi is encountered in the entire data set
4.freq[z;wi;wj] - the frequency of simultaneous occurrence of two attribute values wi and zj for a given class z

```
Algorithm:
Inputs: training set W*, Z *
number of attributes n, and
number of classes m
Outputs: joint frequency vector freq,
class frequency vector data1,
attribute frequency vector data2,
attribute-value frequency vector data3, and
item count counter
Initialize frequencies
counter = 0
Initialize all elements of freq, data1, data2, and data3 to 0
Accumulate frequencies
FOR EACH w; z ε W*; Z*
    count = count + 1
    data1[z] = data1[z] + 1
    FOR i = 1 TO n
        IF wi is known
            data2[i] = data2[i] + 1
            data3[wi] = data3[wi] + 1
            FOR j = 1 TO n
                IF wj is known
                    freq[z; wi; wj ] = freq[z; wi; wj ] + 1
                END IF
            END FOR
        END IF
    END FOR
END FOR
```

During the testing phase of the AODE algorithm, the data elements or instances of the data set are introduced to the algorithm by hiding the class to which they belong. The task of the AODE classifier is to predict the probability of the data element to belong to each of the given classes. The higher probability is then used for deciding the class of the data element.

3.2 Decision Tree-based Averaged one dependence estimator (DTWAODE)

The pre processing of the data is required in DTWAODE. But instead of feature selection, decision tree is used in DTWAODE. DTWAODE consists of two learning parts DTWAODE-learning & DTWAODE-test. At training time, the weight for the attributes are eagerly learned to fit the training data. The weight is calculated using the formula: $1/\sqrt{d}$, where d is the minimum depth at which the attribute is tested in the tree. If the attribute does not appear in the tree, the weight is set to zero. At classification time, the weighted AODE classifier has been built for each given test instance. In DTWAODE the projected weights are stabilized by building multiple decision trees using bagging & then average the weights across the ensemble. In this algorithm the number of bagging iterations m is set to 10 and the percentage of the training data to use for learning a tree in each iteration n is set to 50.

The algorithm is represented as follows:

```
Algorithm  DTWAODE (Q, z, w, g)
Input: training instances Q, a test instance z, the number of bagging iterations w, the percentage of the training data to learn the tree g.

Output: the target class label t of z.
1. Repeat w times
2. Sample randomly g% of the training instances Q
3. Analyze an unpruned decision tree from the instances which are resampled
4. For each attribute in the training instances Q
5. If the attribute that do appear in the decision tree
6. Find out the minimum depth of the attribute d
7. Set the weight equal to 1/√d
8. Else
9. Set the weight = 0
10. For each attribute in the training instances Q
11. Set the final weight equal to the average of the w weights
12. Remove the instance with all attributes zero weight
13. Train a DTWAODE model to produce the class label t of z
14. Return the class label t of z
```

V. PERFORMANCE ANALASIS

In this section we analyze the results of simulations performed on given data set for all three algorithms: Intrusion Detection using Naïve Bayes, Intrusion Detection using AODE & Intrusion Detection using DTWAODE. The results were quantified based on the following metrics, commonly used for evaluating intelligent classifier:

1. Accuracy= TP+TN/TP+TN+FN+FP
2. Precision= TP/TP+FP
3. Recall= TP/TP+FN
4. TP Rate= TP/TP+FN
5. FP Rate= FP/FP+TN

where ,
TP is the number of actual positives classified correctly as true.
FP is the number of negatives in the data set classified incorrectly as positives.
TN is the number of negatives classified correctly as negatives, and
FN is the number of positives classified incorrectly as negatives.

Table 1 gives the comparative performance of AODE, Naïve Bayes & DTWAODE algorithms. The table lists the attack type data, Count in the original NSL KDD Data and Count, TP Rate, FP rate by Naïve Bayes, AODE & DTWAODE.

For all simulations, we trained and tested the performance of the classifiers on the KDD-99 dataset. For the Naïve Bayes accuracy observed is 93.51% & for AODE accuracy observed is 97.38%. And the accuracy for DTWAODE observed is 98.73%. Thus the proposed method gives the more accuracy as compared to AODE method. Hence DTWAODE is an improvement of AODE algorithm.

