A Prototype Multiview Approach for Reduction of False alarm rate in Network Intrusion Detection System

Premansu Sekhara Rath, Nalini Kanta Barpanda, R P Singh

[1]*Ph.D Scholar in SSSUTMS, Sehore,India*
[2]*Asst Professor, SUIT, SAMBALPUR.*
[3,4]*SSSUTMS, Sehore,*

Abstract
Every now and then we are very much related to the network. It may be internet or intranet. We generally share personal information as well as organizational information through the network. So we should secure our network. Since last twenty years various NIDS have been developed and widely used in the network which detects efficiently the various network threats. One of the contexts of NIDS is generation of alarms when an attack is detected. But sometimes the NIDS produces false alarms. Many machine learning approaches have been applied to reduce false alarm rate, but the approaches are not multi-viewed based approach. Those approaches use single function to model a particular view and then optimize all the functions in the learning process. But here, we develop MVPSys, a practical approach to reduce false alarm which works efficiently. Here a semi-supervised learning approach is implemented on both labeled and unlabeled data. This system clearly analyzed both destination feature data set and source data set. After so many experiments, we are able to achieve 97% reduction of false alarm rate which significantly improves the efficiency.

Keywords:
NIDS, MVPSys, False alarm rate, Accuracy, WEKA, Snort, DARPA

1. Introduction

Now a day, we generally find a rapid growth in computer network application. Hence we also get network intrusions like worms, spamware; Trojan, deniel of service etc are the major threats. They can cause big losses in data. To avoid this NIDS is designed within the network to protect from these attacks. From our theory, based on detection methods NIDS can be classified into two categories:

- Signature based NIDS
- Analog based NIDS

In signature based NIDS, a rule-based description for known attacks is installed. Then incoming traffic behaviors are compared with the signature. If a match is found then a alarm is produced. But in anomaly based approach, a predefined normal profile determines a threshold value. If the deviation exceeds a threshold value then an alarm is generated. In both the system, we found major drawbacks which are generation of lot of false alarms. This alarm is called false alarm because it produces alarm for normal event as an intrusion. This case is called false positive alarm. In heavy traffic system, the number of false positive alarms is more. From Pietraszek (2004) proposed system, we found 99% of generated false alarms are type of false positives. Hence, we can say reducing rate of false positive alarm is the key factor for improving the efficiency of NIDS.

Manuscript received April 5, 2025
Manuscript revised April 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.4.19

2. Literature Survey:

From literature survey, we found number learning approach. In 2004, Law and Kwok proposed a NIDS which reduced false positives. In 2005, Alhaby and Imai developed a NIDS which reduces rate of false positive but not so much extent. In 2015, Wenjun Li, Luo and kwon proposed a type of MVPSys in NIDS which reduced nearly 90% of false positive alarm rate.

These efforts define that an appropriate alarm filter can be constructed to improve the efficiency of NIDS. In 2013, Sun defined a multiview approach for the first time where each view represents a set of features. In machine learning area like SVM, kernel machine etc all multiple views are concatenated into a single view to adopt to the learning settings. It may cause over fitting problems. In contrast to the single view learning, many studies show that multiview learning can improve and optimize learning process (Xu 2013; sun 2013). From above studies, we conclude that a multiview learning should be given more importance and attention in NIDS.

Apart from this, we are motivated because of its practical implementation. Most existing studies did not implement their algorithms into a practical alarm reduction system. Our research work attempts to develop a practical alarm reduction system in the context of multiview learning approach.

3. Contribution:

The contribution in our work can be described as follows: first we develop a two view based false reduction system which extracts two feature sets from an incoming NIDS alarm; i)Target feature set ii)Origin feature set. Then we implemented in a practical based prototype system called MVPSys. One can reduce false alarm in both offline

mode and real time mode. Further, we evaluated the performance of NIDS on snort false alarm reduction system (Rosech 1994) with two data sets and in two real network environment. By taking DARPA dataset our system can able to get accuracy of 96.2% wide while other best similar algorithm can only get 91.2 %(LI, Meng, 2015).

4. Theory:

In this section, we presented the theory of NIDS alarm and the challenges in it.

4.1 *What is NIDS Alarm?*

From study of both signature based NIDS and anomaly based NIDS, we found four different types of alarms are generated in four different cases which are given below.

1. True Positive: Where an intrusion is found and alarm is generated.
2. False Positive: Where there is no intrusion but alarm is generated.
3. True Negative: Where intrusion is not found and no alarm is generated.
4. False negative: where an intrusion is found but no alarm is generated.

From above we can say that an alarm in IDS is false if there is either false negative and false positive. We can reduce false negative by improving the method of detection that is improving the contexts and the number of rules for signature based and creating more accurate normal profiles for anomaly based detection method. But in this paper we mainly focus on false positives.

4.2 Challenges in false alarm reduction:

Due to more number of alarms, it is very difficult for manual analysis and classification. The following are some of challenges for analysts regarding false alarm reduction in NIDS.

- Analyst must be an expert in the field of intrusion detection system.
- For an expert it is very difficult to write general rules to characterize the whole set of these alarms. Hence like MVPSys and expert knowledge combinely form the rules.
- Due to dynamic environment, the rule based database has to be updated.
- To cope with the environment, analysts have to spend a lot of time and effort for classifying alarm and reducing false alarms. Hence, it needs expensive operations.

For solving above challenges, a machine learning based technique has to be used. Hence, we proposed a MVPSys which is used to reduce false alarms and also reduces the burden of analysis.

