

# $M$ of $N$ Features vs. Intrusion Detection

Zhuowei Li and Amitabha Das

School of Computer Engineering, Nanyang Technological University  
zhwei.li@pmail.ntu.edu.sg, asadas@ntu.edu.sg

**Abstract.** In order to complement the incomplete training audit trails, model generalization is always utilized to infer more unknown knowledge for intrusion detection. Thus, it is important to evaluate model generalization with respect to the detection performance of intrusion detection. In this paper, based on a general intrusion detection methodology,  $M$  out of  $N$  features in a behavior signature are utilized to detect the behaviors ( $M \leq N$ ) instead of using all  $N$  features. This is because  $M$  of  $N$  features in a signature can generalize the behavior model to incorporate unknown behaviors, which are useful to detect novel intrusions outside the known behavior model. However, the preliminary experimental results show that all features of any signature should be fully utilized for intrusion detection instead of  $M$  features in it. This is because the  $M$  of  $N$  features scheme will make the behavior identification capability of the behavior model lost by detecting most behaviors as ‘anomalies’.

## 1 Introduction

Intrusion detection has become a very important defense mechanism in the face of increasing vulnerabilities exposed in today’s computer systems and the Internet [2], where authentication, cryptography, and access control mechanisms routinely prove inadequate in preventing new and increasingly numerous and disastrous attacks. In general, there exist two approaches for detecting intrusions into computer systems and networked information systems: signature-based intrusion detection (a.k.a. misuse detection), where an intrusion is detected if its behavior matches existing intrusion signatures, and anomaly-based intrusion detection, where an intrusion is detected if the resource behavior deviates from known normal behaviors significantly.

In principle, most intrusion detection techniques build the behavior models from the known knowledge of a computing resource (e.g., the training audit trails, the program binary/source code etc.). In general, this knowledge is always incomplete due to the limits of the knowledge collection tools. To make up for the knowledge incompleteness, most existing intrusion detection techniques try to infer the unknown behaviors via *model generalization* [6] [7]. In this paper, as a part of effort to analyze the problems in intrusion detection, we evaluate the usefulness of the model generalization led to by  $M$  of  $N$  features in a signature with respect to its influence on the detection performance of the behavior model. For example, suppose that there exist one signature ‘*height*  $\in$  (156cm, 189cm], *weight* = (45kg, 75kg], and *Nationality* = Singapore’. If all  $N$ (=3) features are utilized,

the instance ‘*height = 174cm, weight = 65kg, and Nationality = China*’ is not identified by the signature. But if only any  $M(=2)$  features are utilized, the signature will identify the instance.

In summary, our main contributions in this paper are as follows. (1) A novel intrusion detection methodology is introduced briefly. (2) Model generalization led to by M of N features in a signature is discussed, and then an evaluation methodology on it is designed. In addition, an average detection cost function is defined to quantify the detection performance for intrusion detection.

The remaining parts of this paper are organized as follows. Section 2 talks about the related work. Section 3 describes the formal intrusion detection methodology in brief. In section 4, an evaluation methodology is also designed for the  $M$  of  $N$  features scheme. Experiments in section 5 reveal the useless of the scheme on intrusion detection. Lastly, we draw conclusions in section 6.

## 2 Related Work

Our research work in this paper is generally related to model generalization in the behavior model. First, the intrusion signatures in SID techniques can be generalized to cover more behavior space. In [1], using a fitness function which depends on false positive rate and detection rate, the generalized signatures (represented by a *finite state transducer*) is optimized by the evolution programming. In general, the model generalization on intrusion signatures can solve the intrusion variations detection problem partially.

Secondly, the normal behavior model of AID techniques can be generalized as well to detect novel instances, and it can be done in several ways. Based on a distance metric and a distance threshold [3][9], the instances in the existing audit trails are clustered unsupervisedly, and the new instances are labeled by the existing instances in their clusters. In statistical methods for intrusion detection [6] [7] [9], the (statistical) resource usage profiles are mined from the existing audit trails. The novel instances are detected according to whether they fall into these profiles. Among these two styles, the existing audit trails are modeled inexactly to accommodate more resource behaviors in the profiles, and thus to achieve the model generalization.

