Noname manuscript No.
(will be inserted by the editor)

Using Instance Weighted Naive Bayes for Adapting Concept Drift in Masquerade Detection

Sevil Sen

Received: date / Accepted: date

Abstract Although there are many approaches proposed for masquerade detection in the literature, few of them consider concept drift; the problem of distinguishing malicious behaviours from the natural change in user behaviours. Researchers mainly focus on updating user behaviours for adapting concept drift in masquerade detection. However these approaches rely on the accuracy of the detector and do not take into account malicious instances which are erroneously added to the updating scheme. In this study, we show that conventional approaches based on instance selection are affected dramatically when misclassified intrusive data is added to the training data. Therefore we propose a new approach based on instance weighting which updates user behaviours gradually according to the weights assigned to each instance, regardless of them being malicious or non-malicious. The results show that the proposed approach outperforms the other updating schemes in the literature, where the malicious instances are more than 5% of the benign instances in the updating, which is very likely to happen due to the high miss rate of the existing detectors.

Keywords concept drift · masquerade detection · anomaly-based detection · insider threats · instance weighting · instance weighted naive bayes

S. Sen
Department of Computer Engineering,
Hacettepe University, 06800,
Ankara, TURKEY
Tel.: +90-312-2977500
Fax: +90-312-2977502
E-mail: ssen@cs.hacettepe.edu.tr

1 Introduction

We usually think that most of the threats are coming from outside. However it is shown that threats from inside can be much more harmful than from the outside [3]. Studies on this research area become more important and more popular, with research on insider threats having accelerated, specifically on masquerade detection, over the last decade. Masqueraders, who impersonate another user for malicious activities, such as exposure of private data, modifying/accessing critical data, installing malicious software for future attacks, pose one of the most significant problems in security today.

Masquerade detection has been studied extensively in the literature. Various techniques such as text mining, Hidden Markov Models, Naive Bayes, Support Vector Machines, information-theoretic approaches and the like are proposed for this problem [2]. However concept drift, which is one of the main problems in masquerade detection, is hardly considered. Concept drift is unavoidable in any anomaly-based detection system and crucial for the ongoing detection of attackers. The conventional approaches mainly overcome this issue through the updating of user profiles and they show that such detectors outperform the detectors without updating. However the main limitation of these approaches is that they update user profiles based on the output of the detectors trained with labelled data. The ability of their adaptation to the concept drift depends on the accuracy of the detector. If the detector misses an intrusive behaviour, it will be added to the training data as benign datum. As will be shown in this study, the performance of these approaches decreases considerably when intrusive data is added to the training data.

An ideal concept drift handling system should quickly adapt to concept drift, be robust to noise and distinguish it from concept drift [31]. To achieve that, a new approach based on Instance Weighted Naive Bayes [11] is proposed in this study for the problem of masquerade detection. Our main assumption is that the training data can contain both benign and malicious data. However we assume that malicious data occurs more rarely than benign data, which is believed to be a realistic assumption. We aim to update a classifier correctly even in the presence of the malicious data. The proposed approach is evaluated with different amounts of malicious data and shown that it outperforms many techniques in the literature. This approach could be applied to any anomaly-based detection system.

To summarize, the contributions of this paper are as follows:

- We showed that the updating mechanisms which employ instance selection methodology relying on the accuracy of a detector are affected dramatically when misclassified intrusive data is added to the training data.
- We adapted the Instance Weighted Naive Bayes approach (IWNB) [11] for masquerade detection based on the assumption that benign data occurs more frequently than malicious data in the training data. This instance weighting approach adjusts to the changes in user behaviours slowly, based on weights assigned according to the similarity of the data to the labelled instances. We demonstrate that this approach accurately characterizes the change in user behaviours, even in the presence of intrusive data. As far as we know, this is the first use of an instance weighting approach for concept drift in masquerade detection.
- The proposed approach was compared with some other techniques in the literature. It was particularly compared with the approach of Naive Bayes with Simple Updating (NBwSU), which outperforms many techniques in the literature. It is shown that the proposed approach outperforms NBwSU when more than 5% of the training data belongs to masqueraders, which is very likely to happen due to the high miss rate of the proposed approaches ($\approx$ 30-40 % or more) in the literature.

