Middle-East Journal of Scientific Research 22 (8): 1212-1218, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.22.08.22028

Adaptive Anomaly Intrusion Detection System Using Optimized Hoeffding Tree and Adaptive Drift Detection Method

S. Ranjitha Kumari and P. KrishnaKumari

Department of Computer Applications, Rathnavel Subramaniam College of Arts and Science, Sulur, Coimbatore – 641402, India

Abstract: In the real time Intrusion Detection system, the main confront is to detect the new attacks rapidly and update the underlying intrusion detection immediately. The data are dynamic in nature in the real time environment and the data evolve over the time gradually or abruptly. This has decreased the performance of the Intrusion Detection system in terms low accuracy rate and high false alarm rate. In order to overcome these pitfalls, we have proposed an Adaptive Anomaly Intrusion Detection system using Optimized Hoeffding Tree and Adaptive Drift Detection method. The proposed model identifies the new attack immediately, identifies changes in the data over time, updates the underlying model and predicts the attacks with high accuracy rate and low false alarm rate. We have used Optimized Hoeffding Tree where the node splitting is controlled using error rate. The concept drift in the evolving data is identified using Adaptive Drift Detection method which uses probability of error rate (Misclassification rate as well as False Alarm Rate) from the Optimized Hoeffding Tree. The use of probability of Misclassification rate as well as False Alarm Rate in identifying the drift has increased the accuracy rate and reduced the false alarm rate of our model. We have compared the results of our Adaptive Anomaly Intrusion Detection system Model with ADWIN change Detector, Page Hinkley Test and EWMA (Exponentially Weighted Moving Average) Control chart detection method using NSL-KDD Dataset. Our model performed better than other models in terms of Accuracy and Low False Alarm Rate in dynamic environment.

Key words: Anomaly Intrusion Detection system · Optimized Hoeffding Tree · Concept Drift · Adaptive Drift Detection Method · NSL KDD dataset

INTRODUCTION

The incredible growth in the field of Information technology has made the entire humankind to depend on the Internet. Business, Entertainment, E-commerce, Education, Social media, Stock Market, etc are fully dependent on the Internet for Information and resource sharing. This has fascinated the hackers to identify the vulnerability and launch new attacks every day. In order to protect the information and data, organizations are deploying Intrusion detection system to protect their network from hackers. An Intrusion is an activity which compromises the confidentiality, integrity and availability of the system.

Intrusion detection system can be categorized into two types: Signature based Intrusion detection system and Anomaly based Intrusion detection system. Signature based Intrusion detection system monitors the network events and compares the incoming events with already available attack patterns. It generates an alarm for intrusion if it finds the match between new event and already existing attack pattern. Usually Signature based Intrusion detection system detects the attack with high accuracy and low false alarm rate. But their ability to detect new attack is low as the new network pattern is not in the existing attack pattern. In contrast, Anomaly based Intrusion detection system learns the normal behavior of the system and any deviation from the normal behavior is identified as an attack. Unlike the former technique, Anomaly based Intrusion detection system identifies new attack competently but tend to produce high false alarm rate as any deviation from the normal behavior by the legitimate user is identified as an attack.

The main challenge in Intrusion detection system is to detect unknown and new attacks with high accuracy rate and low false alarm rate. Several techniques like Data

Corresponding Author: S. Ranjitha Kumari, Department of Computer Applications, Rathnavel Subramaniam College of Arts and Science, Sulur, Coimbatore – 641402, India. Tel: +0422-2687421, E-mail: kkumari@rvsgroup.com.

Mining, Neural Networks, Machine Learning, Statistical techniques, rule mining are used in Intrusion detection system. In the real time Intrusion detection system, we need a model which monitors the incoming data continuously, identifies the attack instantaneously and updates the underlying model for the new attack pattern. As the data in the real time evolve gradually or abruptly, the performance of underlying model depreciates in terms of classifying the attack correctly. Consequently the model should also adapt to the changes to upgrade the performance of the underlying model [1, 2].

In this paper, we have proposed an Adaptive Anomaly Intrusion detection system using stream mining concepts. We have used Optimized Hoeffding Tree and Adaptive Drift Detection method. Optimized Hoeffding Tree, examines the incoming data only once, detects the changes and updates the underlying model incrementally with time and memory constraint. We have used the Optimized Hoeffding Tree where the node splitting phase is controlled using error rate (misclassification rate and false alarm rate) [2].