Most of past works only credited the overall efficiency of an intrusion detection technique to such model generalization, and there is hardly any evaluation of the effect of the model generalization. This is partially due to the difficulty in pinpointing the contribution of model generalization to the overall efficiency. Fortunately, our methodology not only overcomes the problem, it also allows one to adjust the extent of model generalization.

## 3 Brief Introduction: An Intrusion Detection Methodology

Any intrusion detection system builds the behavior models of the resources using a set of features, or a *feature vector*  $FV = \{F_1, F_2, \dots, F_n\}$ , where  $F_i$  is a feature

in the feature set. Every feature in the feature vector can be one of these types: A feature associated with an instant of time (e.g., *the fields in the current packet*), or with a time interval (e.g., *the number of SYN packets within 2 seconds*), or with the context of a current event (e.g., *the system-call events in stide [4], the state events in STAT [8]*). The context is defined over the timeline proceeding the point in time when the event in question happens. In general, an atomic feature  $F_i$  in the feature vector can be categorized into *nominal*, *discrete* or *continuous* one. A feature vector for intrusion detection can contain any number of nominal, discrete, and/or continuous features. In addition, a feature can also be as complex as a compound feature (see Section 3.3).

In this methodology, we assume that there is a training audit trail indexed by its timestamp, in which the normal and the intrusion audit trails are labelled correctly. We also assume that the training audit trails represent our known knowledge about the computing resource. Then, the instances of the feature vector are collected from the training audit trails as  $\{I_{FV}^1, I_{FV}^2, I_{FV}^3, \dots\}$ . For each feature instance  $I_{FV}^i$ , there is a **status** that indicates the label of audit trails where it is collected. For example, if an instance  $I_{FV}^i$  is left by an intrusion ‘Nimda’, its status is ‘Nimda’.

### 3.1 Basic Concepts and Notations

For a feature  $F$ , several of its basic concepts are defined formally as follows.

- Its feature space  $Dom(F)$  is the defining domain of the computing resource.
- Any value in  $Dom(F)$  is defined as a feature value  $v_F$ , and  $v_F \in Dom(F)$ . In general, there are many feature values in the feature space  $Dom(F)$ .
- A feature range  $R_F(v_F^1, v_F^2)$  is the range between any two feature values  $v_F^1$  and  $v_F^2$  in its feature space, which includes all feature values falling between  $v_F^1$  and  $v_F^2$ . For a discrete or continuous feature,  $R_F(v_F^1, v_F^2) = [v_F^1, v_F^2]$ . For a nominal feature, every feature value is independent. Thus, each nominal feature value is referred to as a feature range in this methodology so that for a nominal feature  $F$ ,  $[v_F^i] = [v_F^i, v_F^i]$ . If a feature value  $v_F^j$  is within the bounds of a feature range, we say that it falls within the feature range, denoted as  $v_F^j \in R_F(v_F^1, v_F^2)$ . The concept of *feature range* is used to treat uniformly every (nominal, discrete, or continuous) feature in our methodology. For  $R_F(v_F^1, v_F^2)$ , we further define  $upper(R_F) = v_F^1$  and  $lower(R_F) = v_F^2$ .

**Notations.** In reality, some feature values of a feature will never occur, and thus unreasonable. As mentioned above, there is a series of feature instances  $\{I_{FV}^1, I_{FV}^2, \dots\}$  collected from existing audit trails. The feature values in every feature instance  $I_{FV}^i$  are *reasonable* as it occurs. The following notations are given (Note that in order to avoid cluttering the expressions, we have dropped the subscript of  $F$ ): if  $F \in FV$ ,

- $v(I_{FV}^i, F)$  is the feature value of the feature  $F$  in the feature instance  $I_{FV}^i$ .
- $I(v_F, F)$  is the set of feature instances whose values of  $F$  are equal to  $v_F$ .