In Section 2, approaches proposed for masquerade detection and the concept drift problem are summarized. In Section 3, the details of the de-facto dataset employed in this paper are presented. In Section 4, we describe the proposed method for adapting concept drift on masquerade detection. The experimental results are discussed in Section 5 and it is shown that our proposed approach outperforms other updating mechanisms in the presence of intrusive data. Section 6 is devoted to concluding remarks.

2 Related Work

Schonlau *et al.* [28] presented the problem of differentiating users from masqueraders by introducing a publicly available dataset, called the SEA dataset [24]. They compared six different approaches to detect masqueraders in their study : Uniqueness; Bayes One-Step Markov; Hybrid Multistep Markov; Compression; IPAM; and Sequence-Match. The results showed very low detection rates (between 34.2% and 69.3%) with the false positive rate between 1.4% and 6.7%. Since then, researchers have been applying different methods to the problem and using the SEA dataset as the de-facto standard.

Maxion and Townsted [16] applied Naive Bayes classification to masquerade detection and showed that it outperforms the methods introduced by Schonlau *et al.* [28]. They also discussed what makes a user a harder target, and what makes a masquerader more successful. In the study [16], they applied the same approach to the Greenberg dataset [8] and demonstrated that enriched command lines (with flags, arguments, error codes, and the like) improved masquerade detection. They also demonstrated that Naive Bayes with updating showed the best performance [18]. In Naive Bayes with updating, a detector is initially trained with labelled data. Then the unlabelled data assigned as self data by the detector is added to update the model. Yung [33] updated the user behaviours with feedback from the user. A recent approach using Naive Bayes, where the detection of an attacker is deferred for 2-3 blocks, is also proposed [7]. However, detecting a masquerader after 2-3 consecutive blocks (200-300 commands) might not be acceptable in real world applications. Jung proposed the self-consistent Naive Bayes which estimates the probability of a session being a masquerading session in a multiuser system [34] .

Most of the existing approaches employ multi-class training. However multi-class training might be more appropriate for the problem, due to the issues of privacy. Furthermore, obtaining data from multiple users is time-consuming and non-trivial. Hence, Wang and Stolfo applied one-class Naive Bayes and one-class SVM to the problem, and demonstrated that one-class training works well with the advantages of collecting much less data and more efficient training [32]. Chen proposed one-class classification using length statistics of emerging patterns whose frequency changes significantly from one dataset/class to another [4]. The method is based on the assumption that two command blocks have long emerging patterns if both blocks are typed by the same

user. Salem and Stolfo compared few one-class bag-of-words techniques and showed that one-class SVMs are most practical for users on average [26].

Other approaches exist such as compression-based techniques which assume that data from the same user compresses more readily than mixed data from different users [1][9][28], and sequence-based techniques which use the similarity of command sequences for differentiating users from masqueraders [14][15]. Oka *et al.* [21] take into account not only connected events but also events that are non-adjacent to each other while appearing within a certain distance. They also included both normal and intrusive data in the training data while updating user behaviours. They improved their method using layered networks which outperform all techniques in the literature except Naive Bayes with updating [22]. Seo and Cha [29] applied SVM with sequence based kernel methods which suggested better performance than those applying the SVM with RBF kernel. Another sequence-based approach generally used in bioinformatics is proposed in [5]. Even though the results of the approach are competitive with the Naive Bayes with updating, its computational requirements are quite high. They also discussed command grouping where similar commands in the same group can be substituted for the first time. Huang and Stamp have recently proposed a technique based on Profile Hidden Markov Model with positional information [13].

There are also recent approaches on how to evade masquerade detection. Tapiador and Clark show that a resourceful attacker can achieve his goals and evade detection at the same time [30]. Razo-Zapata *et al.* generate synthetic attacks which are able to escape from the masquerade detectors [23].

Here, we only consider the problem of masquerade detection based on command lines. However there are other approaches using different characteristics of users in order to detect masqueraders such as using system calls [20], modelling users' search behaviours [27], or employing GUI-based features [12]. A detailed review of previous work on masquerade detection can be found in [25].

2.1 Data

The Schonlau dataset is the de-facto standard and widely used in most of the conventional approaches in the literature due to being one of the first datasets made publicly available [28]. We use this standard dataset in this study. In this dataset, 70 users were recorded for several months, with 15,000 commands logged for each user. However the command arguments were not logged due to issues of privacy.