4.3 Work objective:

In this research,

- We have developed a multi-view based false positive reduction system which refines NIDS false alarms.
- We have implemented a semi-supervised learning algorithm.
- Our system has achieved a better output in case of unlabelled data.

4.4 Proposed System:

So far we have seen the traditional machine learning algorithms like KNNs, SVM, NN etc are applied single view based data. But it creates over fitting problem. But to make more intelligent learning system, a multi-view approach has to be adopted. In multi-view, one function is used to model a particular view and then jointly optimizes all functions to exploit two or more views of the same input that improves the learning performance. Hence, we emphasize more on multi-view approach for finding false alarms. Let's define different terms used for our proposed system.

4.5 What is multi-view learning?

In contrast to single-view learning, multi-view learning introduces one function to model a particular view and jointly optimizes all the functions to exploit the redundant views of the same input data and improve the learning performance. In a machine learning system when data is represented by multiple distinct feature sets we can say it is multi-view learning. For example, a web page can be viewed as document that itself and anchor text which is attached hyperlinks pointing to this page.

According to Xu et al. (2013), multi-view learning can be classified into three groups: 1) co-training, 2) multiple kernel learning, and 3) subspace learning In particular, co-training algorithms train alternately to maximize the mutual agreement on two distinct views of the data; multiple kernel learning algorithms exploit kernels that naturally correspond to different views and combine kernels either linearly or non-linearly to improve learning performance; and subspace learning algorithms aim to obtain a latent subspace shared by multiple views by assuming that the input views are generated from this latent subspace.

4.6 Semi supervised Learning:

Here we have adopted one of the semi-supervised learning approaches. Naturally, we can generate multiple learners with these multiple view and then use the multiple learners to start disagreement- based semi supervised learning.

Let's describe a two view based semi supervised learning classifier.

Let L= labeled data set={ (a_1,b_1), (a_2,b_2), ……..(a_m,b_m)}

U=unlabeled data set={a_1',a_2',……a_n'}

We can construct a function **A** ⟶ **B** to define L and U as a learning algorithm.

Where A= Input Space

B= Output Space

a_i ,a_j∈ A, i=1,2,……..,m j=1,2,……..,n

In the context of two views, A and B can be represented as L= { (<a_1,b_1>, c_1), (<a_2,b_2>, c_2), ……..(<a_m, b_m>, c_m)} And U={ (<a_1',b_1'>, c_1'), (<a_2',b_2'>, c_2'), ……..(<a_n', b_n'>, c_n')} respectively.

4.7 multi-view semi supervised algorithm:

To the best of our knowledge, multi-view learning has not been extensively studied in NIDS false alarm reduction. In the literature, we can only find one relevant article on alarm reduction (Chiu et al., 2010), which developed a multi-view semi-supervised algorithm called two-teachers-one-student (2T1S). Their algorithm combines the concepts of co-training and consensus training. Through co-training, the classifier generated by one view can "teach" other classifiers constructed from other views to learn, and vice versa; and by consensus training, pre-dictions from more than one view can provide higher confidence for labeling unlabeled data.

In this work, we advocate the use of semi-supervised learning as it can utilize both labeled data and unlabeled data without human intervention. As mentioned earlier, the number of labeled data is insufficient and is expensive to obtain in the area of intrusion detection. In contrast, unlabeled data are abundant and easy to collect. Thus, semi-supervised learning can greatly reduce the workload of analysts and it is the most important reason for us to choose it. Our work is different from Chiu et al. (2010) in that we develop a multi-view based algorithm on NIDS alarms directly rather than TCP connections, while the features and algorithms are distinct as well. Besides, we further build a real alarm reduction system and evaluate it in real environments. It is worth noting that Mao et al. (2009) also developed a multi-view-based approach for detecting intrusions, through combining semi-supervised learning and active learning. In their work, active learning will generate a set of ambiguous in-stances which require an expert to label. In contrast, 2T1S and our algorithm do not need any human intervention. Due to this reason, their algorithm (Mao et al., 2009) is not included in our evaluation part. In the next section, we will give an in-depth description of our developed system.

4.8 MVPSys(Multi-view based false reduction system)

A three-layered architecture is given below. There are mainly three objectives of our proposed system. i) The system can extract proper feature from incoming NIDS alarm. Then it can construct a multi-view dataset that is two views on data set; one is source feature and another is destination feature. ii) A multi-view based semi-supervised learning algorithm can work on both labeled and unlabeled data automatically. iii) It reduces false alarm rate in both offline and real time environment. The proposed system is having four major components which is shown in the diagram. These are Preprocessing, Future extraction, Alarm classifier and Alarm filter.

4.8.1 Preprocessing:

It is used i) to check the format of incoming alarm. ii) To identify some incomplete alarm that is an incoming NIDS alarm may miss some contents like alarm description during the transmission. Hence, before processing the format needs to be checked which can guarantee the stability and reliability of the whole system?

4.8.2 Feature extraction:

It extracts features from incoming alarms and dividing into two views. One is called feature preparation which converts all incoming alarms into a common format. Second one is called feature extraction which collects these common features and converts them into two attribute sets: destination feature set and source feature set. Destination feature set includes the attributes related to destination environment such as destination IP address, port address , target application, generation id, priority ,classification etc. similarly source feature set includes the attributes such as

source IP address, port number , source operating system etc.