$$I(v_F, F) = \{I_{FV}^k | v_F = v(I_{FV}^k, F)\}$$

–  $I(R_F, F)$  is the set of feature instances whose values of  $F$  fall in  $R_F$ .

$$I(R_F, F) = \{I_{FV}^k | v(I_{FV}^k, F) \in R_F\}$$

### 3.2 NSA Label

**Definition 1 (NSA label of a feature value).** *If a feature value  $v_F$  occurs only in the normal audit trails, it is normal. If it occurs only in the intrusive audit trails, for example, intrusion signatures, it is labeled as anomalous. Otherwise, i.e., if it occurs in both normal and intrusive audit trails, it is labeled as suspicious. For brevity, we will refer to the normal, suspicious, or anomalous label as the **NSA label** of the feature value  $v_F$ , denoted as  $L(v_F) = \{‘N’, ‘S’, ‘A’\}$ .*

Note that a feature value is either normal or anomalous in a specific feature instance, but its NSA label is collected from all related feature instances. We will further extend the concept of **NSA label** to feature ranges of a feature.

**NSA Labels of Feature Ranges.** With respect to a user-defined splitting strategy, the feature space  $Dom(F)$  can be split into a set of mutually exclusive feature ranges  $\{R_F^1, R_F^2, \dots, R_F^m\}$ , such that **(1)** there is no common feature value  $v_F$ , which falls in  $R_F^j$  and  $R_F^k$  at the same time ( $j \neq k$ ), and **(2)**  $I(R_F^i, F) \neq \Phi$  ( $i \geq 1$ ). Then, the concept of NSA labels can be extended to these feature ranges as follows. For the feature range  $R_F$ ,

$$\begin{aligned} L(R_F) = ‘N’ &\Leftrightarrow \forall i(v(I_{FV}^i, F) \in R_F \rightarrow L(v(I_{FV}^i, F)) = ‘N’) \\ L(R_F) = ‘A’ &\Leftrightarrow \forall i(v(I_{FV}^i, F) \in R_F \rightarrow L(v(I_{FV}^i, F)) = ‘A’) \\ L(R_F) = ‘S’ &\Leftrightarrow \exists i \exists j (v(I_{FV}^i, F) \in R_F \wedge L(v(I_{FV}^i, F)) = ‘A’) \\ &\quad \wedge (v(I_{FV}^j, F) \in R_F \wedge L(v(I_{FV}^j, F)) = ‘N’) \end{aligned}$$

**Feature Subspaces.** After grouping the feature ranges of a feature  $F$  based on NSA labels, we can partition the feature space  $Dom(F)$  into three feature subspaces: *normal*, *suspicious* and *anomalous*, denoted as  $N(F)$ ,  $S(F)$  and  $A(F)$ , respectively. Thus we have,

$$\begin{aligned} N(F) &= \{R_F^j | j \geq 1, L(R_F^j) = ‘N’\} \\ S(F) &= \{R_F^j | j \geq 1, L(R_F^j) = ‘S’\} \\ A(F) &= \{R_F^j | j \geq 1, L(R_F^j) = ‘A’\} \end{aligned}$$

In the following description, we denote  $\Omega(F) = N(F) \cup S(F) \cup A(F)$ .

### 3.3 Combining: Compound Feature

**Definition 2 (compound feature).** *With respect to two features  $F_1$  and  $F_2$ , a compound feature  $F_{12}$  is defined as a subset of the cartesian product of  $\Omega(F_1)$  and  $\Omega(F_2)$ , such that each element in this set actually represents at least one feature instance in the audit trails. In other words, if an ordered pair  $(R_{F_1}^a, R_{F_2}^b)$  is a compound feature range (i.e.,  $(R_{F_1}^a, R_{F_2}^b) \in \Omega(F_{12})$ ), then  $I((R_{F_1}^a, R_{F_2}^b), F_{12}) \neq \Phi$ .*

$$\Omega(F_{12}) = \{(R_{F_1}^a, R_{F_2}^b) | R_{F_1}^a \in \Omega(F_1), R_{F_2}^b \in \Omega(F_2), I((R_{F_1}^a, R_{F_2}^b), F_{12}) \neq \Phi\}$$