In the studies on masquerade detection, 50 randomly selected users are used to represent benign users, and the remaining 20 users represent masqueraders. The first 5,000 commands of each user are used as training data and the other 10,000 commands are used for testing in the literature. The commands of masqueraders are added into the benign users' testing set randomly in blocks of 100 commands. However these blocks are not equally distributed to each user. If the current block is not belonging to a masquerader, a masquerader block is inserted with 1% probability, otherwise with 80% probability. This setting is called the SEA dataset and has been employed for comparison in most of the studies in the literature.

2.2 The Method

In this study, we adapted an algorithm called Instance Weighted Naive Bayes (IWNB), which was recently proposed by Jiang [11], for the problem of masquerade detection. In this study, this algorithm is proposed to update user profiles for handling the changes in user behaviours, even in the presence of intrusive data. Furthermore, the performance of IWNB is compared with some other updating approaches in the literature. One of these conventional approaches is Naive Bayes with updating which we call Naive Bayes with Simple Updating (NBwSU) here. Since NBwSU outperforms the conventional approaches without updating in the literature, it is employed here as a comparison to our approach. Firstly, for a better understanding we explain below about Naive Bayes and NBwSU, and then IWNB.

2.3 Naive Bayes with Simple Updating (NBwSU)

Naive Bayes is a supervised learning method which has been successfully employed to a range of applications including masquerade detection [16][17]. In Naive Bayes, the probability of a text, t belongs to a class, y is computed as the probability P(y|t) and the highest probability predicts the class in which the text belongs to. In our problem, given a command sequence s, the probability that the command sequence belongs to user x (u_x) can be computed as :

$$P(u_x|s) = \frac{P(s|u_x)P(u_x)}{P(s)}. \quad (1)$$

$P(s)$ is the probability of that specific command sequence occurring and it is usually omitted based on the assumption that each command has equal probability [30][32]. Naive Bayes assumes that all commands in a sequence are independent of each other. Based on this

assumption, the probability that a command sequence s is typed by a particular user u_x can be computed as:

$$P(s|u_x) = \prod_{i=1}^{|s|} P(s_i|u_x). \quad (2)$$

Hence, the formula which calculates the probability that the sequence belongs to user x becomes (the prior $P(u_x)$ is also ignored) :

$$P(u_x|s) = \log(\prod_{i=1}^{|s|} P(s_i|u_x)) = \sum_{i=1}^{|s|} \log(P(s_i|u_x). \quad (3)$$

$P(s_i|u_x)$ is the probability of a command s_i for a particular user x. In this study we use the multinominal event model (bag-of-words approach) for Naive Bayes which usually outperforms the multi-variative Bernoulli model at large vocabulary sizes [19]. Therefore, the probability of a command for a particular user is computed based on the frequency of the command in the training data with the given formula below [16]:

$$P(s_i|u_x) = \frac{N_{s_i,u_x} \ + \ \alpha}{D_l \ + \ \alpha \ A}. \quad (4)$$

Here, N_{s_i,u_x} is the count of the command s_i in the labelled training data D_l. α is a pseudo count to ensure that there are no zero counts for unseen commands, or for command sequences. The lower the α is, the more sensitive the classifier is. Therefore, it is chosen as 0.01 as in [16]. Alphabet size A, which is the number of different commands that the user's type, is determined separately for each user. Training data length is the same and fixed for each user. The labelled training data length is 5,000 here, as in the SEA setting. The block size is chosen as 100 (commands), following other approaches on the SEA dataset in the literature.

In this study, only users' self data is employed as training data. This design called one-class Naive Bayes is believed to be more appropriate for the problem due to issues of privacy.

In Naive Bayes with updating, a detector is initially trained with the labelled data (D_l). Then the unlabelled data (D_u) assigned as self data by the detector is added for updating of the model. If the detector misses an intrusive behaviour, it will also be included as a benign instance in the training data. In this study, updating continuously took place for each command received. We also included all data, regardless of being assigned as self or non-self data by the detector, in updating in order to evaluate how the performance of the detector was affected where malicious data was erroneously included in the training data. Therefore, we named the approach Naive Bayes with Simple Updating, to distinguish it from its application in the literature [18]. In the NBwSU approach, all instances in the training data have equal weights. Hence, the probability of a command s_i for a particular user x is computed as below:

$$P(s_i|u_x) = \frac{N_{(s_i,u_x)_l} \ + N_{(s_i,u_x)_u} \ + \ \alpha}{D_l \ + \ D_u \ + \ \alpha \ A}. \quad (5)$$