In real time environment the data evolves over time which degrades the performance of underlying classifier [3, 4]. To handle these evolving incoming data and to increase the performance accuracy of the underlying model, we have used Adaptive Drift Detection method. We have used the probability false alarm rate as well as misclassification rate from the optimized hoeffding tree for drift identification unlike the usual drift detection method where only probability of misclassification rate is used. This method monitors the incoming data and adapts the underlying model once the drift is detected. We have used NSL KDD dataset for experiment. We have compared our results with conventional change detection algorithms like ADWIN Change Detector, Page Hinckley Test and EWMA (Exponentially Weighted Moving Average Control chart) change detector. Our Model has higher performance in terms of high Accuracy rate and low False Alarm rate.

Adaptive Anomaly Intrusion Detection Using Stream Mining: Most of the researchers have contributed towards the highly accurate Intrusion detection models which are static. Static Intrusion Detection system are trained first and implemented in the network. These static models are retrained at regular intervals to update for the new attacks. As there can be an attack at any time in the real time network, the static intrusion detection models are not sufficient to handle new attacks and we need an adaptive model which can handle attacks in real time [5]. Stream mining is the branch of machine learning which handles continuous supply of data with time and memory constraints. Stream mining algorithms are efficient in handling large volume of data continuously in real time unlike data mining algorithms. The incoming data in stream mining are examined once and the underlying data model is updated incrementally with limited memory and time [6, 7]. Stream mining is well suited for adaptive anomaly intrusion detection system as they continuously monitor huge volume of incoming events, identifies the changes in the events quickly and updates the underlying model incrementally for the changes detected [8]. The contribution of our proposed model is

Optimized Hoeffding Tree: The node splitting at the best attribute is controlled using cost of misclassification rate and false alarm rate along with Hoeffding Bound ε and τ to increase the accuracy rate and to minimize the false alarm rate [2].

Adaptive Drift Detection Method: Probability of error rate is used to check the occurrence of drift in the incoming data. If the drift is detected, the model forgets old observations and trains the classifier using new observations. Misclassification rate as well as False Alarm rate from the optimized Hoeffding tree is used to compute the probability of error rate.

Binary Classification: The input data to the intrusion detection system is labeled as 'normal' or 'anomaly' and it uses binary classifier to analyze the same. The performance of the binary classifier is evaluated based on its prediction of the classes precisely. The prediction of the classifier is compared with actual prediction of the classes. The Table 1 shows the confusion matrix of the predictions made by the classifier. The prediction classes are indicated as True Positive, False Negative, False Positive and True Negative.

The performance of a good Intrusion Detection system is measured in terms of Accuracy and False Positive Rate. The ability of the system to correctly classify the input traffic as normal or anomaly is called as Accuracy rate which should be high. False alarm rate is a condition when the system generates alarm when a normal traffic is detected as an anomaly and it should be low always. The accuracy of the number of correctly classified classes is calculated using

$$\text{Accuracy} = (TP + FN)/ (TP + FN + FP + TN) \text{ such that}$$
$$TP + FN = FP + TN = 1$$

Table 1: Confusion Matrix

	Predicted Class Positive	Predicted Class Negative
Actual Class Positive	True Positive (TP)	False Negative(FN)
Actual Class Negative	False Positive (FP)	True Negative (TN)

True positive - Prediction of class as 'normal' and actual class is 'normal'
False positive - Prediction of class as 'anomaly' and actual class is 'normal'
True negative - Prediction of class as 'normal' and actual class is 'anomaly'
False negative - Prediction of class as 'anomaly' and actual class is 'anomaly'

The no. of misclassified instances is calculated using equation

$$C_{mis} = 1 - \text{Accuracy} = 1 - \frac{(TP + FN)}{(TP + FN + FP + TN)}$$

The false alarm rate is calculated using

$$C_{FAR} = \frac{FP}{FP + TN}$$

The objective of this paper is to minimize misclassification (in turn increase accuracy rate) and false alarm rate; hence the total cost to minimize the error in intrusion detection system is computed using the equation

$$C_{error} = C_{mis} + C_{FAR}$$

The minimum and maximum value of C_{mis} and C_{FAR} is 0 and 1 respectively and the mean value is 0.5. Here, the cost of error C_{error} is computed by adding the mean of C_{mis} and C_{FAR}. Hence, the best payoff and the worst payoff for C_{error} are considered as 0 and 1 respectively. The node splitting in the Hoeffding Tree model is controlled by the cost of error rate C_{error} [2].