Based on the definition of *cartesian product*, for any feature instance  $I_{FV}^i \in I((R_{F_1}^a, R_{F_2}^b), F_{12})$ , it will be recognized by feature ranges  $R_{F_1}^a$  and  $R_{F_2}^b$  as well (i.e.,  $I_{FV}^i \in I(R_{F_1}^a, F_1)$  and  $I_{FV}^i \in I(R_{F_2}^b, F_2)$ ), and vice versa. Therefore,  $I((R_{F_1}^a, R_{F_2}^b), F_{12}) = I(R_{F_1}^a, F_1) \wedge I(R_{F_2}^b, F_2)$ . For the sake of ambiguity, we will refer to a single feature as an *atomic* feature.

**Theorem 1.** *The feature ranges of a compound feature are mutually exclusive, i.e., for two different feature ranges of a compound feature  $(R_{F_1}^a, R_{F_2}^b)$  and  $(R_{F_1}^c, R_{F_2}^d)$ , there is no such feature instance  $I_{FV}^i$  so that  $I_{FV}^i \in I((R_{F_1}^a, R_{F_2}^b), F_{12})$  and  $I_{FV}^i \in I((R_{F_1}^c, R_{F_2}^d), F_{12})$ .*

Similar to atomic features, every feature range of  $F_{12}$  has an NSA label, and all of its feature ranges are mutually exclusive (**Theorem 1**, please see its proof in the appendix). A compound feature space can be partitioned into three feature subspaces like an atomic feature, i.e.,  $\Omega(F_{12}) = N(F_{12}) \cup S(F_{12}) \cup A(F_{12})$ .

**Compounding More Features.** In summary, the compound feature built from two atomic features shows the same properties as any of its component atomic features. Therefore, we can treat the compound feature as an atomic one to build higher order compound features. Using this recursive procedure, the feature vector  $FV$  for intrusion detection can be converted into an equivalent  $n$ -order compound feature  $F_{1\dots n}$  with normal  $N(F_{1\dots n})$ , suspicious  $S(F_{1\dots n})$  and anomalous  $A(F_{1\dots n})$  feature subspaces. We will rewrite the compound feature ranges according to the following rule:  $(R_F^a, (R_F^b, R_F^c)) = (R_F^a, R_F^b, R_F^c)$ .

### 3.4 Behavior Signature

In our methodology, the behavior models of the resource are constituted by (*normal, suspicious and intrusion*) behavior signatures, which are defined as:

**Definition 3 (behavior signature).** *Assuming that there exists a feature vector  $FV = \{F_1, F_2, \dots, F_n\}$ , and that the feature ranges of every feature are determined beforehand. A behavior signature  $Sig_{FV}^i$  is a feature range of the compound feature  $F_{1\dots n}$  with its NSA label. In other words, the behavior signature is the combination of feature ranges of all features in  $FV$  labelled by its statuses of corresponding feature instances in the existing audit trails.*

As indicated in the above definition, every behavior signature<sup>1</sup> represents a state of the resource at a specified time point. According to NSA labels of signatures, the behavior model can be split into three parts: normal, suspicious, and intrusion behavior models. In anomaly-based intrusion detection, only the normal behavior model is utilized, but signature-based intrusion detection identifies intrusions based on the intrusive behavior model. However, the best scenario is to do intrusion detection using the complete behavior model.

<sup>1</sup> For brevity, ‘behavior signature’ will be simplified as ‘signature’ within the context of this paper.

## 4 Intrusion Detection via Signatures

### 4.1 Building Behavior Models

To use our methodology for intrusion detection, an splitting strategy is designed as follows to build the feature ranges for every feature. For nominal features, the splitting strategy do nothing except building one feature range for every feature value. For every discrete/continuous feature, an initial feature range is built for every feature value. Two initial feature ranges are *neighboring* if there are no feature values between them in the audit trails.