2.4 Instance Weighted Naive Bayes (IWNB)

The Instance Weighted Naive Bayes (IWNB) algorithm is a semi-supervised algorithm which employs both labelled and unlabelled data for training data. This approach firstly trains a model using only the labelled data. Then, unlabelled data is included as training data according to the weights calculated using the labelled data. Different weights for different instances are assigned in order to obtain a fine-tuned model. Our approach is similar to the Instance Weighting Naive Bayes approach proposed in 2012, in order to train a multi-classification model when only limited labelled data is available. Jiang [11] proposes an approach where each instance in the unlabelled data is weighted according to the maximal class membership probability for multi-classification. As Jiang states [11], the probability of labelling an unlabelled instance to some extent represents the similarity of this instance to some labelled instances. In this study, we employ a similar approach to the one-class Naive Bayes for masquerade detection.

Firstly, the model is trained by using only labelled data. Then, both labelled and unlabelled data are employed for updating user profiles to deal with the concept drift. Anomaly-based intrusion detection systems are based on the assumption that malicious behaviour will deviate from benign behaviour. Since each datum is weighted according to its similarity with labelled instances, intrusive instances will be assigned different weights than benign instances. Even if malicious instances are added to the training data, their effect will be very low as the majority of the data belongs to benign instances.

Equation 6 shows the probability of a particular command c for a user x, when an unlabelled command sequense s is typed. It is specified as $P(c|u_x)^{|s|}$. When no unlabelled data is typed ($|s| = 0$), the formula becomes equal to Equation 4.

$$P(c|u_x)^{|s|} = \frac{N_{c,u_x} \ + \ \sum_{i=0}^{|s|-1} \delta(s_{i+1}, \ c) P(c|u_x)^i \ + \ \alpha}{D_l \ + \ \sum_{i=0}^{|s|-1} P(s_{i+1}|u_x)^i \ + \ \alpha \ A}. \quad (6)$$

Here, $\delta(s_{i+1}, c)$ is a binary function. It returns one when the current unlabelled command s_{i+1} is equal to

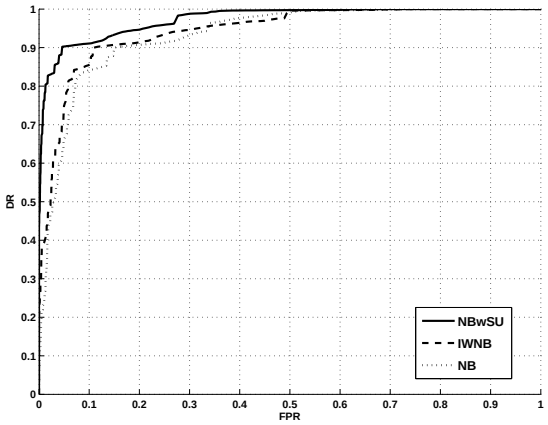

Fig. 1 The Comparison of our approach IWNB with NB, NBwSU

the command c which we want to update the weight (the probability) of, else it is zero. In equation 6, the lowest weights are given to the new commands. The weight is increased gradually as the command is typed. According to our basic assumption, if the new command is typed by a masquerader; the weight will become lower than the weight of new commands typed by benign users. Furthermore, if the attacker types commands used frequently by the user, the probability of a command for the user will not be affected as much under the NBwSU approach. Hence, even if such intrusive instances are added to the training data, they will not be able to change the user behaviour easily over a short period of time.

3 Experimental Results

To understand the effects of the proposed updating scheme on masquerade detection, we performed a number of experiments. Firstly, we compared our approach with one-class Naive Bayes and Naive Bayes with Simple Updating. Here, each user is trained with 5,000 commands as in the SEA setting. The masqueraders are assumed to compromise the system immediately after the training. Hence, the masqueraders will not affect the results. Since we want to observe the performance of these approaches before including intrusive data in the training, thus the updating of user profiles is carried out only with benign data in this experiment and the results are shown in Figure 1. The performance of each method is shown by a ROC curve where the detection rate (DR) is plotted on the Y axis and the false positive rate (FPR) is plotted on the X axis. Each ROC curve is the average of 50 users' results computed by employing threshold averaging [10].