Proposed Model

Optimized Hoeffding Tree: Hoeffding Tree [9] is a decision tree induction algorithm which learns from continuous and massive data stream. This algorithm examines the data only once and constructs decision tree incrementally with memory and time constraint. The decision tree is constructed by replacing the leaves with decision nodes. The leaves in the decision tree contain the class labels and the nodes contain split attributes. Initially, the data enter through the root of the Hoeffding Tree; the sufficient statistics of the attributes of the data are collected in the leaves. Once the sufficient statistics are collected in the leaf, Hoeffding Bound and information gain of the two best attributes is used to split the attributes and the leaf is converted into a node. Hence the tree grows. In Hoeffding Tree, splitting of nodes is an important phase where the Tree grows by recursively replacing leaves by nodes. The node splitting is performed using information gain difference between two best attributes (x_a,x_b) and Hoeffding Bound HB. If r is the real valued random variable with range R and n is independent observation of this variable r, then the Hoeffding Bound HB states that, with probability $1 - \delta$, the true mean of r is r - □ where

$$\varepsilon = \sqrt{R2 \ln(\frac{1}{\delta})/2N}$$

In this paper we have proposed optimized Hoeffding Tree where the node splitting is controlled using error rate; the prediction phase is performed using Naives Bayes classifier to increase the accuracy rate and to minimize the false alarm rate. The incoming data enters through the root (initially single leaf) of the optimized hoeffding tree and the sufficient statistics of the attributes of the data are collected. In each leaf, the parameter n1 specifies the count of data collected at the leaf. The information gain of the attributes of incoming data is calculated after the grace period of n_{min}, as it is a very amalgam task to calculate the information gain on the arrival of each and every data. After the grace period n_{min}, let x_a be the attribute with the highest information gain and let x_b be the attribute with second highest information gain. The difference between two attributes is computed using

$$\Delta G = (\overline{G}_1(X_a) - (\overline{G}_1(X_b)$$

If $\Delta G > \varepsilon$, then leaf is converted into a node with split on X_a. There are situations when the information gain of two attributes is similar and the decision to select the best attribute may degrade the accuracy rate of the tree. To overcome such situation a user defined threshold τ is used such that, if the Hoeffding bound ε becomes less than τ, the node splits on the current best attribute irrespective of the next best attribute. In the proposed paper, misclassification rate and False alarm rate is used to control the node splitting along with Hoeffding Bound and τ to increase accuracy rate and to minimize the false alarm rate in intrusion detection. The cost of error rate C_{error} is within 0 and 1.The node splitting in optimized Hoeffding tree occurs when $\Delta G > \varepsilon$ and C_{error} is within 0 and 1, else if ϵ < [2].

Algorithm 1: Optimized Hoeffding Tree Algorithm.

1: Let HT be a tree with a single leaf (the root)
2: **for all** training examples **do**
3: Sort example into leaf l using HT
4: Predict class using NaiveBayesPrediction
{
Compute Accuracy = $\frac{TP + FN}{TP + FN + FP + TN}$
Compute $C_{mis} = 1 - Accuacy$
Compute $C_{FAR} = \frac{FP}{FP + TN}$
Return C_{mis}
Return C_{FAR}
}
5: Update sufficient statistics in l
6: Increment n_l the number of examples seen at l
7: **if** n_l **mod** n_{min} **= 0 and** examples seen at l not all of same class **then**
8: Compute $\bar{G}_l(X_i)$ for each attribute
9: Let X_a be attribute with highest $\bar{G}_l$
10: Let X_b be attribute with second-highest $\bar{G}_l$
11: Compute Hoeffding bound $\in = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n_l}}$
12: Calculate $C_{error} = C_{mis} + C_{FAR}$
13: **if** $X_a \neq X_\emptyset$ and $[(\bar{G}_l(X_a) - (\bar{G}_l(X_b) > \in$ and $(0 < C_{error} < 1)]$ or $\in < \tau$ or
14: Replace l with an internal node that splits on X_a
15: **for all** branches of the split **do**
16: Add a new leaf with initialized sufficient statistics
17: **end for**
18: **end if**
19: **end if**
20: **end for**