Specific for every discrete feature, the unknown feature subrange between any two neighboring initial feature ranges is split and combined into these two initial feature ranges as follows. If the size of the unknown feature subrange is an odd number  $n$ ,  $(n - 1)/2$  of it will combine into every side, and the left 1 is assigned to one side randomly. If the size is an even number  $n$ ,  $n/2$  of it will combine into every side. In contrast, specific for every continuous feature, the unknown feature subrange between any two neighboring initial feature range will be split equally and combined into both sides.

In the following step, if two neighboring feature ranges have the same NSA label, they will be combined into a single feature range by expanding its range size. This can economize the storage space for the ultimate behavior models.

### 4.2 Detecting Behaviors Using M of N Features in a Signature

In our evaluation methodology, an instance in the test audit trails will be detected as follows. Utilizing the feature ranges of every feature, a temporal signature will be formed for the instance. If it matches any signature in the behavior model with  $M$  among  $N$  features, the status list of the signature will be inserted into the status list of the temporal signature. Obviously, the status list of the temporal signature is empty initially. After comparing with all signatures in the behavior model, the detection results for the instance is aggregated into its status list.

Then, the average cost for every instance in the test audit trails is calculated to quantify the detection performance. For a normal behavior, it will be detected as an anomaly if the status list include other status(es) other than ‘normal’. Otherwise, it is detected as ‘normal’. For a intrusive behavior, it will be detected as the same intrusion if the status list is identical to the status of the behavior, and it will be detected as normal if the status list only include the ‘normal’ status. Otherwise, it will be detected an an ‘anomaly’.

### 4.3 To Measure the Detection Performance

The two main objectives of intrusion detection are (1) to detect the intrusions correctly (as anomalies), and (2) to identify the behaviors correctly (i.e. normal behaviors or its original intrusions). With respect to the detection results, every instance in the test audit trails will be assigned a *cost* value as the detection performance of the behavior model to it [5]. Specifically, if the *normal* instance

is detected as ‘normal’, the cost is 0, otherwise, the cost is 3. Simultaneously, if the intrusive instance is identified as its original intrusion label, the cost is 0. If the intrusive instance is detected as an anomaly, the cost is 1. If the intrusive instance is detected as ‘normal’, the cost is 3.

Suppose that there are  $T$  instances in the test audit trails. According to the detection results, several statistics are further defined as follows.

- $\#_{(N,N)}(M)$ : the number of *normal* instances detected as ‘normal’;
- $\#_{(N,A)}(M)$ : the number of *normal* instances, but detected as ‘anomalies’;
- $\#_{(N,*)}(M)$ : the number of *normal* instances in the test audit trails;
- $\#_{(I,I)}(M)$ : the number of *intrusive* instances detected as their original intrusions;
- $\#_{(I,A)}(M)$ : the number of *intrusive* instances detected as ‘anomaly’;
- $\#_{(I,N)}(M)$ : the number of *intrusive* instances detected as ‘normal’;
- $\#_{(I,*)}(M)$ : the number of *intrusive* instances in the test audit trails.

Where, it is obvious,

$$\#_{(N,*)}(M) = \#_{(N,N)}(M) + \#_{(N,A)}(M) \quad (1)$$

$$\#_{(I,*)}(M) = \#_{(I,I)}(M) + \#_{(I,A)}(M) + \#_{(I,N)}(M) \quad (2)$$

$$T = \#_{(N,*)}(M) + \#_{(I,*)}(M) \quad (3)$$

With respect to specific  $M$ , the average cost of every instance in the test audit trails is defined as:

$$cost(M) = (\#_{(N,A)}(M) \times 3 + \#_{(I,N)}(M) \times 3 + \#_{(I,A)}(M) \times 1) \times \frac{1}{T} \quad (4)$$

From above equation, with the increase of  $cost(M)$ , the detection performance with the parameter  $M$  becomes worse. Obviously, the average cost at  $M = N$  is the baseline for the detection performance. If  $cost(M) > cost(N)$ , the efficiency for intrusion detection has been degraded by such  $M$  of  $N$  scheme. Otherwise, it is useful for intrusion detection. An efficient intrusion detection technique will cause smaller average cost for every instance.