Figure 1 shows that the updating schemes outperform the Naive Bayes without updating as has been indicated in the literature. NBwSU decreases the false positive dramatically as expected, because the training size is increased and each benign instance is added to the training with equal weights. Even though the IWNB approach decreases the false positive rate, the updating in our approach is slower than the NBwSU approach due to the weights assigned to each instance. In this scenario, no intrusive instance is added to the training. However the NBwSU approach adds the instances to the training according to a binary decision given by the detector which does not show perfect accuracy. As a result of that, some intrusive data will eventually be added to the training. Considering the miss alarm rate of the detectors ($\approx$ 30-40 % or more) in the literature, this noise is unavoidable. The falsely classified benign instances will not be included in the training data either in that approach. These falsely classified instances could result from the natural change in user behaviours and they are not taken into account in updating in the NBwSU approach. This is the main issue in concept drift: the difficulty of distinguishing malicious behaviours from the change in user behaviours. Therefore, we assume that intrusive data could also occur in the training data in this study.

In order to evaluate the performance of the updating schemes in the presence of the intrusive data, we add different amounts of intrusive data (from 5% to 50% of self data) to each user. In this experiment, each user has approximately 10,000 commands of self data and it is tested under 500 (5%), 1,000 (10%), 2,000 (20%), 3,000 (30%), 4,000 (40%), and 5,000 (50%) commands of non-self data. The intrusive data is distributed uniformly across the testing data. Each user is evaluated as a potential masquerader for other users. The masquerader data is obtained from other users' training data (first 5,000 commands) in the SEA setting. A weighted Naive Bayes classifier is built for each user whose behaviour is mixed with each masquerader separately. In each test, commands belonging to one masquerader data are added to the training data. Therefore, we can investigate if a masquerader could result in the mis-updating of the user's behaviour, which is crucial for the ongoing detection of masqueraders.

Figure 2 and Figure 3 respectively show the performances of NBwSU and our approach, and IWNB when user behaviours are updated with intrusive data. The results shown here are the averages of all experiments. Figure 2 demonstrates that the performance of NBwSU is severely affected by the presence of intrusive data. On the other hand, since IWNB updates user

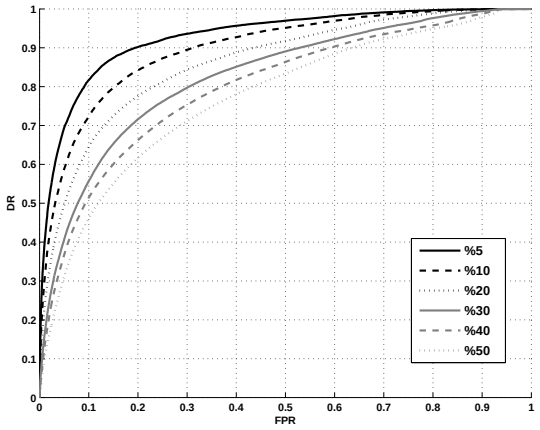

Fig. 2 The Performance of NBwSU under Varying Amount of Intrusive Data

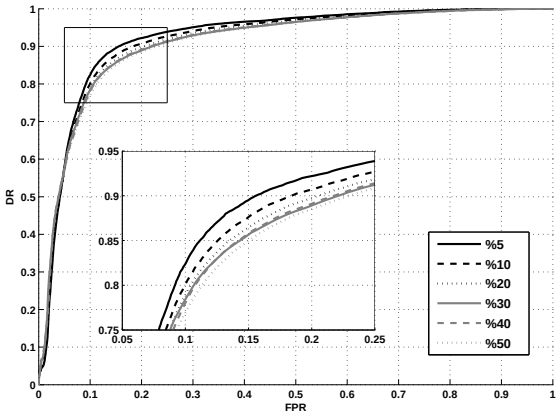

Fig. 3 The Performance of IWNB under Varying Amount of Intrusive Data

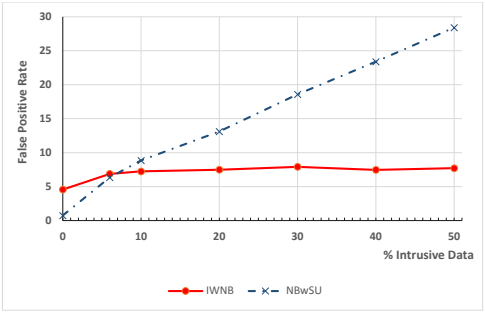

Fig. 4 The Comparison of the False Positive Rates under Varying Amount of Intrusive Data