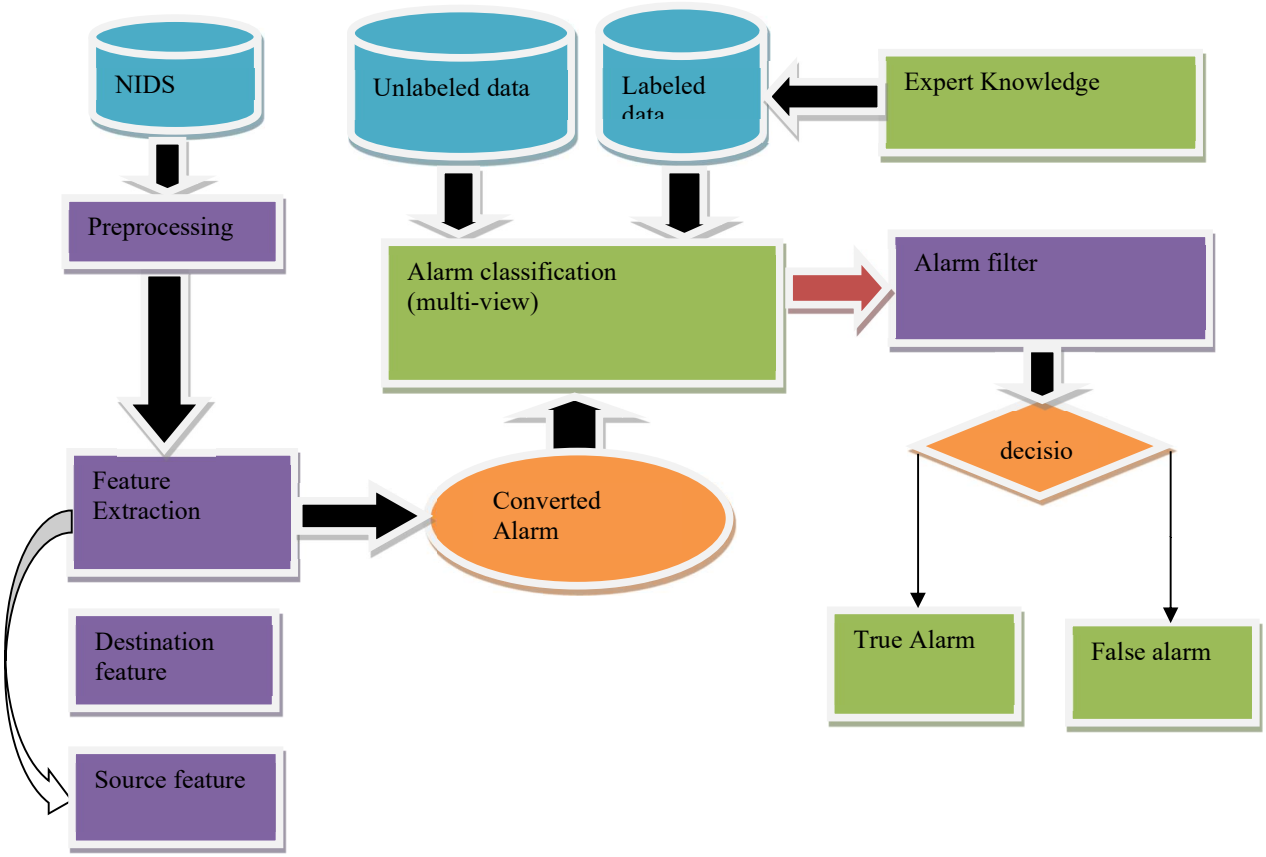

Fig-1 multi-viewed architecture of NIDS.

4.8.3 Alarm Classifier:

This component classify alarms in two main phases: training phase and classification phase. In training phase, a model is established for semi supervised learning from both labeled and unlabeled alarms. Also an expert can update labeled alarms to re-train the algorithm. In classification phase, this component classifies NIDS alarms into false positives and true positive based on semi-supervised algorithm.

4.8.4 Alarm Filter

It handles the classified data from above component by filtering out false alarms and maintains true alarms (saving them to database).

4.8.5 Implementation:

We have adopted a prototype approach that is MVPSys to implement two-view based semi-supervised learning algorithm in detail. Here the prototype MVPSys is given in fig-2 what we took for implementation.

From figure-2 we find there are four major components in our proposed system: preprocessing, feature standardization, alarm classifier and alarm filter. Preprocessing unit can check the format of incoming alarms including both training data and test data or real test data. Feature standardization having two buttons: format conversion and feature extraction. Alarm classifier is a semi-supervised algorithm applied both on labeled data and unlabelled data. At last alarm filter shows' 1' for true alarms and '0' for false alarms. The whole system was developed based on Java (about 1300 lines of codes) and can be executed from a *jar* file, which can be run in any Java-compatible platforms.