Prediction Phase: When an event (x, y) arrives where x is the vector of d attributes and y is the class label; it is sorted from root to the leaf using Hoeffding tree algorithm. Three prediction strategies: Majority class, Naïve Bayes and Hybrid adaptive method are used to predict the class of the event in Hoeffding Tree. But in the proposed method Naives Bayes classifier is used to predict the classes. We have not used majority class for prediction because when a new event occurs, it predicts based on the frequent class of examples that were observed during training process. Hence it does not predict the minority class accurately and is partial towards the majority class prediction. Naives Bayes algorithm is based on the Bayesian Model with the independence of the attributes. Bayesian model is easy to build and is suitable for large datasets [10]. Naives Bayes algorithm predicts according to the posterior probability of the class and is represented using

$$p(c/x) = \frac{p(x/c)p(c)}{p(x)}$$

The accuracy of the prediction of the classifier is calculated using the equation

$$Accuracy = \frac{TP + FN}{TP + FN + FP + TN}$$

Limitations of Optimized Hoeffding Algorithm: The optimized Hoeffding algorithm predicts the attack with high accuracy rate and low false alarm rate. Also[10], optimized Hoeffding tree examines the incoming data, detects the changes and updates the model incrementally, but they have certain limitations like

- In the real time intrusion detection, the incoming events evolve over time called as concept drift which degrades the performance of the underlying model in terms of accuracy rate [11].
- The adaption by optimized hoeffding tree is comparatively slow towards concept drift [12].
- As the data are evolving over time, the underlying model must be retrained by removing the old observations and the updations must be done using new observations [12].

Adaptive Drift Detection Method Based on Misclassification and False Alarm Rate: In order to overcome these limitations, Adaptive Drift Detection Method is used in our model. Unlike the regular drift detection method where only misclassification rate is used to detect the drift, both misclassification and false alarm rate are used for the given set of observations to calculate drift [11]. On the detection of change in the incoming data events, the underlying classifier is retrained from the time of warning level using new observations. Hence our model adapts to the changes quickly and prevents the degradation of underlying model in terms of accuracy rate. For all the incoming data events in the stream, Adaptive Drift Detection method monitors the probability of error as a random variable using sequence of Bernoulli trials from the optimized Hoeffding Tree. The probability of error rate is computed using Misclassification rate and False Alarm rate. The error rate and the standard deviation is represented using

$$P_{err} = \frac{C_{mis} + C_{FAR}}{N} \text{ and } S_{sd} = \sqrt{P_{err}(1 - P_{err})/N}$$

where, N is the no. of observations

There are two threshold levels: Warning level and Drift level declared in the Adaptive drift detection method. Initially the minimum error rate and the standard deviation are initiated to infinity ∞. For the given set of observations, if the error rate crosses the first threshold level than the system enters into warning level and the time t_w is stored. Again the error rate is monitored such that, if the error rate drops, then warning level is cancelled. Else, if the error rate increases beyond the second threshold level then system declares the drift detection and the time t_d is stored. The optimized Hoeffding Tree is retrained using the observations from the time t_w and two threshold levels are reset [12].

Algorithm 2 Adaptive Drift Detection Method.

1: Initialize the minimum classification error rate $P_{minerr} = \infty$ and $S_{minsd} = \infty$.
2: Initialize Warning level threshold $t_w = 2$; Drift level threshold $t_d = 3$; No. of observations $w = 30$.
3: Train optimized Hoeffding Tree for the no. of observations $w = 30$.
4: Compute the error rate for the current observations.

5: Let the current error rate $P_{err} = \frac{c_{mis} + C_{FAR}}{N}$ *and* $S_{sd} = \sqrt{P_{err}(1 - P_{err})/N}$

6: if $(P_{err} + S_{sd}) < (P_{minerr} + P_{minsd})$ then $P_{minerr} = P_{err}$ and $S_{minsd} = S_{sd}$
7: if $(P_{err} + S_{sd}) > (P_{minerr} + t_w * S_{minsd})$ then set warning threshold w = 1 and store the time t_w

Else warning threshold is set to w = 0.

8. if $(P_{err} + S_{sd}) > (P_{minerr} + t_d * S_{minsd})$ then change has been detected, set detection threshold $d = 1$ and store the time t_d. Take all the observations since the time t_w and train the current events using optimized hoeffding tree. Reset the value of $t_w = \infty$, $P_{minerr} = \infty$ and $S_{minsd} = \infty$.