behaviours gradually, it is not affected that much by the intrusive data. Even if the masquerader types commands amounting to half of what the user types, the detector demonstrates a plausible false positive rate. In these experiments, the malicious data is added uniformly across the users' data for the purpose of simplicity. There could be some usage scenarios where an insider might have regular access to a victim's system, for example; a victim could leave his system unattended for a regular meeting at the same time every day, where a malicious user could step in to take advantage. On the other hand, the access could also be random in real world applications and that randomness could affect the results. For example, if a huge amount of masquerader commands is included in the updating, the model could be deflected to become that of behaviour of the masquerader. It is especially likely to be highly effective in the NBwSU approach, since each masquerader command will update the model with the same weight as that of a user. The model could in fact remain flawed until a sufficient amount of user data is collected.

Figure 4 shows the change in the false positive rate at a detection rate of 70% on both approaches, NBwSU and our approach IWNB, when differing amounts of intrusive data are added to the model. If the intrusive data within the training data is more than $\approx 5\%$ of the self data, our approach clearly shows a better performance than NBwSU.

Lastly, in Figure 5 we compare our approach with some of the approaches in the literature. Many approaches in the literature generally test their results with the six methods presented in [28] and the Naive Bayes methods proposed in [18]. We also present the performance of some recently proposed approaches here.

Table 1 Comprasion with Other Approaches

Method	Hits	False Alarms	Cost
ECM [22]	72.3	2.5	42.7
Two-class NB with updating [18]	61.5	1.3	46.3
IWNB	70.2	4.5	56.8
OCLEP [4]	59.2	2.9	58.2
HMM [13]	70.0	5	60.0
Adaptive NB [7]	83.9	8.8	60.1
Two-class NB [18]	66.2	4.6	61.4
Uniqueness [28]	39.4	1.4	69.0
Hybrid Markov [28]	49.3	3.2	69.9
Bayes One-Step Markov [28]	69.3	6.7	70.9
IPAM [28]	41.1	2.7	75.1
Sequence Matching [28]	36.8	3.7	85.4
Compression [28]	34.2	5.0	95.8

Furthermore the approaches are compared with the cost function below which is introduced in [16].

$$Cost = Misses + 6False\ Alarms \tag{7}$$

In Table 1, the detection methods are ranked by this cost function. As can be seen in the table, IWNB is among the best of the classifiers. Although its performance is below the NB with updating, its advantage is that it performs with a reasonable false positive rate with some intrusive data erroneously included in the training data, as shown in Figure 3 and Figure 4. Moreover, it employs one-class training unlike most other approaches in the table.

To sum up, we indicate that even though a certain amount of intrusive data is included in the updating,

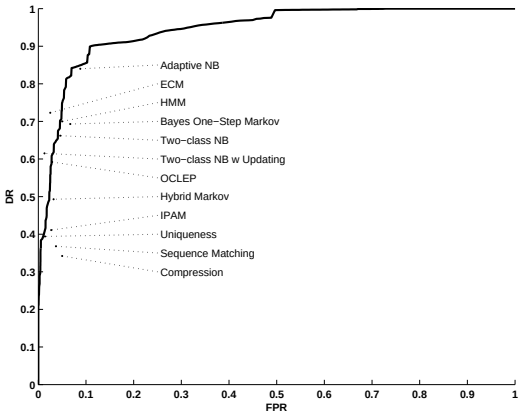

Fig. 5 The Performance of IWNB on the SEA dataset

the integrity of the proposed updating scheme will not be degraded by these malicious instances as much as updating schemes based on instance selection, such as the NBwSU approach. Since all instances have equal weights in NBwSU, a large amount of intrusive data could seriously degrade the performance of the detector. On the other hand, our approach updates user profiles gradually. Since each datum is weighted according to similarity with some labelled instances, intrusive instances will be assigned different weights than benign instances. Contrary to the NBwSU approach, intrusive instances have different weights than benign instances in our approach.