The system calls the API from Weka (WEKA, 2015) to realize the conventional algorithms (see Fig. 2), with the purpose of avoiding any implementation bias. Weka is an

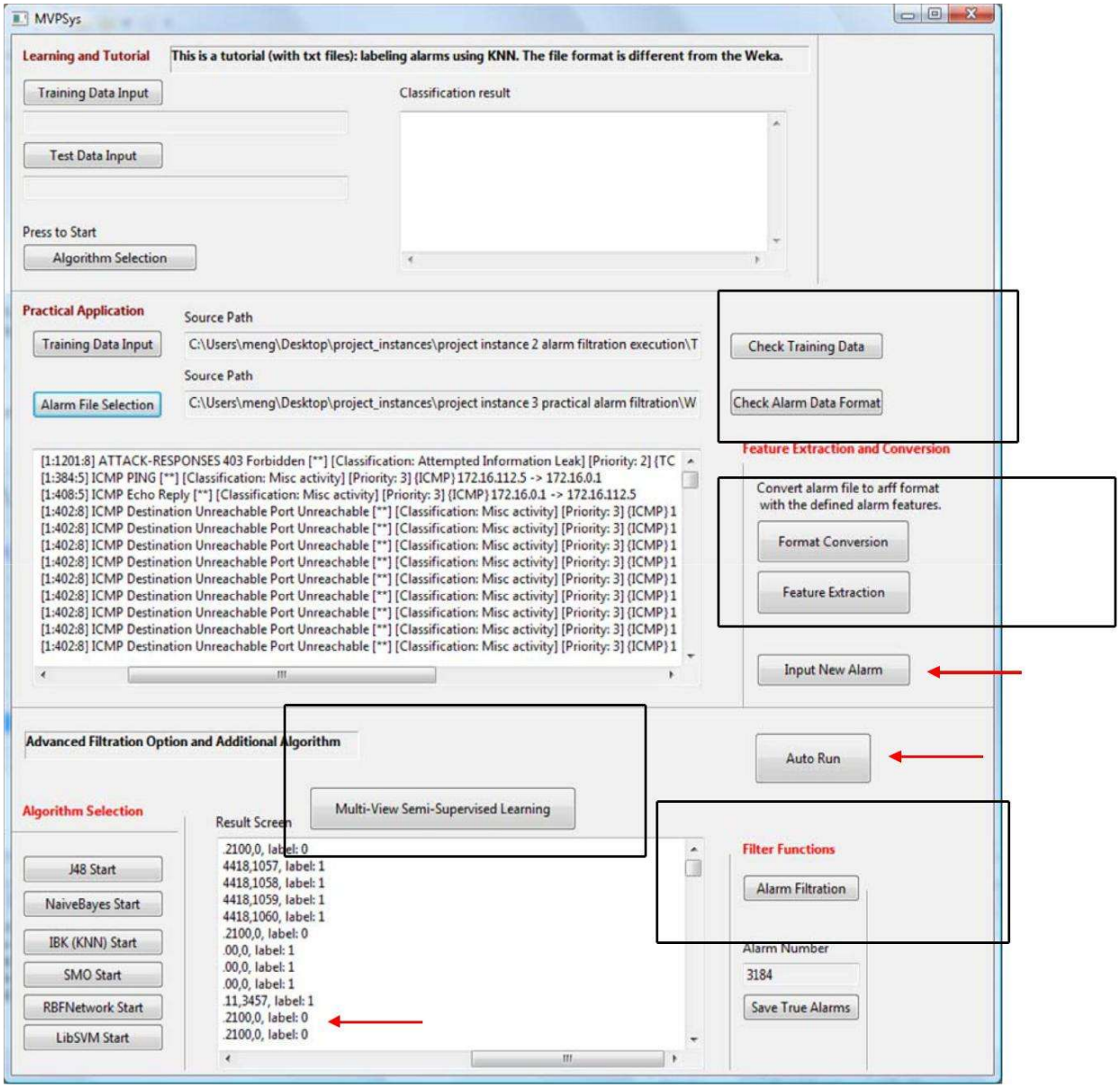

Figure 2. Our developed MVPSys

open-source tool as well as a collection of machine learning algorithms for data mining tasks. Our algorithm is also implemented in Weka platform. We used the default settings for these algorithms based on Weka platform. In practice, this system can be conducted in three modes:

- *Tutorial mode.* On the top of the system, we can input training data and test data to learn the performance of a particular algorithm in a text file format. Then we can select an algorithm and study the performance.
- *Off-line mode.* In the middle of the interface, the system can perform false alarm reduction off-line. We can simply input training data and incoming alarms, and run the algorithm. Note that all data are processed into *Weka* format.
- *Real-time mode.* The system can be configured to conduct false alarm reduction in real-time, in which we can press the button *auto run* to run the system. In this case, the proto-type can automatically get data from pre-

defined paths, extract features into *Weka* format, and filter out false alarms and save true alarms.

5. Two-view based semi-supervised learning algorithm

In this work, we use *MVPSys* to process Snort alarms as a study. There are two reasons for this selection. First, Snort (2015) is an open-source signature-based NIDS, and is very popular and widely adopted in both academy and industry. Second, our CSLab integrates Snort as one of the major intrusion detection systems to detect abnormal events, thus, it is easier for us to evaluate the performance of *MVPSys* in such a real environment.

5.1 One Labeled with Two Views Algorithm (OLTV)

Let's learn about labeled dataset before using unlabelled data. Let A and B denote two sufficient views. ({a,b},c) denote a labeled instance with class c, Given ($\{a_0,b_0\}$,1) and an unlabeled dataset U=($\{a_i,b_i\}$,c_i) (i=1,2,3,…….; c_i is unknown), our goal is to exploit U to enrich the labeled data.