The main advantages of our Adaptive Anomaly Intrusion Detection Model using optimized Hoeffding Tree and Adaptive Drift Detection Method are

- The model is suitable for the dynamic environment where the intrusions in the network are identified instantly and the classifier is trained incrementally.
- The adaptive Drift Detection algorithm increases the prediction accuracy of the underlying model by identifying the changes in incoming events rapidly.
- Optimized Hoeffding tree has increased the accuracy rate and decreased the false alarm rate in the underlying model.

NSL-KDD Dataset: We have used NSL-KDD data set for our experiment. NSL-KDD data set is used as it solves the problem in KDD'99 training and test sets which contains huge number of redundant data. The redundant data may lead classification algorithms to be biased towards these redundant records and thus preventing it from classifying other records [13]. NSL-KDD data set are created randomly by sampling records from the #successfulPrediction such that each group has an inverse proportion to the percentage of records in the original group. These train and test data sets called KDD-Train$^+$ and KDD-Test$^+$, because they contain a number of records from all groups and create new data sets. New train and test data sets include 20% of KDD-Train$^+$ and KDD-Test$^+$ data sets without any record with #successfulPrediction equal to 21 [13]. The generated data sets, KDDTrain$^+$ and KDDTest$^+$, includes 125,973 and22,544 records, respectively. Furthermore, one more test set was generated that did not include any of the records that had been correctly classified by all 21 learners, KDDTest^{-21}, which incorporated 11,850, records [4]. In this paper we have used KDD train data as the training data set and have tested our proposed model with KDD test$^+$ and KDD test^{-21} dataset. The NSL-KDD intrusion data set contains 41 attribute categorized into DoS (Denial of Service), R2L (Remote to Local Attack), U2R (User to Root Attack) and Probing Attack [14].

RESULTS

We have evaluated our experiment using MOA tool [15].We have compared the performance of our proposed model in terms of accuracy, false alarm rate, memory and time with other change detectors like ADWIN change detector, Page Hinkley Test and EWMA change detector. The algorithms were trained and tested using prequential valuation method which tests and then trains the dataset. The given model was tested using KDDTrain$^+$, KDDTest$^+$ and KDDTest^{-21} respectively. Table 2 and 3 shows the accuracy and false positive rate using training and test data sets respectively. Figure 1 and Figure 2 depicts the accuracy of correctly classified instances and false positive rate using NSL KDD Test$^+$ and KDDTest^{-21} datasets.

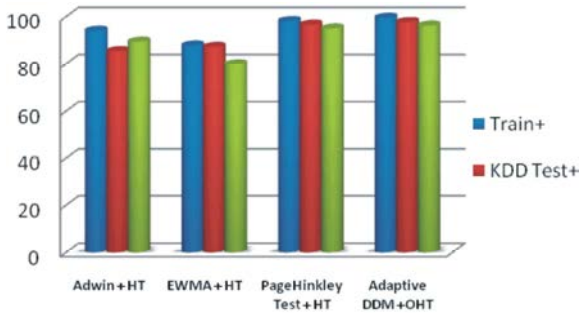

Fig. 1: Accuracy (%)

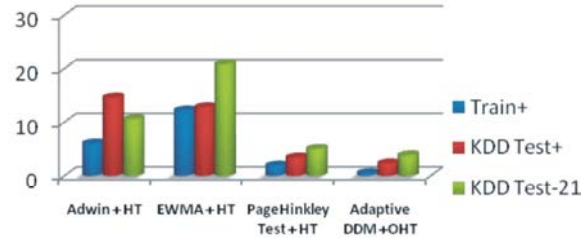

Fig. 2: False Alarm Rate (%)

Table 2: Accuracy

	Accuracy (%)		
Algorithm	Train^{+}	KDD Test^{+}	KDD Test^{-21}
ADWIN + HT	94	85.4	89.4
EWMA + HT	87.8	87.2	79.8
PageHinkley Test + HT	98.1	96.6	95.0
Adaptive DDM + OHT	99.5	97.7	96.2

Table 3: False Alarm Rate

	False Positive Rate (%)		
Algorithm	Train^{+}	KDD Test^{+}	KDD Test^{-21}
ADWIN + HT	6	14.6	10.6
EWMA + HT	12.2	12.8	20.8
PageHinkley Test + HT	1.9	3.4	5
Adaptive DDM + OHT	0.5	2.3	3.8