For the sake of simplicity, our training data consists of all malicious and benign data. However we could only append the benign data assigned by the detector as in the Naive Bayes with updating. Hence we might produce better results with smaller false positive rates. While we could eliminate most of the malicious data with this approach, we could also miss the benign data misclassified by the detector. These misclassifications could occur due to the sudden changes in user behaviour. Hence this might result in changes in user behaviours not being represented in the updating model. Two kinds of concept drift could happen in the real world; abrupt and gradual concept drift. We mainly expect users to change their behaviours gradually. Nevertheless, sudden changes could also occur in real life, such as a user's new job, or the installations of new programs, and such like. As IWNB, which includes all instances, regardless of them being malicious or not, it seems to be a better approach than NBwSU in order to handle these abrupt changes.

4 Conclusion

In this study, we proposed a new approach based on instance weighting for adapting concept drift in masquerade detection. Even though there are existing approaches based on instance selection, as far as we know, this is the first application of an instance weighting approach to the problem. Our approach, Instance Weighted Naive Bayes (IWNB), updates user profiles gradually according to the weights assigned to each instance. It has been shown that our approach performs better than the detector without updating.

Masquerade detection is one of the most challenging problems in security today. It becomes even more important as the increase of insider threats are reported [6]. Although there are many solutions proposed for the problem, no detector exists with perfect accuracy, hence misclassified malicious instances will result in the updating schemes. Therefore in this study we evaluated the performance of Naive Bayes with Simple Updating (NBwSU) in the presence of malicious instances. We especially compared our approach with NBwSU, the modified version of Naive Bayes with updating [16], which is indicated as one of the best techniques in the literature. Since equal weights are assigned to each instance in NBwSU, the user behaviours are updated quickly. Although the NBwSU approach shows the best performance in the ideal scenario, where only benign instances are included in the training data during the updating, it is however, dramatically affected with the presence of malicious data. We have shown that the false positive rate increases enormously with the increase of malicious data in the training data. On the other hand, our approach gives a reasonable accuracy even when the masqueraders form 50% of the benign data. It outperformed the NBwSU when the malicious data is much more than 5% of the benign data. This is very likely to happen due to the performance of the current detectors. We have shown that the IWNB approach is highly robust to noise by changing user behaviours slowly. It provides slow but permanent adjustments to the user behaviours. Since we expect a permanent concept drift in this domain, it is very important to be able to update user behaviours accurately even in the presence of masqueraders.

To conclude, we investigated the adapting ability of the updating schemes in the literature, and have presented here our proposed new approach based on instance weighting for the concept drift problem. In the future, we will continue investigating instance weighting approaches on masquerade detection which seem more suitable for the problem due to the performance of the conventional detectors.