Here we apply OLTV algorithm (Zhou , 2007) to exploit the unlabeled instances effectively and improve the learning performance. The main advantage is the correlation between the two views. we took this particular algorithm because the quality of the additional labeled instances derived by the OLTV algorithm is much better than that derived by using strategies such as K-nearest neighbor in the original feature space, which uses the k unlabeled instances nearest to ($\{a_0$, b0},1) as additional positive instances as additional negative ones(Zhou, 2010).The algorithm is given below

Process:

1. $L_P \leftarrow seed$, $L_N \leftarrow \varnothing$
2. Identify all pairs of correlated projections, obtaining α_i, β_i *and* λ_i .
3. For $j = 0, 1, 2, \ldots, l-1$ do *Project* $\langle a_i, bi \rangle$ into the m pairs of correlated projections.
4. For $j = 1, 2, \ldots, l-1$ do compute ρ_i
5. $P \leftarrow argmax_{\gamma+}(\rho_i)$, $N \leftarrow argmin_{\gamma-}(\rho_i)$
6. For all $\langle a_j, b_j \rangle \in P$ do $L_p \leftarrow L_P \cup (\langle a_j, b_j \rangle, 1)$
7. For all $\langle a_j, b_j \rangle \in P$ do $L_P \leftarrow L_P \cup (\langle a_j, b_j \rangle, 0)$
8. $L \leftarrow L_P \cup L_N$, $U \leftarrow U - (P \cup N)$

Output: L, U.

The main advantage of is that if we can design tow sufficient views for a concerned task , then asking the user to label only one example for the target class is sufficient for training a good predictor , which will make machine learning more readily available.

Let n denote number of identified pairs of correlated projections. In the jth projections, the similarity between an original unlabeled instance ($\{a_i,b_i\}$) and the labeled instance ($\{a_0,b_0\}$) can be measured as $sim_{i,j}$. Due to ({x0,y0},1), $p=\sum_{j=1}^{m} \lambda_j sim_{i,j}$ describe the confidence of ($\{a_i,b_i\}$) being positive instance where λ_j is a co-efficient. Thus positive and negative instances can be assigned according to the highest and lowest p values. At first OLTV is run then additional labeled training examples obtained and the semi supervised learning methods can be executed.

5.2 Semi-supervised Learning

In the literature, it is noted that most traditional multi-view learning algorithms require independent and redundant views; however, it is difficult to fulfill this requirement in most scenarios (Zhou and Li, 2005). The situation is the same with our two-views in this work, thus, it is crucial to develop or employ an algorithm that does not need or can lose the conditions of independent and redundant attributes. In this work, we therefore employ a disagreement-based ensemble co-training algorithm based on our previous work (Li et al., 2014), which does not require independent and redundant attributes, but to use multiple base classifiers with different learning algorithms instead of using the same base learner on the different subsamples of original labeled data. As a study, we employ a well-known co-training algorithm developed by Blum and Mitchell (1998) in the evaluation to compare the performance. The details of these algorithms are described below:

1. Disagreement-based semi-supervised learning. For our algorithm, each classifier h is first trained on the original

labeled data. Ensembles *H* are then established by means of all classifiers except one (*eh*) to search for a subset of high confidence unlabeled data. These ensembles estimate the error rate for each classifier from the agreement among the classifiers. Later, a subset of *U* is selected by *eh* for *h*. Data that can improve the error over a pre-defined threshold are added to the labeled training dataset. In this case, each classifier has its own training dataset. Note that data that are labeled for the classifier are not deleted from the unlabeled dataset. The above training process will be repeated until there are no more data that can be labeled to improve the performance of any classifier. An outline of this co-training is shown in Table 4 and more details can be referred to Meng and Kwok (2012).[4]

2. Blum and Mitchell algorithm. This is a well-known co-training algorithm that was developed by Blum and Mitchell (1998). They assumed that the data have two sufficient and redundant views (i.e., attribute sets), where each view is sufficient for training a strong learner and the views are conditionally independent to each other given the class label.

In co-training, each learner is generated using the original labeled data. Then, each learner will select and label some high-confident unlabeled examples for its peer learner. Later, the learners will be refined using the newly labeled examples pro-vided by its peer. With such a process, when two learners disagree on an unlabeled example, the learner which misclassifies this example will be taught by its peer. The whole process will repeat until no learner changes or a pre-set number of learning rounds has been executed (Zhou and Li, 2010). The specific co-training algorithm is described in Table 5 while de-tailed settings can be referred to Blum and Mitchell (1998).

6. Evaluation

In this section, we have evaluated the performance of our proposed *MVPSys* with two datasets and under two real network environments. In the remaining parts, we describe our experimental methodology and analyze the experimental results. Here, we have conducted three experiments where the performance of our proposed system is explored.

Experiment No.1: In this experiment, we use two datasets in offline mode to explore its performance. One is DARPA dataset and other is a private dataset from one of the project.

Experiment No.2: In this experiment, we investigated the practical performance of MVPSys in a real network environment in online mode.

Experiment N0.3: Here we have deployed the system in a collaborative intrusion detection network where the performance of MVPSys is validated in real world application.

The performance is evaluated in three parameters: classification accuracy, Hit Rat and AUC(area under an ROC curve).

Accuracy= correctly classified alarm/ Total alarms

Hit Rate= False alarm classified/ Total false alarm generated

Area under an ROC curve (AUC): ROC is a graphical plot that illustrates the performance of a binary classifier system by plot-ting the fraction of true positives out of the total actual positives. AUC is the area under the curve of ROC. Generally, the larger the AUC, the better the experiment is as predicted by the existence of the classification (Rosset, 1989).More specifically, *classification accuracy* is used to measure the capability of identifying both true alarms and false alarms, while the opposite is *error rate*. *Hit rate* is to measure the capability of detecting false alarms. Intuitively, a better classifier is desirable to have higher classification accuracy and a higher hit rate.