## 5 Experiments

We have chosen a typical dataset for network intrusion detection from KDD CUP 1999 contest, in which every record is an instance of a specific feature vector collected from the audit trails. This is because the dataset meets the requirements of our formal framework: labeled audit trails and intrusion-specific feature vector. The number of records in the datasets are: *training-4898431 records, test-311029 records*. For a detailed description of the datasets, please refer to ‘<http://www.cse.ucsd.edu/users/elkan/clresults.html>’.

### 5.1 Evaluating the $M$ of $N$ Scheme

In our experimental evaluations,  $N = 41$  and the parameter  $M$  is variable from 41 to 30. The behavior model is first built from the training audit trails. Then, every instance in the test audit trails is detected with respect to specific  $M$ , and the detection performance is quantified by the average cost of every instance within the detection results.

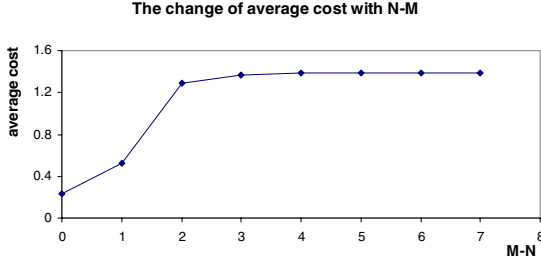


Fig. 1. The influence of M of N features scheme

**Experimental Results** The detection performance baseline with  $M = 41$  is  $cost(41) = 0.228$ . For the sake of comparison, we use  $N - M$  as the horizontal axis in Figure 1, in which the influence of M of N features scheme on intrusion detection is illustrated. It is obvious that the average cost of every instance is decreased with the increase of  $N - M$ , i.e. with the decrease of  $M$ . In other words, even though the M of N features scheme can generalize the behavior model, it will degrade the detection performance for intrusion detection. Therefore, the scheme should not be utilized to enhance the detection performance in intrusion detection.

**The Statistics About the Detection Results** In Table 1, the detection results are detailed with respect to varying  $M$ . In general, an efficient intrusion detection technique should identify most (normal and intrusive) behaviors, and the identification ability is indicated by the numbers in  $\#_{(N,N)}(M)$  and  $\#_{(I,I)}(M)$ . However, in Table 1, these two numbers are decreased with the decrease of  $M$ . In other words, with the decrease of the parameter  $M$ , the identification capability is degraded, and most normal and intrusive behaviors will be detected as ‘anomalies’. As an extreme case, all the behaviors will be detected as ‘anomalies’. This case will also occur if the behavior model is empty. In other words, the behavior model with more generalization caused by the M of N features scheme is almost no use for intrusion detection. In summary, the M of N feature scheme will largely degrade the detection performance for intrusion detection.

Table 1. The detection results with respect to M

M	$\#_{(N,N)}(M)$	$\#_{(N,A)}(M)$	$\#_{(I,I)}(M)$	$\#_{(I,A)}(M)$	$\#_{(I,N)}(M)$	cost(M)
41	57102	3491	215835	21635	12966	0.228293825
40	53739	6854	134578	102487	13371	0.524587739
39	8067	52526	10225	238353	1858	1.290892489
38	2571	58022	39	249817	580	1.368435098
37	26	60567	3	250342	91	1.389953991
36	2	60591	0	250354	82	1.390137254
35	1	60592	0	250418	18	1.389735362
34	0	60593	0	250429	7	1.389674275
33	0	60593	0	250436	0	1.389629263
32	0	60593	0	250436	0	1.389629263
31	0	60593	0	250436	0	1.389629263
30	0	60593	0	250436	0	1.389629263

## 6 Conclusions and Future Work

In this paper, we first present a formal intrusion detection methodology based on a general feature vector. Using the framework, the  $M$  of  $N$  feature scheme is evaluated with respect to the detection performance for intrusion detection. To achieve it, we also propose a average cost function to quantify the detection performance for intrusion detection. The experimental results show that, even though the  $M$  of  $N$  scheme can generalize the behavior model to cover more unknown behaviors, it will degrade the detection performance for intrusion detection by triggering more false alarms. More specifically, the identification ability of every signature will be lost with the decrease of  $M$ , i.e., with more generalization in the behavior model. The conclusion is critical for intrusion detection since all the features in a signature should be used to identify a (normal/intrusive) behavior, which does not follow our intuition.