References

1. Bertaccihini M, Fierens PI. Preliminary results on masquerader information-theoretic detection using compression-based similarity metrics, IElectron J SADIO 7(1), (2007)
2. Bertaccihini M, Fierens PI. A Survey on Masquerader Detection Approaches, In: Congreso Iberoamericano de Seguridad Informtica, Universidad de la Repblica de Uruguay, pp. 46-60, (2008)
3. Cybersecurity Watch Survey, (2011) : http://www.cert.org/insider_threat/
4. Chen L, Dong G. Masquerader Detection Using OCLEP: One-Class Classification Using Length Statistics of Emerging Patterns, In: the Seventh International Conference on Web-Age Information Management Workshops, (2006)
5. Coull SE, Szymanski BK. Sequence Alignment for Masquerade Detection, Computational Statistics and Data Analysis. Computational Statistics and Data Analysis,52(8), pp. 4116-4131, (2008)
6. Cummings A, Lewellen T, McIntire D, Moore P, Trzeciak R. Insider Threat Study: Illicit Cyber Activity Involving Fraud in the U.S. Financial Services Sector, CERT Report, (2011)
7. Dash SK, Reddy KS, Pujari AK: Adaptive Naive Bayes Method for Masquerade Detection, Security and Communication Networks, vol. 4, pp. 410–417 (2011)
8. Greenberg S. Using Unix: Collected Traces of 168 Users. Research report 88/333/45, Department of Computer Science, University of Calgary, Calgary, Canada (1998)
9. Evans S, Eiland E, Markham S, Impson J, Laczo A, MDLCompress for Intrusion Detection: Signature Inference and Masquerade Attack, In: Proceedings of Military Communications Conference, (2007)
10. Fawcett T. An Introduction to ROC Analysis, Pattern Recognition Letters 27, pp. 861–874, 2006.
11. Jiang L.: Learning Instance Weighted Naive Bayes from Labeled and Unlabeled Data, Journal of Intelligent Information Systems, vol. 38, pp. 257–268 (2012)
12. Garg A, Rahalkar R, Upadhyaya S, Kwiat K. Profiling Users in GUI Based Systems for Masquerade Detection, pp. 48-54, (2006)
13. Huang L, Stamp M. Masquerade Detection Using Profile Hidden Markov Models, Computers & Security 30, pp. 732-747, (2011)
14. Lane T, Brodley CE. Sequence Matching and Learning in Anomaly Detection for Computer Security. In: AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, pp. 43-49, (1997)
15. Latendresse M. Masquerade Detection via Customized Grammars. In: 2nd International Conference on Detection of Intrusion and Malware, and Vulnerability Assessment, LNCS, vol. 3548, pp. 141-159, (2005)
16. Maxion RA, Townsend TN. Masquerade Detection Using Truncated Command Lines. In: International Conference on Dependable Systems & Networks, pp. 219–228, (2002)
17. Maxion RA. Masquerade Detection Using Enriched Command Lines. In: International Conference on Dependable Systems & Networks, pp. 5–14, (2003)
18. Maxion RA, Townsted TN. Masquerade Detection Augmented with Error Analysis. IEEE Transactions on Reliability vol. 53, pp. 124–147, (2004)
19. McCallum A, Nigam K. A Comparison of Event Models for Naive Bayes Text Classification, In: AAAI Workshop on Learning for Text Categorization, (1998)
20. Nguyen N, Reiher P, Kuenning GH. Detecting Insider Threats by Monitoring System Call Activity, In: IEEE Workshop on Information Assurance, pp. 45-52, (2003)
21. Oka M, Oyama Y, Kato K. Eigen Co-occurrence Matrix Method for Masquerade Detection. In: In Publications of the Japan Society for Software Science and Technology, (2004)
22. Oka M, Oyama Y, Abe H, Kato K. Anomaly Detection Using Layered Networks Based on Eigen Co-occurrence Matrix. In: Recent Advances in Intrusion Detection, pp. 223-237, (2004)
23. Razo-Zapata IS, Mex-Perera C, Monroy R. Masquerade Attacks Based on User's Profile, The Journal of Systems and Software 85, pp. 2640–2651, (2012)
24. Schonlau Dataset (2001) `http://www.schonlau.net`
25. Salem MB, Hershkop S, Stolfo SJ. A Survey of Insider Attack Detection Research, Advances in Information Security 39, pp. 69–90, (2008)
26. Salem BS, Stolfo SJ. Detecting Masqueraders: A Comparison of One-Class Bag-of-Words User Behavior Modeling Techniques. In: 2nd International Workshop on Managing Insider Security Threats, pp. 3–13, (2010)
27. Salem MB, Stolfo SJ. Modeling User Search Behavior for Masquerade Detection, In: 14th International Conference on Recent Advances in Intrusion Detection, pp. 181-200, (2011)
28. Schonlau M, DuMoucel W, Ju H, Karr AF, Theus M, Vardi Y. Computer Intrusion: Detecting Masqueraders. Statistical Science. 16(1), 58–74 (2001)
29. Seo J, Cha S. Masquerade Detection Based on SVM and Sequence-Based User Commands Profile, In: 2nd ACM symposium on Information, computer and communications security, pp. 398-400, (2007)
30. Tapiador JE, Clark JA. Masquerade Mimicry Attack Detection: A Randomised Approach, Computers & Security 30, pp. 297-310, (2011)
31. Tsymbal A. The Problem of Concept Drift: Definitions and Related Work, Technical report, TCD-CS-2004-15, Trinity College Dublin, (2004)
32. Wang K, Stolfo SJ. One-Class Training for Masquerade Detection. In: 3rd IEEE Workshop on Data Mining for Computer Security, (2003)
33. Yung, KH. Using Feedback to Improve Masquerade Detection. In: Zhou, J, Yung, M, Han, Y (eds.) ACNS 2003. LNCS, vol. 2846, pp. 48–62. Springer, Heidelberg (2003)
34. Yung, KH. Using Self-Consistent Naive-Bayes to Detect Masquerades. Stanford Electrical and Computer Science Research Journal, pp. 14–21, (2004)