TABLE I

Attack Type	**Original NSL KDD Data**	**AODE**			**Naïve Bayes**			**DTW AODE**		
		Count#	**TP Rate**	**FP Rate**	**Count#**	**TP Rate**	**FP Rate**	**Count#**	**TP Rate**	**FP Rate**
Neptune	4657	4506	0.968	0.017	4506	0.968	0.017	4532	0.973	0.016
Normal	9711	9248	0.952	0.079	8284	0.853	0.021	9320	0.96	0.08
Saint	319	0	0	0	0	0	0	7	0.022	0
Msscan	996	846	0.849	0.007	846	0.849	0.007	846	0.849	0.006
Guess_passwd	1231	1099	0.893	0.003	1099	0.893	0.003	1193	0.969	0.003
Smurf	665	665	1	0.003	665	1	0.005	665	1	0.003
Apache2	737	733	0.995	0.002	733	0.995	0.002	733	0.995	0.002
Satan	735	467	0.635	0.014	466	0.634	0.015	465	0.633	0.014
Buffer_overflow	20	0	0	0	13	0.65	0.003	2	0.1	0
Back	359	356	0.992	0	356	0.992	0	354	0.986	0
Warezmaster	944	722	0.765	0.009	722	0.765	0.009	719	0.762	0.003
Snmpgetattack	178	0	0	0	1	0.006	0	0	0	0
Processtable	685	685	1	0.005	685	1	0.005	683	0.997	0.005
Pod	41	5	0.122	0	5	0.122	0	5	0.122	0
Httptunnel	133	1	0.008	0	1	0.008	0	0	0	0
Nmap	73	73	1	0	73	1	0	73	1	0
Ps	15	0	0	0	0	0	0	3	0.2	0
Snmguess	331	0	0	0	328	0.991	0.055	0	0	0
Ipsweep	141	139	0.986	0.005	139	0.986	0.005	139	0.986	0.005
Mailbomb	293	293	1	0.01	293	1	0.011	293	1	0.01
Portsweet	157	90	0.573	0.002	90	0.573	0.002	80	0.51	0
Multihop	18	0	0	0	0	0	0	0	0	0
Named	17	0	0	0	0	0	0	4	0.235	0
Sendmail	14	0	0	0	0	0	0	0	0	0
Loadmodule	2	0	0	0	0	0	0	0	0	0
Xterm	13	0	0	0	0	0	0	0	0	0
Worm	2	0	0	0	0	0	0	0	0	0
Teardrop	12	0	0	0	0	0	0	0	0	0
Rootkit	13	0	0	0	0	0	0	0	0	0
Xlock	9	5	0.556	0	5	0.556	0	5	0.556	0
Perl	2	0	0	0	0	0	0	0	0	0
Land	7	0	0	0	0	0	0	0	0	0
Xsnoop	4	0	0	0	1	0.25	0	0	0	0
Sqlattack	2	0	0	0	0	0	0	0	0	0
FTP write	3	0	0	0	0	0	0	0	0	0
Imap	1	0	0	0	0	0	0	0	0	0
UDPstorm	2	0	0	0	0	0	0	0	0	0
PHF	2	0	0	0	0	0	0	0	0	0

VI. CONCLUSION

In this paper, we have implemented Naïve Bayes & Averaged One Dependence Estimator based intrusion detection systems on NSL-KDD data set. These two schemes classifies the network traffic based on attributes of the traffic. For the NSL-KDD data set, our implemented intrusion detection scheme i.e. DTWAODE outperforms Naïve Bayes & AODE in terms of accuracy in attack detection, lowered false alarm rates, and improved precision and recall values, as observable from the simulation results. As Naïve Bayes assumes the attributes are independent of each other which lead to accuracy of 93.51%. This problem is resolved by AODE giving accuracy of 97.38% irrespective of input. And our approach Decision Tree-based Attribute Weighted AODE (DTWAODE) has been implemented to improve the accuracy further by using decision tree. In DTWAODE, the decision tree is built by assigning weights to the attributes to classify the network traffic as normal or anomalous. With the DTWAODE the accuracy observed is 98.73% which higher as compare to Naïve Bayes & AODE.

As a part of future work we can test the performance of these algorithms on different types of data set. Also to improve the performance of DTWAODE we may develop more efficient method for assigning weights to the attribute.

ACKNOWLEDGMENT

The authors wish to acknowledge sincere thanks to the principal Govt. College of Engineering & Prof. V.P Kshirsagar, Head of Department, Computer Science & Engineering, Government College Of Engineering, Aurangabad.

My special thanks to Prof.S.G.Shikalpure, my project guide, for his sincere efforts & for his kind guidance specially in finalizing this project topic. He took deep interest in checking the minute details of the report and guides me throughout the project.

REFERENCES

[1] Maheshkumar Sabhnani & Gursel Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context".

[2] V. Jaiganesh, S. Mangayarkarasi, Dr. P. Sumathi, "Intrusion Detection Systems: A Survey and Analysis of Classification Techniques", IJARCCE Vol. 2, Issue 4, April 2013

[3] Abhinav Jain, Sanjay Sharma, Mahendra Singh Sisodia, "Network Intrusion Detection by using Supervised & Unsupervised Machine Learning Techniques: A Survey", International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 1, Issue 3.

[4] Zubair A. Baig, Abdulrhman S. Shaheen, and Radwan AbdelAal, "An AODE-based Intrusion Detection System for Computer Networks",IEEE 2011.

[5] Mrutyunjaya Panda & Manas Ranjan Patra, "Network Intrusion Detection using Naïve Bayes", IJCSNS International Journal of Computer Science and Network Security,VOL.7,No.12,December 2007.

[6] Dewan Md. Farid, Nouria Harbi & Mohammad Zahidur Rahman, "Combining Naïve Bayes & Decision Tree For Adaptive Intrusion Detection", International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010.

[7] Nahla Ben Amor, Salem Benferhat Zied Elouedi, "Naïve Bayesian Networks in Intrusion Detection Systems".

[8] Geoffrey I. Webb, Janice Boughton & Zhihai Wang, "Averaged One Dependence Estimators: Preliminary Results".

[9] Poonam Dabas & Rashmi Chaudhary, "Survey of Network Intrusion Detection Using K-Mean Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013.

[10] Jia Wu, Zhihua Cai , "Learning Averaged One Dependence Estimators by Attribute Weighting" Journal of Information & Computational Science 8:7(2011).

BIOGRAPHY

Vasudha K. Deshpande received the B.E. in Computer Science and Engineering and currently pursuing the ME in Computer Science and Engineering. I was working as Assistant Professor in CSE department at MIT, Aurangabad.