For comparisons with supervised machine learning algorithms, we employ a set of single supervised learning algorithms (e.g., J48) and ensemble supervised algorithms (e.g., J48 + IBK) in the evaluation. To compare with semi-supervised learning algorithms, we mainly employ 2T1S in the evaluation due to two reasons: (1) it is the only relevant work on applying multi-view to alarm reduction; and (2) it uses a semi-supervised learning algorithm without human intervention. As active learn-ing is used in Mao et al. (2009) that requires an expert to label alarms, their algorithm is not

considered in the evaluation and a relevant discussion is made in Section 6.

6.1 Experiment-1:

Here two data sets are analyzed by proposed system whose statistics is given below.

Statistics	DARPA Dataset	Private Dataset
False Alarms	14,295	6237
True Alarms	5910	2325
Unlabeled Alarms	600	600

We have used Snort Version 2.9.3.1 for our above analysis. Let us see some of Experimental results for both DARPA and real data set.

6.1.1 Experimental Results for DARPA dataset:

Here we randomly took 300 labeled alarms including 150 positive and 150 negative points to train our system. We run our algorithm at 80 iterations under cross validation 10 times. The performance in terms of accuracy, hit rate and AUC is given in fig number -3.

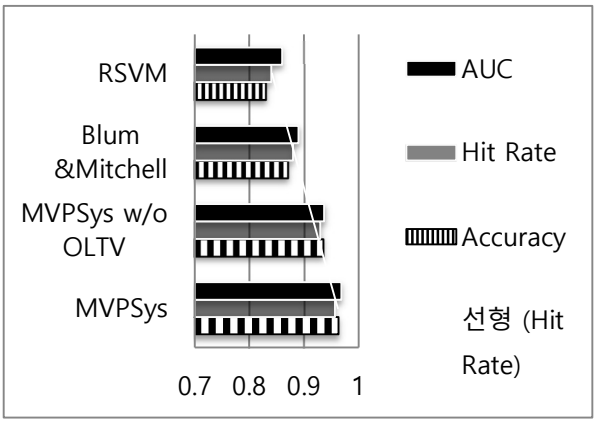

Fig.3 Performance of some related algorithms for DARPA data set.

It is found that our system can achieve the best classification accuracy of 96.5%,hit rate of 95.9% and AUC of 0.975 as compared to other related algorithms. We saw in the MVPSys without OLTV algorithm, the accuracy and hit rate is decreased to 93.6% and 93.3%, respectively because the OLTV algorithm can make a semi supervised learning algorithm more effective to learn multi-view unlabeled data.

6.1.2 Comparison with Supervised learning:

Here we have seen the number of traditional supervised classifiers and also some ensembles and compared these with our algorithm. It has been observed that among the single algorithms, SVM can achieve the best classification results with accuracy of 88.5% , hit rate of 87.5% and AUC of 0.878. Similarly among ensembles, the ensemble (j48+IBK) can obtain the best accuracy of 90.7%, hit rate of 91.4% and AUC of 0.921. So we can say MVPSys can outperform these supervised learning classifiers. The performance of supervised classifiers is shown below.

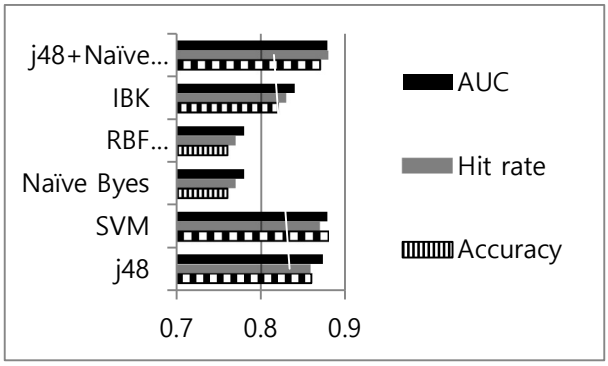

Fig.4 Performance of Some of Supervised Classifiers

6.1.3 Experimental results for the private (real) data set:

For the private data, we similarly select 300 labeled alarms in a random way, including 150 positive and 150 negative points to train our system. Then we run our algorithm at 80 iterations and conduct the experiment under cross validation (10 times). The aim of this experiment is to further investigate the performance of our approach with real traffic. The results of *MVPSys* regarding classification accuracy, hit rate and AUC are shown in Fig. 5. It is noticeable that our algorithm can achieve an accuracy of 96.7%, hit rate of 96.8% and AUC of 0.972, respectively. Similar to the above experiment, Fig. 5 shows a comparison among several related algorithms, while Fig. 6 describes a comparison among some supervised classifiers and ensembles. These comparisons are stated below.

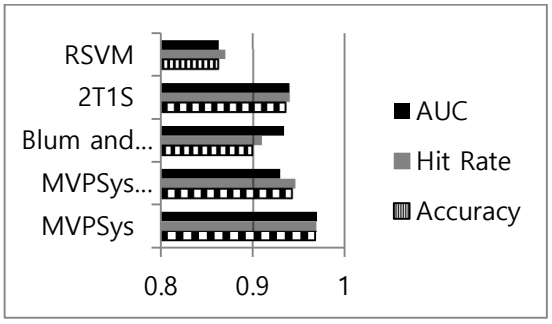

Fig.5 The results of classification accuracy, hit rate and AUC for the private dataset

6.1.4Comparision with supervised learning algorithm

In my experiment, then I have compared our algorithm with some supervised classifiers and ensembles which is given as below.