The experimental results show that proposed model performs better in accuracy and false positive rate. Also, it is efficient in identifying the changes in the incoming data and updating the underlying model at the earliest. The ensemble of PageHinkley Test and Hoeffding Tree performed better than ensemble of Adwin change detector and EWMA change detector. But our Model had the accuracy of 99.50%, 97.70% and 96.20% for KDDTrain^{+}, KDD Test^{+} and KDDTest^{-21} respectively which is better than all other methods. The false positive rate is 0.5%, 2.3% and 3.8% KDDTrain^{+}, KDD Test^{+} and KDDTest^{-21} respectively in our model. Also, the main advantage of our model is that, it detects the changes in the network efficiently and adapts the underlying model instantly in dynamic environment.

CONCLUSION

In this paper we have proposed an Adaptive Anomaly Intrusion Detection Model using Optimized Hoeffding Tree and Adaptive Drift Detection Method which learns quickly and adapts easily to the changes in the network traffic. The node splitting in Hoeffding Tree is optimized using cost of error rate to improve the accuracy rate. The Adaptive Drift Detection Method improved the detection speed of the optimized Hoeffding Tree and identified the drift quickly. The old observations which may degrade the performance of the classifier were removed and the new observations were used to train the model. We have compared the results of proposed model with change detectors like ADWIN change detector, Exponentially Weighted Moving Average Control chart and Page Hinkley Test. The proposed model has got greater accuracy in classifying instances and has low false positive. The Adaptive Drift Detection Method requires less memory compared to other detectors as it does not require separate data structure to evaluate the drift.

REFERENCES

1. Mustafa Amir Faisal, Zeyar Aung, John R. Williams and Abel Sanchez, 2012. Securing Advanced Metering Infrastructure Using Intrusion Detection System with Data Stream Mining, M. Chau *et al*. (Eds.): PAISI 2012, LNCS 7299, 96–111.
2. Ranjitha Kumari, S. and P. KrishnaKumari, 2014. Adaptive Anomaly Intrusion Detection System Using Optimized Hoeffding tree, ARPN Journal of Engineering and Applied Sciences, 9(10).
3. Wang Shuo, Leandro L. Minku, Davide Ghezzi, Daniele Caltabiano, Peter Tino and Xin Yao,Concept Drift Detection for Online Class Imbalance Learning, ieeexplore.ieee.org/iel7/6691896/6706705/06706768.pdf.
4. Petr Kosina and Joao Gama, Handling Time Changing Data with Adaptive VeryFastDecision Rules,www.cs.bris.ac.uk/~flach/ECMLPKDD2012papers/1125570.pdf.
5. Farzaneh Geramiraz, Amir Saman Memaripour and Maghsoud Abbaspour, 2012. Adaptive Anomaly-Based Intrusion Detection System Using Fuzzy Controller.In International Journal of Network Security, 14(6): 352-361.
6. Joao Gama, 2010. Knowledge Discovery from Data Streams, Chapman and Hall/CRC.

7. Pedro Domingos,2000. Mining high-speed data streams. ACM Press, pp: 71-80.
8. Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Richard Kirkby and Ricard Gavald, 2009. A New ensemble methods for evolving data streams. In Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09, New York, NY, USA () ACM, pp: 139-148.
9. Albert Bifet, *et al*., 2011. Data stream Mining - A Practical Approach.
10. http://www.saedsayad.com/naive_bayesian.htm
11. Raquel Sebastiao and Joao Gama., 2009. A Study on Change Detection Methods, epia2009.web.ua.pt/ onlineEdition/ 353.pdf.D.
12. Joao Gama, Indre Zliobaitea, Albert Bifet, Mykola Pechenizkiy And Abdelhamid Bouchachia, 2013. Survey on Concept Drift Adaptation, ACM Computing Surveys, 1(1).
13. Tavalaee, M., E. Bagheri, W. Lu and A. Ghorbani, 2009. A detailed analysis of the KDD CUP 99 data set. In: IEEE Symposium: Computational Intelligence for Security and Defense Applications, CISDA'09, 1-6.
14. nsl.cs.unb.ca/NSL-KDD/.
15. Bifet, A., G. Holmes, R. Kirkby and B. Pfahringer, 2010. Moa: Massive online analysis. Journal of Machine Learning Research (JMLR).