In the future work, we will further utilize the formal framework to analyze the problems in intrusion detection, and try to propose solutions or suggestions for these problems. At the same time, we will propose an efficient intrusion detection methodology based on the behavior signatures of the computing resource.

## References

1. K.P. Anchor, J.B. Zydallis, G.H. Gunsch, and G.B. Lamont. Extending the computer defense immune system: Network intrusion detection with a multiobjective evolutionary programming approach. In *ICARIS 2002: 1st International Conference on Artificial Immune Systems Conference Proceedings*, University of Kent, 2002.
2. H. Debar, M. Dacier, and A. Wespi. A revised taxonomy for intrusion detection systems. *Annales des Telecommunications*, 55(7-8):361-378, 2000.
3. E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *D. Barbara and S. Jajodia (editors), Applications of Data Mining in Computer Security*, Kluwer, 2002.
4. S.A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151-180, 1998.
5. W. Lee, M. Miller, and S. Stolfo. Toward cost-sensitive modeling for intrusion detection. Technical Report No. CUCS-002-00, Computer Science, Columbia University, 2000.
6. W. Lee and S.J. Stolfo. A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information and System Security*, 3(4):227-261, Nov. 2000.
7. M.V. Mahoney and P.K. Chan. Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks. In *SIGKDD 2002*, July 23-26 2002.
8. G. Vigna and R.A. Kemmerer. NetSTAT: A Network-based Intrusion Detection System. *Journal of Computer Security*, 7(1):37-71, 1999.
9. K. Wang and S.J. Stolfo. Anomalous payload-based network intrusion detection. In *Proceedings of RAID*, 2004.

**Theorem 1.** *The feature ranges of a compound feature are mutually exclusive, i.e., for two different feature ranges of a compound feature  $(R_{F_1}^a, R_{F_2}^b)$  and  $(R_{F_1}^c, R_{F_2}^d)$ , there is no such feature instance  $I_{FV}^i$  so that  $I_{FV}^i \in I((R_{F_1}^a, R_{F_2}^b), F_{12})$  and  $I_{FV}^i \in I((R_{F_1}^c, R_{F_2}^d), F_{12})$ .*

*Proof.* We prove this by contradiction. If there exists a feature instance  $I_{FV}^i$  so that  $I_{FV}^i \in I((R_{F_1}^a, R_{F_2}^b), F_{12})$  and  $I_{FV}^i \in I((R_{F_1}^c, R_{F_2}^d), F_{12})$ .

$$\begin{aligned} & I_{FV}^i \in I((R_{F_1}^a, R_{F_2}^b), F_{12}) \\ \Leftrightarrow & I_{FV}^i \in I(R_{F_1}^a, F_1) \wedge I(R_{F_2}^b, F_2) \\ \Leftrightarrow & v(I_{FV}^i, F_1) \in R_{F_1}^a, v(I_{FV}^i, F_2) \in R_{F_2}^b \end{aligned} \tag{5}$$

Similarly,

$$\begin{aligned} & I_{FV}^i \in I((R_{F_1}^c, R_{F_2}^d), F_{12}) \\ \Leftrightarrow & v(I_{FV}^i, F_1) \in R_{F_1}^c, v(I_{FV}^i, F_2) \in R_{F_2}^d \end{aligned} \tag{6}$$

Recall that there is no common feature value  $v_F$ , which falls into  $R_F^j$  and  $R_F^k$  simultaneously ( $j \neq k$ ). From (1) and (2), we can get  $R_{F_1}^a = R_{F_1}^c$  and  $R_{F_2}^b = R_{F_2}^d$ . Thus,  $(R_{F_1}^a, R_{F_2}^b) = (R_{F_1}^c, R_{F_2}^d)$ . This contradicts the assumption that the two feature ranges are different.