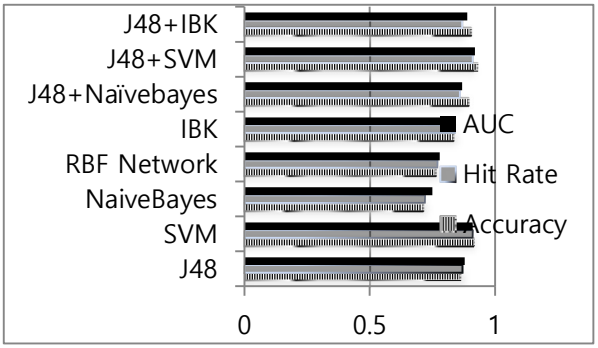

Fig.6 Results of Supervised classifiers for the Private dataset

From above we can clearly define that our algorithm achieves better performance than these supervised classifiers. Here we have used WEKA for our analysis.

6.2 Experiment 2

In this experiment, we deploy MVPSys in a real network environment that is within our CSLab to investigate its practical performance. The system was deployed on a windows platform with Intel core 2 Duo CPU, processor 2.8 GHz and 4GB of RAM. It is seen that every day, there are over thousands alarms that could be generated by Snort which shown in following table. To obtain ground-truth information, we invited three security officers including one chief security administrator to label and validate the alarms.

Table-6: Snort alarms produced on each day in real environment and the remaining alarms after filtration.

Days	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
The number of generated alarms	1724	2493	1870	2571	3223	2645	2823
Remaining alarms after filtration	114	165	102	145	176	134	102

Our system sets to work in real-time mode and is deployed behind Snort to collect and process all generated alarms. Several additional settings are described as below:

- The Snort alarms should be outputted in a fixed path;
- *MVPSys* retrieves the training data and current alarms from the given paths;
- An administrator can input new labeled alarms as training data at any time.

We train the system with 300 labeled alarms including half positive and half negative over 70 iterations to make the classification performance stable and then we run the system in the network

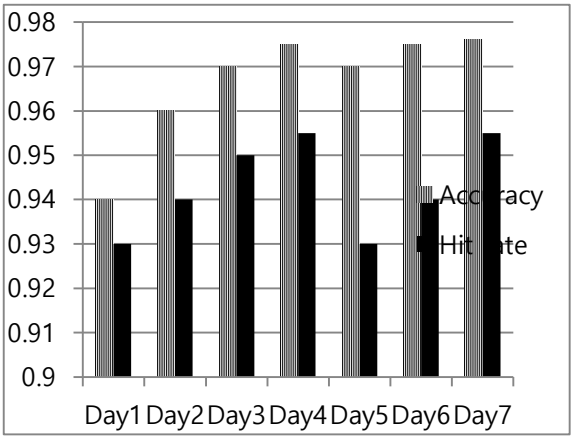

Fig.7 Filtration accuracy and hit rate of MVPSys in the network environment over a week

It is noticeable that our system can reduce the false alarms in the range of 93% - 97.6% with high classification accuracy and hit rate. Taking Day 7 as an example, only 102 alarms remained from 2823 alarms where accuracy is 97.6% and hit rate is 96.4%. The comparison of classification accuracy is given in fig 8.it is seen that our system can outperform the algorithm of 2T1S and keep a stable performance at a high accuracy level. These promising results demonstrate the effectiveness of our proposed alarm filter in a real network application.

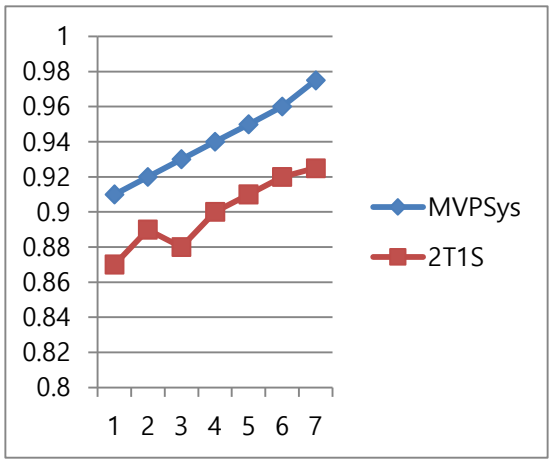

Fig-8 Comparision on accuracy over one week: MVPSys Vs 2T1S

6.3 Experiment 3:

In this experiment, we deploy MVPSys in CIDN (collaborative intrusion detection network) within an organization including over 100 personnel. A CIDN enables an IDS node to exchange information and learn experience with other nodes. In particular, this CIDN is a wired network and consists of 32 Snort nodes. The high-level architecture of this CIDN is depicted in Fig. 10. All nodes can access Internet resources freely and require computing resources through a server The main purpose of this experiment is to test *MVPSys* in a real and public network environment (not lab environment) and validate its performance. We randomly deploy *MVPSys* in ten nodes to filter false alarms, and the whole evaluation was conducted for a week collaborating with the security officers. The settings are the same as described in the last experiment. To get the ground-truth information, we invited four security officers from this organization to guide labeling and validate the results. The average accuracy results for these nodes are shown in Fig. 11.

It is easily seen that *MVPSys* outperforms 2T1S algorithm in terms of classification accuracy in this real network environment. For example, for *Node 2*, the average accuracy of *MVPSys* is 94.3%, but is 91.8% for 2T1S algorithm. In addition, it is found that the performance of *MVPSys* is more stable. The security officers also confirm our findings and consider that the performance is encouraging in real-world scenarios.

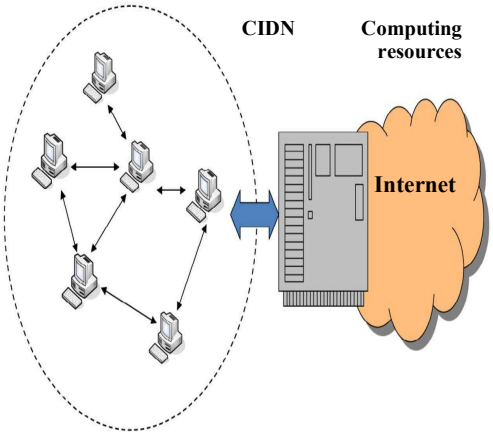

Fig 9 Architecture of applied CIDN

7. Discussion:

Our work complements the existing research efforts and aims to stimulate more studies in real network environment. In the literature, we find most related algorithms are only tested using datasets. But we find that real traffic is often more dynamic and complicated. This is the main reason why we apply our system in real network environment to test the performance. Again under a real environment, it is feasible to validate the caused workload and stability of an algorithm.

8. Conclusion:

In NIDS, the most challenging job is to reduce false alarms. Many machine learning algorithms have been deployed as a false alarm filter. But here we have developed a practical multi-view based approach to reduce false alarm effectively. We have shown that the experimental results on two data sets and in two real network environment demonstrate that our proposed MVPSys is more effective and achieve better performance in terms of accuracy as compared to the similar algorithms.

Our topic leaves many possible future scopes like deployment of our system in other real network environment and addition of active learning algorithm with our approach. Also we can apply our approach to other research field where false alarm is the big challenge.

References

[1] DARPA intrusion detection evaluation data set,<http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/1999data.html>; 1999. [access on August, 2015].

[2] Snort. The open source network intrusion detection system. Homepage: <http://www.snort.org/>; 2015. [access on August, 2015].
[3] Meng Y, Kwok L-F. Adaptive non-critical alarm reduction using hash-based contextual signatures in intrusion detection. Comput Commun Elsevier 2014;38:50–9.
[4] Hubballi N, Suryanarayanan V. False alarm minimization techniques in signature-based intrusion detection systems: a survey. Comput Commun 2014;49:1–17.
[5] Meng Y, Li W, Kwok L-F. Towards adaptive false alarm reduction using Cloud as a Service. In: Proceedings of CHINACOM. 2013. p. 420–5.
[6] Sun S. A survey of multi-view machine learning. Neural Comput Appl 2013;23:2031–8.
[7] Wang S-H. An exchange framework for intrusion alarm reduction in mobile ad-hoc networks. J Compute2013;8(7):1648–55.
[8] WEKA. Data mining software in Java. Homepage: <http://www.cs.waikato.ac.nz/ml/weka/>; 2013
[9] Xu C, Tao D, Xu C. A survey on multi-view learning. CoRR abs/ 1304.5634, pp. 1–59. 2013.
[10] Zhang M, Mei H. A new method for filtering IDS false positives with semi-supervised classification. In: Huang D-S, Jiang C, Bevilacqua V, Figueroa JC, editors. ICIC 2012. LNCS, vol. 7389. Heidelberg: Springer; 2012. p. 513–19.
[11] Meng Y, Kwok L-F. Enhancing false alarm reduction using pool-based active learning in network intrusion detection. In: Proceedings of the 9th information security practice and experience conference (ISPEC), lecture notes in computer science 7863. Springer; 2013b. p. 1–16.
[12] Kotthoff L, Gent IP, Miguel I. An evaluation of machine learning in algorithm selection for search problems. AI Commun 2012;25(3):257–70.
[13] Cui Y, Shi J, Wang Z. Multi-state adaptive BIT false alarm reduction under degradation process. IEEE Trans Instrument Meas 2014;64(3):671–682.
[14] Seliya N, Khoshgoftaar TM. Active learning with neural networks for intrusion detection. In: Proceedings of the 2010 IEEE international conference on information reuse and integration (IRI). 2010. p. 49–54.
[15] Chiu C-Y, Lee Y-J, Chang C-C, Luo W-Y, Huang H-C. Semi-supervised learning for false alarm reduction. In: Proceedings of the 10th IEEE international conference on data mining (ICDM). 2010. p. 595–605.
[16] Al-Mamory SO, Zhang H. Intrusion detection alarms reduction using root cause analysis and clustering. Comput Commun 2009;32(2):419–30.
[17] Li Y, Guo L. An active learning based TCM-KNN algorithm for supervised network intrusion detection. Comput Secur 2007;26(7–8):459–67.
[18] Mao CH, Lee HM, Parikh D, Chen T, Huang SY. Semi-supervised co-training and active learning based approach for multi-view intrusion detection. In: Proceedings of the 2009 ACM symposium on applied computing (SAC). 2009. p. 2042–8.
[19] Alharby A, Imai H. IDS false alarm reduction using continuous and discontinuous patterns. In: Ioannidis J, Keromytis AD, Yung M, editors. ACNS 2005. LNCS, vol. 3531. Heidelberg:Springer; 2005. p. 192–205.
[20] Ghosh AK, Wanken J, Charron F. Detecting anomalous and unknown intrusions against programs. In: Proceedings of the 14th annual computer security applications conference (ACSAC). 1998. p. 259–67.